On Motion Estimation Problems in Computer Vision

Jiaolong Yang

A thesis submitted for the degree of
Doctor of Philosophy of
The Australian National University

September 2016

Except where otherwise indicated, this thesis is my own original work. The content is mostly based on the publications during my PhD study as listed below.

Publications

- YANG, J.; LI, H.; AND JIA, Y., 2013. Go-ICP: Solving 3D Registration Efficiently and Globally Optimally. In *International Conference on Computer Vision (ICCV)*, 1457–1464.
- YANG, J.; DAI, Y.; LI, H.; GARDNER, H.; AND JIA, Y., 2013. Single-shot Extrinsic Calibration of a Generically Configured RGB-D Camera Rig from Scene Constraints. In *International Symposium on Mixed and Augmented Reality (ISMAR)*, 181–188.
- YANG, J.; LI, H.; AND JIA, Y., 2014. Optimal Essential Matrix Estimation via Inlier-Set Maximization. In *European Conference on Computer Vision (ECCV)*, 111–126.
- YANG, J. AND LI, H., 2015. Dense, Accurate Optical Flow Estimation with Piecewise Parametric Model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1019–1027.
- YANG, J.; LI, H.; DAI, Y.; AND TAN, R. T., 2016. Robust Optical Flow Estimation of Double-Layer Images under Transparency or Reflection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1410–1419.
- YANG, J.; LI, H.; CAMPBELL, D.; AND JIA, Y., 2016. A Globally Optimal Solution to 3D ICP Point-Set Registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, *38*(11), 2241–2254.
- YANG, J.; REN, P.; CHEN, D.; WEN, F.; LI, H.; AND HUA, G., 2016. Neural Aggregation Network for Video Face Recognition. In *arXiv preprint*, arXiv:1603.05474.

Jiaolong Yang
15 September 2016

Dedicated to my parents and wife.

Abstract

Motion estimation is one of the fundamental problems in computer vision. It has broad applications in the fields of robot navigation, mixed and augmented reality, visual tracking, image and video processing, intelligent transportation systems and so on. Up until now, motion estimation is far from a solved problem, and it is still one of the active research topics in and beyond the computer vision community. This thesis is dedicated to both camera motion estimation – including motion estimation for 3D and 2D cameras – and dense image motion for color images. We push the limits of the state of the art in various aspects such as optimality, robustness, accuracy and flexibility. The main contributions are summarized as follows.

First, a globally optimal 3D point cloud registration algorithm is proposed and applied to motion estimation of 3D imaging devices. Based on Branch-and-Bound (BnB) optimization, we optimally solve the registration problem defined in Iterative Closest Point (ICP). The registration error bounds are derived by exploiting the structure of the $\mathrm{SE}(3)$ geometry. Other techniques such as the nested BnB and the integration with ICP are also developed to achieve efficient registration. Experiments demonstrate that the proposed method is able to guarantee the optimality, and can be well applied in estimating the global or relative motion of 3D imaging devices such as 3D scanners or depth sensors.

Second, a globally optimal inlier-set maximization algorithm is proposed for color camera motion estimation. We use BnB to seek for the optimal motion which gives rise to the maximal inlier set under a geometric error. An explicit, geometrically meaningful relative pose parameterization – a 5D direct product space of a solid 2D disk and a solid 3D ball – is proposed, and efficient, closed-form bounding functions of inlier set cardinality are derived to facilitate the 5D BnB search. Experiments on both synthetic data and real images confirm the efficacy of the proposed method.

Third, a scene constraint based method for relative pose estimation between a 2D color camera and a 3D sensor is developed. We formulate the relative pose estimation as a 2D-3D registration problem minimizing the geometric errors from the known scene constraints. Our method takes only a single pair of color and depth images as input, and is correspondence-free. In addition, a new single-view 3D reconstruction algorithm is proposed for obtaining initial solutions. The experiments show that the method is both flexible and effective, producing accurate relative pose estimates and high-quality color-depth image registration results.

Fourth, a highly-accurate optical flow estimation algorithm based on piecewise parametric motion model is proposed. It fits a flow field piecewise to a variety of parametric models where the domain of each piece (i.e., shape, position and size) and its model parameters are determined adaptively, while at the same time maintaining a global inter-piece flow continuity constraint. The energy function takes into

account both the piecewise constant model assumption and the flow field continuity constraint, enabling the proposed algorithm to effectively handle both homogeneous motions and complex motions. The experiments on three public optical flow benchmarks show that the proposed algorithm achieves top-tier performances.

At last, we propose a robust algorithm for optical flow estimation in the presence of transparency or reflection. It deals with a challenging, frequently encountered, yet not properly investigated problem in optical flow estimation: the input two frames contain one background layer of the scene and one distracting, possibly moving layer due to transparency or reflection. The proposed algorithm performs both optical flow estimation and image layer separation. It exploits a generalized double-layer brightness consistency constraint connecting these two tasks, and utilizes the priors for both of them. The experiments on synthetic and real images confirm its efficacy.

Key Words: Camera Motion, Image Motion, Point Cloud Registration, Branch and Bound, Relative Pose Estimation, Optical Flow, Piecewise Parametric Model, Image Layer Separation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Literature Overview

The view of the 3D world that an imaging device can see at one time-frame or at one single position is always limited. Moving is arguably the foremost way for a vision system to perceive and explore a large 3D space. Motion estimation has become one of the most fundamental and important topics in computer vision. The study begins with 2D projection cameras and their images in the early 1980s [Longuet-Higgins, 1981; Horn and Schunck, 1981; Lucas and Kanade, 1981] as the emergence of the computer vision field, and remains an active topic nowadays with many challenging problems yet to be solved. With the development of ranging techniques, 3D cameras including range-finders, 3D scanners, depth sensors have since become popular in many computer and robotic vision systems. Recovering the motion of 3D cameras, as well as estimating the relative pose between a regular 2D camera and a 3D camera, are also of high interest.

Motion estimation refers to computing or analyzing the motion pattern of the camera or the scene image, based on the data acquired by the vision system at the different time and/or locations. Two typical motion estimation problems exist in computer vision: i) *camera ego-motion estimation*, and ii) *image motion estimation*. Camera ego-motion estimation is the problem of estimating the motion of camera in a static or partially-dynamic scene. It aims at recovering the 3D rigid motion (i.e., rotation and translation) of the camera, or, equivalently, the 3D rigid transformation of camera coordinate systems, using the color or depth/range data captured by the camera. The image motion estimation problem is to compute the pixel movement vectors on the 2D image plane, in the presence of a dynamic scene and/or a moving camera. The problems of camera ego-motion estimation and image motion estimation are closely related to each other: image motion is induced by and reflects the relative motion between the camera and the scene, and the camera motion can be solved from image motion estimates.

Camera motion estimation is a crucial technique to achieve 3D localization and data fusion. It has a broad application in the areas of robot navigation, 3D reconstruction/mapping, mixed and augmented reality, human-computer interaction *etc*. Meanwhile, image motion analysis is widely applied in many high-level computer vi-

sion tasks such as event detection, action recognition, object detection, and tracking. It is also playing an important role in image and video processing such as deblurring, enhancement and compression, and in the autonomous driving and intelligent transportation area which has seen a recent surge of interest.

1.1 The Camera Motion Estimation Problem

Camera motion estimation aims at recovering the motion of the camera in the 3D Euclidean world, using the data obtained at two different spatial locations. The motion can be parameterized by a rigid transformation $\Theta = (\mathbf{R}, \mathbf{t}) \in \mathrm{SE}(3)$, where the 3×3 rotation matrix $\mathbf{R} \in \mathrm{SO}(3)$ is a Special Orthogonal matrix with $\mathbf{R}^{\mathrm{T}}\mathbf{R} = \mathrm{diag}(1,1,1)$ and $\det(\mathbf{R}) = 1$. The 3×3 vector $\mathbf{t} \in \mathbb{R}^3$ is a 3D translation vector. $\mathrm{SE}(3) = \mathrm{SO}(3) \times \mathbb{R}^3$ is called the Special Euclidean Group which corresponds to the parameter space of 3D rigid motion.

Three camera motion estimation problems are considered in this thesis: 1) *3D camera motion estimation*, 2) *2D color camera motion estimation*, and 3) *2D color camera and 3D camera relative pose estimation*. These three camera motion estimation problems are introduced as follows.

1.1.1 3D Camera Motion Estimation

In this thesis, we refer "3D cameras" generally to the devices that can obtain the 3D information of a scene or an object, especially those which can provide dense 3D measurements, including 3D laser rangefinders, Time-of-Flight Cameras, Microsoft Kinect sensors and so on.

As shown in Figure 1.1, a 3D camera can perceive the 3D world by measuring the distances or depths of scene objects, and a 3D point cloud can be obtained from these measurements. A 3D point cloud can be denoted as $\mathcal{X} = \{\mathbf{x}_i\}, i = 1, \ldots, M$, where $\mathbf{x}_i \in \mathbb{R}^3$ is the coordinate vector of a 3D point, and M is the number of 3D points. At two different locations, two point clouds $\mathcal{X}$ and $\mathcal{Y}$ can be obtained by the 3D camera. Since the two point clouds are obtained in the camera coordinate system, *the problem of estimating camera motion* $(\mathbf{R}, \mathbf{t})$ *is equivalent to rigidly registering the two point clouds*. Specifically, given two point clouds from two partial scans of the scene with different camera poses, the relative motion can be estimated by registering the two point clouds. If a global point cloud model $\mathcal{Y}$ of the scene or object is known a priori, then given a point cloud $\mathcal{X}$ from a partial scan, the global motion or absolute pose of the camera can be obtained by registering $\mathcal{X}$ onto $\mathcal{Y}$.

Point cloud registration. The point cloud registration problem can be generally written as

$$\min_{(\mathbf{R},\mathbf{t}) \in \mathrm{SE}(3)} \Phi\Big(f\big(\mathcal{X}, (\mathbf{R}, \mathbf{t})\big),\ \mathcal{Y}\Big), \tag{1.1}$$

where function f applies rigid transformation $(\mathbf{R}, \mathbf{t})$ on $\mathcal{X}$, and $\Phi(\cdot, \cdot)$ is a cost function measuring the registration error. For example, in the well-known Iterative Clos-

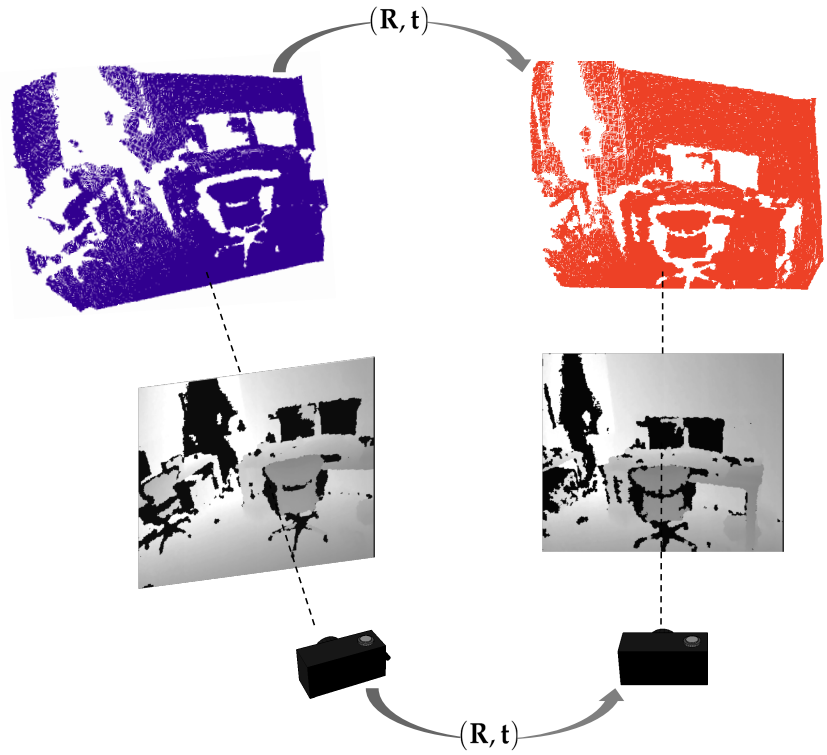


Figure 1.1: 3D camera motion estimation and point cloud registration, illustrated with a depth camera. The gray levels on the visualized depth images indicate the depth to the camera plane: the darker the smaller (closer), expect for the black regions where depth measurements are missing.[1]

est Point (ICP) algorithm [Besl and McKay, 1992; Chen and Medioni, 1991; Zhang, 1994], the registration error in (1.1) is defined as the sum of squared closest-point distances,

$$E(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{N} \min_{j=1,\dots M} \|\mathbf{R}\mathbf{x}_i + \mathbf{t} - \mathbf{y}_j\|^2, \tag{1.2}$$

where $\mathbf{x}_i$, $i = 1, \dots, N$ and $\mathbf{y}_j$, $j = 1, \dots, M$ are the 3D points in $\mathcal{X}$ and $\mathcal{Y}$, respectively. ICP is one of the most classic and widely used algorithms for point-set registration in 2D or 3D. Its concept is simple and intuitive: given an initial transformation (rotation and translation), it alternates between building closest-point correspondences under the current transformation and estimating the transformation with these correspondences, until convergence. Appealingly, ICP is able to work directly on the raw point-sets, regardless of their intrinsic properties (such as distribution, density, and noise level). Due to its conceptual simplicity, high usability and good performance in practice, ICP and its variants are very popular and have been successfully applied in numerous real-world tasks [Newcombe et al., 2011; Seitz et al., 2006; Makela et al.,

[1] Depth images from [Shotton et al., 2013]

2002].

There are two drawbacks of the classic ICP algorithm. First, it is known for its susceptibility to local minima, due to the non-convexity of the problem as well as the local iterative procedure it adopts. It requires a good initialization, without which the algorithm may converge to a wrong registration. Second, the L_2-norm least squares fitting in (1.2) is susceptible to outliers, and a small number of outliers may lead to erroneous registration. Many approaches have been proposed to address the above issues and improve the performance of ICP.

Robust estimation. To address the outlier issue, two common strategies exist based on either 1) robust statistics or 2) robust cost function (instead of the squared distances). For example, some methods reject the corresponding points more than a given distance apart [Champleboux et al., 1992; Pulli, 1999], while some others reject the worst $n\%$ of the pairs with the largest distances (i.e., the trimming strategy in robust statistics) [Pulli, 1999; Chetverikov et al., 2005]. Chetverikov et al. [2005] also proposed an automatic overlap ratio estimation method to find the best trimming percentage. Masuda and Yokoya [1994] computed the rigid motion which minimizes the median of the squared distance residuals. Fitzgibbon [2003] applied robust kernels such as a Lorentzian function or a Huber function to gain outlier-robustness and used Levenberg-Marquardt algorithm [Moré, 1978] to directly minimize the cost function. [Bouaziz et al., 2013] replaced the squared distances in (1.2) with L_p-norm distances where $0 < p \leqslant 1$. It was shown by Jian and Vemuri [2005] that if the point-sets are represented with Gaussian Mixture Models (GMMs), ICP is related to minimizing the Kullback-Leibler (KL) divergence of two GMMs. Improved robustness to outliers are achieved by GMM-based techniques [Jian and Vemuri, 2005; Tsin and Kanade, 2004; Myronenko and Song, 2010; Campbell and Petersson, 2015] using the KL-divergence or the L_2 distance of GMMs.

Dealing with the local minima issue. To deal with the issue of local minima, previous efforts have been devoted to widening the basin of convergence [Fitzgibbon, 2003; Tsin and Kanade, 2004], performing heuristic and non-deterministic global search [Sandhu et al., 2010; Silva et al., 2005] and utilizing other methods for coarse initial alignment [Rusu et al., 2009; Makadia et al., 2006]. For example, better convergence than ICP was observed using the method of Fitzgibbon [2003], especially with the use of robust kernels. The GMM-based methods [Jian and Vemuri, 2005; Tsin and Kanade, 2004; Myronenko and Song, 2010; Campbell and Petersson, 2015] have also shown improved robustness to poor initializations. Some methods extend ICP by robustifying the distance between points using invariant feature descriptors [Sharp et al., 2002] or color [Johnson and Sing, 1999]. A typical family of global registration methods adopts stochastic optimization such as Genetic Algorithms [Silva et al., 2005; Robertson and Fisher, 2002], Particle Swam Optimization [Wachowiak et al., 2004], Particle Filtering [Sandhu et al., 2010] and Simulated Annealing schemes [Blais and Levine, 1995; Papazov and Burschka, 2011]. Another family introduces shape descriptors for coarse alignment, such as Spin Images [Johnson and Hebert, 1999],

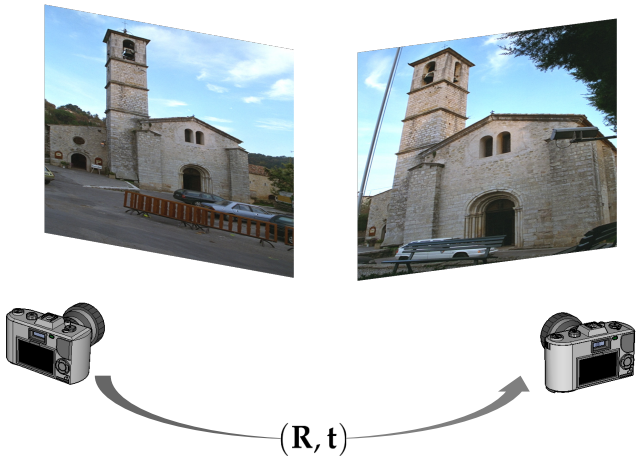


Figure 1.2: 2D color camera motion estimation.[2]

Shape Contexts [Belongie et al., 2002], Integral Volume [Gelfand et al., 2005], Point Feature Histograms [Rusu et al., 2009] and Extended Gaussian Images (EGI) [Makadia et al., 2006]. These descriptor-based methods are typically equipped with random sampling [Rusu et al., 2009], greedy algorithms [Johnson and Hebert, 1999], Hough Transforms [Woodford et al., 2014] or BnB algorithms [Gelfand et al., 2005; Bazin et al., 2012] to compute the registration parameter. The RANSAC algorithm [Fischler and Bolles, 1981] is also used to register raw point clouds directly [Irani and Raghavan, 1999; Aiger et al., 2008].

Although the local minima issue can be alleviated by the aforementioned methods, the global optimality cannot be guaranteed by them. Furthermore, some methods, such as those based on feature matching, are not always reliable or even applicable when the point-sets are not sampled densely from smooth surfaces. Registration methods that guarantee optimality have been published in the past, albeit in a smaller number. Most of them are based on BnB algorithms. For example, geometric BnB has been used for 2D image pattern matching [Breuel, 2003; Mount et al., 1999; Pfeuffer et al., 2012]. However, extending them to 3D is often impractical due to the heightened complexity [Breuel, 2003]. A few optical 3D registration method have been proposed recently, but they either make unrealistic assumptions such as the two point-sets are of equal size [Li and Hartley, 2007], the translation is known a prior [Bazin et al., 2012; Bustos et al., 2014], or assume a small number of putative correspondences exists [Gelfand et al., 2005; Enqvist et al., 2009].

1.1.2 2D Color Camera Motion Estimation

The study of motion estimation for regular, perspective 2D cameras has a long history. 2D color camera motion estimation is often coupled with 3D structure estimation. The process of inferring 3D structure as well as the camera motion from

[2]The color images are from http://www.robots.ox.ac.uk/~vgg/data/data-mview.html.

2D images is known as *Structure from Motion (SfM)*. The relative motion estimation problem of two views is illustrated in Figure 1.2.

Perspective camera model. A 2D camera maps the 3D world (object space) onto a 2D image plane. Cameras with perspective projections are most commonly used, and are of interest in this thesis. We first consider a simple case where the camera coordinate frame coincides with the origin of 3D world coordinate frame. Under a perspective camera model, a point $\mathbf{X} = (X, Y, Z)^{\mathrm{T}}$ in the 3D world is projected onto the image plane as a 2D image point via the following equation:

$$\lambda \tilde{\mathbf{x}} = \mathbf{K}\mathbf{X}. \tag{1.3}$$

In this equation, $\tilde{\mathbf{x}} = (x, y, 1)^{\mathrm{T}}$ is the homogeneous coordinate representation of 2D image pixel location $(x, y)^{\mathrm{T}}$. λ is called the projective "depth" of the point (which equals Z in this simple case). The 3×3 matrix $\mathbf{K}$ is the camera intrinsic matrix, which encompasses the camera's focal length, image sensor format, principal point and axis skew. The intrinsic matrix can be obtained by a calibration procedure using a 3D object with a known 3D shape (such as a planar checkerboard pattern [Zhang, 2000]), or be estimated via self-calibration [Faugeras et al., 1992; Pollefeys et al., 1999]. Throughout this thesis, the intrinsic matrices of the cameras are assumed to be *known*.

In a general coordinate system, the projection relationship is given by

$$\lambda \tilde{\mathbf{x}} = \mathbf{K}(\mathbf{R}\mathbf{X} + \mathbf{t}) = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \tilde{\mathbf{X}} = \mathbf{P}\tilde{\mathbf{X}}. \tag{1.4}$$

where $(\mathbf{R}, \mathbf{t})$ is the extrinsic parameter of the camera, i.e., the 3D rigid transformation from the world coordinate system to the camera coordinate system. $\tilde{\mathbf{X}} = (X, Y, Z, 1)^{\mathrm{T}}$ is the homogeneous coordinate representation, and the 3×4 matrix $\mathbf{P}$ which contains both the intrinsic and extrinsic parameters of the camera is called the camera projection matrix.

Epipolar geometry of two views. Studying two-view geometry is the first step for 3D reconstruction and camera relative motion recovery. The history of the study is closely related to the field of photogrammetry, where the task is obtaining reliable measurement by means of photographs [Slama et al., 1980]. Hauck [1883] is probably the first to develop the relationship between projective geometry and photogrammetry, where the concept of epipoles – the points on the image planes cut by the line joining the two camera centers – was introduced. Von Sanden [1908] then presented comprehensively how to determine the epipoles. Thompson [1959] proposed to an iterative solution of five simultaneous third-order equations to solve the rotation matrix and translation with five points. In computer vision, Longuet-Higgins [1981] proposed a linear algorithm to solve a 3×3 essential matrix with 8 points, and simple solutions to recover translation and rotation from the essential matrix and reconstruct the points. More details about the history of the two-view epipolar geometry can be

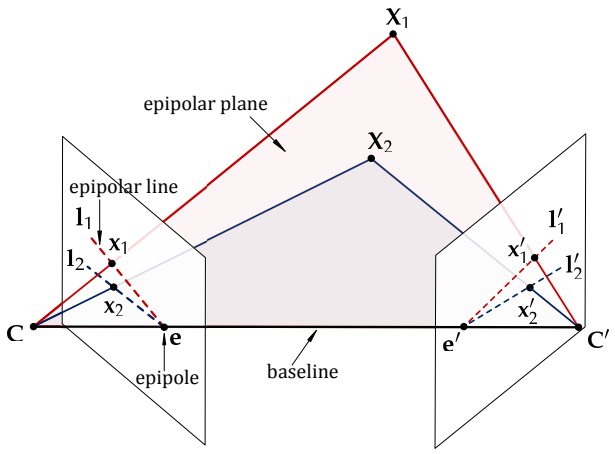


Figure 1.3: Epipolar geometry. The two viewing rays for one 3D point and the camera baseline lie in one epipolar plane, which intersects the two image planes with two epipolar lines. All epipolar lines on each image plane intersect at the epipole.

found in [Kim et al., 2008] (Chapter 2.2).

The epipolar geometry is illustrated in Figure 1.3. Essentially, it shows that the two viewing rays of the two cameras targeting at one 3D point and the camera baseline joining the two camera centers lie in the same plane. The plane is called the *epipolar plane*. It intersects the two image planes in the *epipolar lines*. All the epipolar lines on each image plane intersect at one image point, i.e., the *epipole*. Let the camera coordinate system of the first camera be the world coordinate system, and $(\mathbf{R}, \mathbf{t})$ be the rotation and translation from the first camera to the second camera. Let $\dot{\mathbf{x}}$, $\dot{\mathbf{x}}'$ be two corresponding image points expressed in the *normalized coordinates* (i.e., $\dot{\mathbf{x}} = \mathbf{K}^{-1}\tilde{\mathbf{x}}$ and $\dot{\mathbf{x}}' = \mathbf{K}'^{-1}\tilde{\mathbf{x}}'$ where $\mathbf{K}$ and $\mathbf{K}'$ are the camera intrinsic matrix). Then from the epipolar geometry, we have the following equation:

$$\dot{\mathbf{x}}'^{\mathrm{T}}[\mathbf{t}]_{\times}\mathbf{R}\dot{\mathbf{x}} = \dot{\mathbf{x}}'^{\mathrm{T}}\mathbf{E}\dot{\mathbf{x}} = 0, \tag{1.5}$$

where $[\,\cdot\,]_{\times}$ denotes the skew-symmetric matrix representation of a vector[3], and the matrix $\mathbf{E} = [\mathbf{t}]_{\times}\mathbf{R}$ is the *essential matrix*. This linear equation in fact encodes the coplanar constraint: it entails that $\dot{\mathbf{x}}'$ is perpendicular to $[\mathbf{t}]_{\times}\mathbf{R}\dot{\mathbf{x}}$ – the cross product of $\mathbf{t}$ and $\mathbf{R}\dot{\mathbf{x}}$. Another closely related concept in the uncalibrated case is the *fundamental matrix* $\mathbf{F} = \mathbf{K}'^{-\mathrm{T}}\mathbf{E}\mathbf{K}^{-1}$, with which we have $\tilde{\mathbf{x}}'^{\mathrm{T}}\mathbf{F}\tilde{\mathbf{x}} = 0$. More details regarding the essential matrix and fundamental matrix can be found in [Hartley and Zisserman, 2005] (Chapter 9). Based on the epipolar geometry, most SfM algorithms works by first estimating the essential matrix $\mathbf{E}$ or the fundamental matrix $\mathbf{F}$ from image point correspondences, followed by recovering the camera relative motion $(\mathbf{R}, \mathbf{t})$ and reconstructing the 3D points. Note that the scale of translation cannot be recovered, and a convenient way is to set $\mathbf{t}$ to unit length (i.e., $\|\mathbf{t}\|_2 = 1$).

[3] Let $\mathbf{a} = [a_1, a_2, a_3]^{\mathrm{T}}$, then $[\mathbf{a}]_{\times} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$.

An essential matrix has 5 degrees of freedom, and it is well known that an essential matrix can be determined by at least five pairs of image points [Kruppa, 1913; Faugeras and Maybank, 1990; Heyden and Sparr, 1999]. For practical implementation of the minimal solver (i.e., five-point algorithms), Philip [1996] presented an efficient derivation which leads to solving a thirteenth-degree polynomial. Triggs [2000] derived a 20×20 non-symmetric matrix whose eigenvalues and eigenvectors encode the solutions. Nistér [2003, 2004] refined Philip's method by using a better elimination which leads directly in closed form to the tenth-degree polynomial. Based on the hidden variable technique, Li and Hartley [2006]; Hartley and Li [2012] also derived a tenth-degree polynomial which avoids using variable elimination. For non-minimal solutions, the eight-point solver of Longuet-Higgins [1981] is derived from (1.5) and very simple to implement, albeit it minimizes an algebraic error. Hartley [1997] proposed a normalized eight-point algorithm to estimate the fundamental matrix.

Robust relative motion estimation. The image correspondences to compute the relative pose are usually obtained using image feature detection and matching techniques. Some typical techniques in the early days include the Harris detector [Harris and Stephens, 1988] and its variants [Shi and Tomasi, 1994; Mikolajczyk and Schmid, 2004], Laplacian of Gaussian (LoG) detector [Lindeberg, 1998], *etc*. More recently, high performance algorithms have been developed such as SIFT [Lowe, 2004], SURF [Bay et al., 2006], ORB [Rublee et al., 2011], to name a few. In spite of this, natural or man-made scenes often contain similar structures, flat (and ambiguous) regions, repetitive patterns *etc.*, making flawless feature matching nearly impossible. Therefore, correspondence outliers are ubiquitous in reality.

To deal with outliers, RANSAC [Fischler and Bolles, 1981] and its variants have played a major role. The RANSAC algorithm repeatedly samples a small subset of the points randomly to generate model hypotheses, among which the one with most congruent points is chosen such that a largest congruent set or inlier set is estimated. Many RANSAC variants have been proposed in the context of multiple view geometry and SfM, such as the Locally Optimized RANSAC [Chum et al., 2003], Preemptive RANSAC [Nistér, 2005], PROSAC [Chum and Matas, 2005], Optimal Randomized RANSAC [Chum and Matas, 2008] and GroupSAC [Ni et al., 2009]. A review of these variants and a framework called USAC which combines these techniques are given in [Raguram et al., 2013]. The RANSAC methods are efficient and work quite well in practice. However, being based on random sampling, they cannot provide an optimality guarantee in theory, and the inlier sets they find often vary from time to time. Besides, when non-minimal-case solver is used (such as the linear eight-point algorithm [Longuet-Higgins, 1981; Hartley, 1997]), the algebraic solution is not consistent with the geometric reprojection error or Sampson error typically used to determine inliers/outliers.

There have been some research efforts devoted to optimal essential matrix estimation with inlier-set maximization criterion [Enqvist et al., 2011; Enqvist and Kahl, 2009]. Enqvist et al. [2011] proposed a brute-force search method using triangulation

feasibility test. The solution is exhaustively searched over the discretized parameter space formed by two unit spheres. Enqvist and Kahl [2009] made use of double pairs of correspondences and estimated the camera pose by searching the two epipoles via a branch-and-bound method. An approximation is made to solve an otherwise NP-hard problem (minimum vertex cover). Li [2009] proposed a branch-and-bound method to find the optimal fundamental matrix maximizing the inlier set, where an algebraic error was used to determine inliers.

Another line of robust estimation is outlier removal using convex optimization and the L_∞-norm [Sim and Hartley, 2006; Ke and Kanade, 2007; Olsson et al., 2010]. These methods are able to detect and remove potential outliers, at the expense of losing some true inliers. In the SfM problem, they assume known rotation to formulate the problem to be (quasi-)convex.

Multi-view and large-scale SfM. With the epipolar geometry of two views as an atomic building block, SfM can be achieved for multiple views of a large scale scene by registering the cameras with their relative poses.

The widely used approach to doing this is incrementally incorporating the cameras and refining the results [Pollefeys et al., 2004; Snavely et al., 2006; Klein and Murray, 2007; Agarwal et al., 2009]. Specifically, it involves first building a small reconstruction from two or several views, then growing a few views at a time by registering the images with existing 3D points, followed by triangulating new 3D points and running the bundle adjustment [Triggs et al., 1999; Hartley and Zisserman, 2005]. Bundle adjustment is a nonlinear least square optimization technique which jointly optimize the camera motions, 3D structures, and the camera intrinsic parameters via minimizing the image reprojection error:

$$E(\{\mathbf{P}_j\},\{\mathbf{X}_k\}) = \sum_i \rho\Big(\|\pi(\mathbf{P}_{j(i)},\mathbf{X}_{k(i)}) - \mathbf{x}_i\|_2^2\Big), \tag{1.6}$$

where π is the projection function, ρ is a penalty function which usually downweighs outliers, and $\mathbf{P}_j$, $\mathbf{X}_k$ and $\mathbf{x}_i$ are respectively the camera matrices, the 3D points and the 2D pixel locations. The incremental approach can be time-consuming. The costly bundle adjustment need to be run again and again with increasingly expensive computation as the number of views and reconstruction grow.

In contrast to the incremental SfM, some methods work in a global manner and are able to simultaneously recover all the camera motions based on the pairwise epipolar geometries. Given the relative rotations and translations, they typically first solve for the global rotations and then the global translations [Govindu, 2001; Martinec and Pajdla, 2007; Jiang et al., 2013; Moulon et al., 2013; Crandall et al., 2013]. The global rotations can be estimated by rotation averaging techniques [Hartley et al., 2013] independent of translations. The translations can then be estimated with linear solutions [Govindu, 2001; Jiang et al., 2013]. Crandall et al. [2013] proposed to first solve rotation and translation via discrete labeling in an MRF framework to get coarse estimates, and then apply continuous optimization of non-linear least square to refine

them. The results of these methods are usually refined with one round of global bundle adjustment at the end.

Factorization-based methods [Tomasi and Kanade, 1992; Sturm and Triggs, 1996; Oliensis and Hartley, 2007; Dai et al., 2010] can be deemed as another family of the global methods for multi-view SfM. They usually assume that every 3D point is observed in all input views (which might be too restrictive in practice), and make use of the fact that the measurement matrix, i.e., the matrix composed of 2D point coordinates, is of low rank. The camera motion and structure can thus be estimated by factorizing the low-rank measurement matrix. Tomasi and Kanade [1992] pioneered the research in this direction, and proposed a factorization strategy for affine cameras based on Singular Value Decomposition (SVD). Factorization under perspective cameras is much more difficult due to the unknown projective depths in the measurement matrix. Sturm and Triggs [1996] proposed to first estimate the projective depths from epipolar geometry then perform factorization. This method is then extended by iterative solutions [Triggs, 1996; Mahamud and Hebert, 2000; Mahamud et al., 2001; Oliensis and Hartley, 2007] that alternate between estimating projective depths and performing the factorization. However, the problem is still not fully solved. As shown in [Oliensis and Hartley, 2007], the iterations may often converge to trivial or useless solutions, or run into unstable states. The remedy in Oliensis and Hartley [2007] gives rise to a stable solution with convergence, albeit the solution is biased towards all depths being close to one due to the regularization used. [Dai et al., 2010] proposed a non-iterative solution to the problem by reformulating and relaxing it to a convex semi-definite programming problem. Extensions are presented in [Dai et al., 2010] to handle outliers and missing data.

1.1.3 2D Color Camera and 3D Camera Relative Pose Estimation

Depth cameras can provide the three-dimensional perception of the scene, while conventional 2D color cameras can provide the color of the visual world. 3D geometry and color are complementary information, and can be fused for advanced perception. As the depth camera and color camera are at different locations in the 3D space, the depth and color images they captured cannot be fused directly. To achieve color and depth data fusion, the relative pose between the two cameras is required to register the two images.

The goal of relative pose estimation of a color camera and a depth camera is to compute a 3D rigid transformation $(\mathbf{R}, \mathbf{t}) \in \mathrm{SE}(3)$ between the color and depth camera coordinate systems, as shown in Figure 1.4. This is not an easy task, and can not be achieved by conventional relative pose estimation techniques for color cameras described previously (i.e., computing the motion using feature correspondences). The reason is that a color image and a depth image bear different types of information of the scene, and no suitable cross-modality feature extraction and matching technique exist at present. In fact, feature matching is a difficult task even by manual feature point selection: a salient image point on one image may not be salient enough for manual selection on the other image.

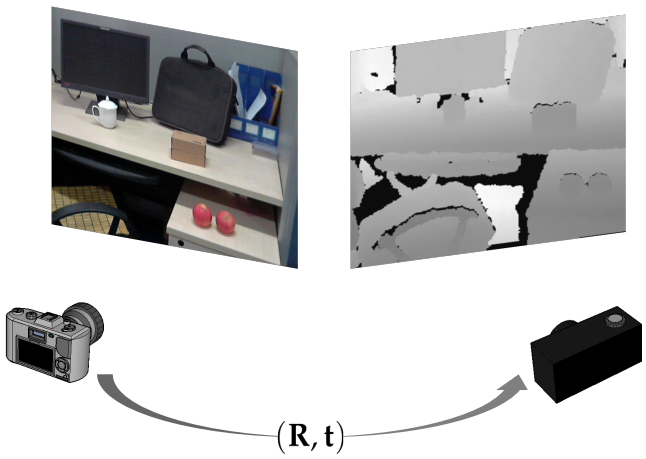


Figure 1.4: 2D color camera and 3D camera relative pose estimation.

Multi-view calibration. Up until now, color and depth camera relative pose estimation has been mostly achieved as a camera extrinsic calibration task, in a way that is very similar to the conventional procedure of calibrating a regular color camera. Typically, this involves the user waving a plate with a checkerboard pattern in front of the camera(s). For example, the camera calibration works by Herrera C et al. [2012] and Zhang and Zhang [2011] are of this type. Herrera C et al. [2012] presented a method to calibrate the intrinsic and extrinsic parameters of two color cameras and a depth camera by using a planar pattern surface. The calibration procedure is similar to the conventional plane-based color camera calibration [Zhang, 2000], i.e., a checkerboard is waved before the cameras and imaged from various poses. The user needs to give the correspondences across the color images and mark the plane region on the depth images. The calibration method of Zhang and Zhang [2011] is similar, although they additionally make use of the correspondences between the color image and the depth image to improve accuracy. Smisek et al. [2011] calibrated Kinect cameras using correspondences between the RGB image and the infrared image.

Among other calibration works, Zhang and Pless [2004] proposed a practical procedure to extrinsically calibrate an RGB camera with a 2D Laser-Rangefinder (LRF), where a checkerboard pattern was moved freely in front of both sensors. Extrinsic calibration was achieved by solving a set of linear constraints which were subsequently refined by iterative minimization of the reprojection error. Likewise, Vasconcelos et al. [2012] also studied the calibration of a color camera with a 2D LRF and they showed that a set of three pairs of planes and lines provides a minimal configuration to solve the calibration problem linearly. Scaramuzza et al. [2007] proposed a method to estimate the relative pose between a color camera and a 3D LRF. However, this method requires manually selecting correspondences between the color image and the depth image. To this end, they convert a range image to a so-called bearing angle image on which natural features of a scene are highlighted to facilitate manual feature selection. Alismail et al. [2012] used a calibration target consisting of a single circle to estimate the extrinsic parameters of a camera-Lidar system. The method

detects the circles (projected as ellipses) and their physical centers on multiple color images, reconstruct them to 3D, and register them onto the point clouds from Lidar to obtain the relative pose.

Single-shot calibration. The aforementioned calibration methods require multiple color and depth (range) image pairs. It is appealing if the relative pose can be estimated by one pair of images from a single shot of the cameras, in a similar way to the 2D camera relative motion estimation described in Section 1.1.2. However, little work that uses a single shot has been published to out knowledge, except for the work by Geiger et al. [2012b]. This is typically because of the difficulty in building cross-modality image correspondences. Geiger et al. [2012b] set up multiple checkerboard patterns in a large scene, such that one color and range image pair is enough to calibrate the cameras. This is essentially similar to a multi-shot configuration. Their calibration process involves an explicit segmentation of the planar regions corresponding to the checkerboard.

1.2 The Image Motion (Optical Flow) Estimation Problem

In computer vision, the apparent movement of brightness/color patterns in a 2D image is called Optical Flow [Horn and Schunck, 1981]. The task of optical flow estimation is to estimate the pixel motions from the observed image data. Specifically, the optical flow estimation is defined as follows.

> *Given two images* $\mathbf{I}$, $\mathbf{I}'$ *which are taken from a regular color camera at two time stamps, estimate for each pixel* $\mathbf{x}$ *in* $\mathbf{I}$ *a 2D motion vector* $\mathbf{u}$, *such that it moves to its corresponding position in* $\mathbf{I}'$.

Figure 1.5 shows two pairs of images and their flow field visualization.

Optical flow is fundamental problem in computer vision. The seminal works are due to Horn and Schunck [1981] and Lucas and Kanade [1981], after which the problem has been heavily investigated in the past decades, with many high performance optical flow algorithms proposed [Brox et al., 2004; Zach et al., 2007; Sun et al., 2014b; Xu et al., 2012; Revaud et al., 2015]. Despite these successes, to obtain dense and accurate flow field remains challenging, especially for general dynamic scenes containing complex and large motions. Up until now, optical flow estimation is still a hot topic in computer vision.

The brightness constancy constraint. The basic assumption to solve the optical flow problem is the so-called Brightness Constancy Constraint (BCC). This constraint entails that, the brightness or color should remain the same when a pixel on the first image moves to its corresponding position in the second image, i.e.,

$$\mathbf{I}(\mathbf{x}) = \mathbf{I}'(\mathbf{x} + \mathbf{u}). \tag{1.7}$$

[4]The two image pairs are from [Liu et al., 2008] and [Butler et al., 2012] respectively. The color coding scheme are from [Baker et al., 2011b].

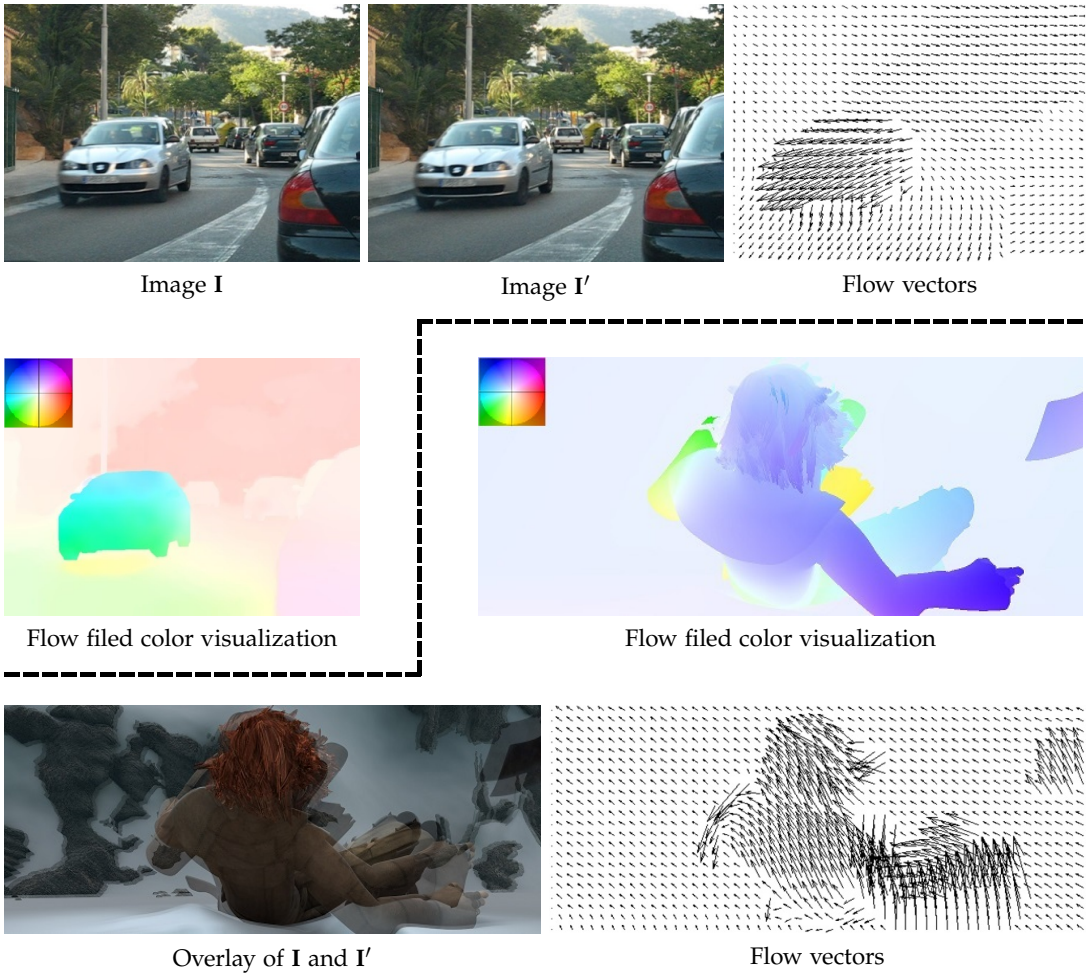


Figure 1.5: Image optical flow estimation.[4]

If we view the image brightness as a function of time, (1.7) can be written as

$$\mathbf{I}(\mathbf{x}, t) = \mathbf{I}(\mathbf{x} + \mathbf{u}, t + 1). \tag{1.8}$$

Using Taylor series, $\mathbf{I}(\mathbf{x} + \mathbf{u}, t + 1)$ can be linearized as

$$\mathbf{I}(\mathbf{x} + \mathbf{u}, t + 1) = \mathbf{I}(\mathbf{x}, t) + u_1 \cdot \mathbf{I}_x + u_2 \cdot \mathbf{I}_y + \mathbf{I}_t + \text{H.O.T.} \tag{1.9}$$

where u_1, u_2 are respectively the horizontal and vertical motions, $\mathbf{I}_x = \partial_x \mathbf{I}(\mathbf{x}, t)$ and $\mathbf{I}_y = \partial_y \mathbf{I}(\mathbf{x}, t)$ are the respectively the horizontal and vertical gradients at $\mathbf{x}$, and $\mathbf{I}_t = \partial_t \mathbf{I}(\mathbf{x}, t) = \mathbf{I}(\mathbf{x}, t + 1) - \mathbf{I}(\mathbf{x}, t)$ is the temporal brightness difference. Consequently, (1.9) can be rewritten as

$$u_1 \cdot \mathbf{I}_x + u_2 \cdot \mathbf{I}_y = -\mathbf{I}_t, \tag{1.10}$$

which is a widely-used formulation. Note that the linearization is only valid for small motions. To handle large motions, one classic strategy is to employ a coarse-

to-fine pyramid warping scheme [Bergen et al., 1992a; Brox et al., 2004; Bruhn et al., 2005].

Although BCC serves as a basic constraint, using it alone cannot solve the optical flow problem effectively, especially for homogenous textureless image regions where the *aperture problem* can occur. To render the optical flow problem trackable, additional constraints should be exploited to regularize the solution, among which the *smoothness constraint* is the most widely used one. According to the use of smoothness constraint, existing methods can be roughly classified into two categories: the *local methods* such as [Lucas and Kanade, 1981] and the *global methods* such as [Horn and Schunck, 1981].

Local methods. The local methods usually compute the flow vector for each pixel independently without considering the global smoothness of the flow field. To compute the flow for a pixel, it only assumes a constant flow vector within a local patch centered at that pixel. A typical formulation is

$$\min_{\mathbf{u}(\mathbf{x})} \sum_{\mathbf{y}\in W_{\mathbf{x}}} \Psi\big(\mathbf{I}(\mathbf{y}) - \mathbf{I}'(\mathbf{y}+\mathbf{u}(\mathbf{x}))\big), \quad \forall \mathbf{x} \in \Omega \tag{1.11}$$

where $\mathbf{u}$ is the flow field and $\mathbf{u}(\mathbf{x})$ is the flow vector for pixel $\mathbf{x}$, $W_{\mathbf{x}}$ is a local image window centered at $\mathbf{x}$, Ω is the 2D image domain of $\mathbf{I}$, and $\Psi(\cdot)$ is a specific penalty function. Typical penalty functions include a squared L_2 norm $\Psi(x) = x^2$ as in [Lucas and Kanade, 1981] and some robust functions such as Truncated L_2 norm and Huber [Senst et al., 2012]. The advantage of local methods is its high efficiency, since the pixel motions can be solved independently, each with small computation costs. For example, linearization is used in [Lucas and Kanade, 1981], and two points or more can be sued to solve an optical flow vector as (1.11) has two unknowns.

Instead of using linearization, some other methods work by enumerating flow vectors and picking the one which best satisfies the BCC [Bao et al., 2014; Lu et al., 2013], thus they suffer less from the local minima issue induced by the linearization and coarse-to-fine warping. One noticeable family which has become popular recently is stimulated by the PatchMatch algorithm [Barnes et al., 2009, 2010]. PatchMatch was originally introduced in the computer graphics community to compute approximate nearest neighbor fields of image patches for image editing. Due to its ability to handle large displacement effectively and efficiently, it has inspired many works in computer vision for optical flow and stereo matching [Bleyer et al., 2011; Lu et al., 2013; Bao et al., 2014; Besse et al., 2014; Li et al., 2015a]. For example, Lu et al. [2013] combined PatchMatch with cost volume filtering techniques [Hosni et al., 2013] and achieved fast and accurate optical flow estimation. Bao et al. [2014] further achieved 5-FPS optical flow estimation with the aid of a modern GPU.

Although being quite fast, purely local methods can hardly compete with global methods in terms of accuracy, according to the optical flow benchmarks [Baker et al., 2011b; Butler et al., 2012; Geiger et al., 2012a]. One reason is their inferior performance for textureless local regions.

Global methods. The global methods have drawn greater attention from the community due to their high-end performances. These methods typically constrains the overall smoothness of the flow field and solve the problem with a variational approach. One widely used variational formulation with a first order smoothness constraint is

$$E(\mathbf{u}) = \int_\Omega \underbrace{\Psi_d\big(\mathbf{I}(\mathbf{x}) - \mathbf{I}'(\mathbf{x}+\mathbf{u}(\mathbf{x}))\big)}_{\text{data term}} + \lambda \underbrace{\big(\Psi_s(\nabla \mathbf{u}_1) + \Psi_s(\nabla \mathbf{u}_2)\big)}_{\text{smoothness term}}\, d\mathbf{x} \tag{1.12}$$

where $\mathbf{u}_1$ and $\mathbf{u}_2$ are respectively the horizontal and vertical flow fields, $\nabla = (\partial_x, \partial_y)^{\mathrm{T}}$ is the gradient operator, and Ψ_d, Ψ_s are the cost functions. For example, with both Ψ_d and Ψ_s as squared L_2 norms $\Psi(x) = x^2$, one may get the formulation in [Horn and Schunck, 1981]. Brox et al. [2004] introduced the Charbonnier penalty function $\Psi(x) = \sqrt{x^2 + \epsilon^2}$ to improve the robustness and preserve discontinuity. The Charbonnier penalty can be viewed as a modified L_1 norm, where ϵ is a small constant (e.g., 0.001 in [Brox et al., 2004]. The Lorentzian penalty $\Psi(x) = \log(1 + \frac{x^2}{2\sigma})$, which is a non-convex penalty is used in [Black and Anandan, 1996] to further improve the robustness. Sun et al. [2010a] investigated a generalized Charbonnier penalty $\Psi(x) = (x^2 + \epsilon^2)^a$ with different a on the Middlebury dataset [Baker et al., 2011b], and concluded that $a = 0.45$ is the best choice. The aforementioned penalty functions are all differentiable everywhere, and solving the minimization problem in 1.13 with these penalty functions typically involve solving a serious of partial differential equations (PDEs) using the Euler-Lagrange equation [Horn and Schunck, 1981].

A non-differentiable formulation was adopted by Zach et al. [2007]; Wedel et al. [2009] where L_1-norm penalty for data term and Total Variation (TV) cost for the smoothness term are used. Specifically, the penalty functions are repsectively $\Psi_d(\cdot) = |\cdot|$ and $\Psi_s(\nabla \mathbf{u}_h) = \sqrt{\partial_x^2 \mathbf{u}_h + \partial_y^2 \mathbf{u}_h}$. To solve the minimization, an auxiliary variable (flow field $\mathbf{v}$) is introduced to split the data and smoothness terms as

$$E(\mathbf{u}) = \int_\Omega \Psi_d\big(\mathbf{I}(\mathbf{x}) - \mathbf{I}'(\mathbf{x}+\mathbf{u}(\mathbf{x}))\big) + \frac{1}{2\theta}(\mathbf{u}-\mathbf{v})^2 + \lambda\big(\Psi_s(\nabla \mathbf{v}_1) + \Psi_s(\nabla \mathbf{v}_2)\big)\, d\mathbf{x} \tag{1.13}$$

where θ is a small constant to ensure $\mathbf{v}$ is close to $\mathbf{u}$. The minimization is performed via optimizing over $\mathbf{u}$ and $\mathbf{v}$ alternately. Solving for $\mathbf{v}$ amounts to a Rudin-Osher-Fatemi (ROF) problem [Rudin et al., 1992], and a efficient, GPU-friendly primal-dual algorithm of Chambolle [2004] was used in [Zach et al., 2007; Wedel et al., 2009]. The TV-L_1 method has become a popular method due to its good performance as well as its efficiency (real-time optical flow estimatin can be obtained using a GPU [Zach et al., 2007; Wedel et al., 2009]). Recently, the Total Generalized Variation (TGV) [Bredies et al., 2010] is proposed, and the second-order TGV which favors piecewise affine fields has been applied to the optical flow problem [Braux-Zin et al., 2013; Ranftl et al., 2014]. The optimization is also based on the primal-dual algorithms [Chambolle, 2004; Chambolle and Pock, 2011].

The evaluation and analysis of some other commonly-used techniques in optical

flow estimation, such as image preprocessing, coarse-to-fine warping and intermediate median filtering, can be found in [Sun et al., 2014b].

1.3 Thesis Outline and Contributions

- In Chapter 2, a globally optimal 3D point cloud registration algorithm is proposed and applied to motion estimation of 3D cameras. Based on Branch-and-Bound (BnB) optimization, we present the first globally optimal solution to the registration problem defined in ICP. By exploiting the special structure of the $\mathrm{SE}(3)$ geometry, novel bounds for the registration error function are derived. Other techniques such as the nested BnB and the integration with ICP are also developed to achieve efficient registration. Experiments demonstrate that the proposed method is able to guarantee the optimality, and can be well applied in estimating the global or relative motion of 3D imaging devices such as 3D scanners or depth sensors.

- In Chapter 3, a globally optimal inlier-set maximization algorithm for color camera motion estimation is proposed to handle feature mismatches. To address the issue that the popular RANSAC algorithm cannot guarantee the largest inlier-set or even can never obtain such a solution, we propose using BnB to seek for the optimal motion which gives rise to the maximal inlier set under a geometric error. An explicit, geometrically meaningful relative pose parameterization – a 5D direct product space of a solid 2D disk and a solid 3D ball – is proposed, and efficient, closed-form bounding functions of inlier set cardinality are derived to facilitate the 5D BnB search. Experiments on both synthetic data and real images confirm the efficacy of the proposed method.

- In Chapter 4, a scene-constraint based method is proposed for relative pose estimation between a 2D color camera and a 3D sensor such as a depth camera. The motivation is to use a single pair of images (i.e. one from each camera similar to color camera motion estimation and depth camera motion estimation) and to provide a correspondence-free solution in order to minimize human intervention. To this end, we propose to make use of known geometric constraints from the scene, and formulate relative pose estimation as a 2D-3D registration problem minimizing the geometric errors from scene constraints. In addition, a new single-view 3D reconstruction algorithm is proposed for obtaining initial solutions. The experiments show that the method is both flexible and effective, producing accurate relative pose estimates and high-quality color-depth image registration results.

- In Chapter 5, a highly-accurate optical flow estimation algorithm based on piecewise parametric motion model is proposed. The proposed algorithm fits a flow field piecewise to a variety of parametric models where the domain of each piece (i.e., shape, position and size) and its model parameters are determined adaptively, while at the same time maintaining a global inter-piece

flow continuity constraint. The novel energy function takes into account both the piecewise constant model assumption and the flow field continuity constraint, enabling the proposed algorithm to effectively handle both homogeneous motions and complex motions. The experiments on three public optical flow benchmarks show that the proposed algorithm achieves top-tier performances.

- In Chapter 6, a robust algorithm for optical flow estimation in the presence of transparency or reflection is proposed. It deals with a challenging, frequently encountered, yet not properly investigated problem in two-frame optical flow estimation. That is, the input frames contain two imaging layers – one desired background layer of the scene, and one distracting, possibly moving layer due to transparency or reflection. The proposed robust algorithm performs both optical flow estimation and image layer separation. It exploits a generalized double-layer brightness consistency constraint connecting these two tasks and utilizes the priors for both of them. To our knowledge, this is the first attempt towards handling generic optical flow fields of two-frame images containing transparency or reflection.

Chapter 2

Globally Optimal 3D Registration and 3D Camera Motion Estimation

Point cloud registration is a fundamental problem in computer and robot vision. Given two sets of points in different coordinate systems, or equivalently in the same coordinate system with different poses, the goal is to find the transformation that best aligns one of the point clouds to the other. Point cloud registration plays an important role in many vision applications. Given multiple partial scans of an object or a scene, it can be applied to merge them into a complete 3D model [Blais and Levine, 1995; Huber and Hebert, 2003] and estimate the relative camera motions. In robot navigation, the global motion of the camera or the robot can be achieved by registering the current view into the global environment [Nüchter et al., 2007; Pomerleau et al., 2013]. In object recognition, fitness scores of a query object with respect to existing model objects can be measured with registration results [Johnson and Hebert, 1999; Belongie et al., 2002]. Given cross-modality data acquired from different sensors with complementary information, registration can be used to fuse the data [Makela et al., 2002; Zhao et al., 2005] or determine the relative poses between these sensors [Yang et al., 2013a; Geiger et al., 2012c].

Among the numerous registration methods proposed in the literature, the Iterative Closest Point (ICP) algorithm [Besl and McKay, 1992; Chen and Medioni, 1991; Zhang, 1994], introduced in the early 1990s, is the most well-known algorithm for efficiently registering two 2D or 3D point sets under Euclidean (rigid) transformation. Its concept is simple and intuitive: given an initial transformation (rotation and translation), it alternates between building closest-point correspondences under the current transformation and estimating the transformation with these correspondences, until convergence. Appealingly, point-to-point ICP is able to work directly on the raw point sets, regardless of their intrinsic properties (such as distribution, density and noise level). Due to its conceptual simplicity, high usability and good performance in practice, ICP and its variants are very popular and have been successfully applied in numerous real-world tasks ([Newcombe et al., 2011],[Seitz et al., 2006],[Makela et al., 2002], for example).

However, ICP is also known for its susceptibility to the problem of local minima, due to the non-convexity of the problem as well as the local iterative procedure it

adopts. Being an iterative method, it requires a good initialization, without which the algorithm may easily become trapped in a local minimum. If this occurs, the solution may be far from the true (optimal) solution, resulting in erroneous estimation. More critically, there is no reliable way to tell whether or not it is trapped in a local minimum.

To deal with the issue of local minima, previous efforts have been devoted to widening the basin of convergence [Fitzgibbon, 2003; Tsin and Kanade, 2004], performing heuristic and non-deterministic global search [Sandhu et al., 2010; Silva et al., 2005] and utilizing other methods for coarse initial alignment [Rusu et al., 2009; Makadia et al., 2006], *etc*. However, global optimality cannot be guaranteed with these approaches. Furthermore, some methods, such as those based on feature matching, are not always reliable or even applicable when the point clouds are not sampled densely from smooth surfaces.

The proposed method in this chapter is, to the best of our knowledge, the first globally optimal solution to the Euclidean registration problem defined by ICP in 3D. Our method always produces the exact and globally optimal solution, up to the desired accuracy. It is named the *Globally Optimal ICP*, abbreviated to *Go-ICP*.

We base the Go-ICP method on the well-established Branch-and-Bound (BnB) theory for global optimization. Nevertheless, choosing a suitable domain parametrization for building a tree structure in BnB and, more importantly, deriving efficient error bounds based on the parametrization are both non-trivial. Our solution is inspired by the $\mathrm{SO}(3)$ space search technique proposed by Hartley and Kahl [2007] as well as Li and Hartley [2007]. We extend it to $\mathrm{SE}(3)$ space search and derive novel bounds of the 3D registration error. Another feature of the Go-ICP method is that we employ, as a subroutine, the conventional (local) ICP algorithm within the BnB search procedure. The algorithmic structure of the proposed method can be summarized as follows.

Use BnB to search the space of $\mathrm{SE}(3)$
  Whenever a better solution is found, call ICP initialized at this solution to refine (reduce) the objective function value. Use ICP's result as an updated upper bound to continue the BnB.
Until convergence.

Our error metric strictly follows that of the original ICP algorithm, that is, minimizing the L_2 norm of the closest-point residual vector. We also show how a trimming strategy can be utilized to handle outliers. With a small effort, one can also extend the method with robust kernels or robust norms.

2.1 Related Work

There is a large volume of work published on ICP and other registration techniques. We will focus below on some relevant Euclidean registration works addressing the

local minimum issue in 2D or 3D.

Robustified Local Methods. To improve the robustness of ICP to poor initializations, previous work has attempted to enlarge the basin of convergence by smoothing out the objective function. Fitzgibbon [2003] proposed the LM-ICP method where the ICP error was optimized with the Levenberg–Marquardt algorithm [Moré, 1978]. Better convergence than ICP was observed, especially with the use of robust kernels.

It was shown by Jian and Vemuri [2005] that if the point sets are represented with Gaussian Mixture Models (GMMs), ICP is related to minimizing the Kullback-Leibler divergence of two GMMs. Although improved robustness to outliers and poor initializations could be achieved by GMM-based techniques [Jian and Vemuri, 2005; Tsin and Kanade, 2004; Myronenko and Song, 2010; Campbell and Petersson, 2015], the optimization was still based on local search. Earlier than these works, Rangarajan et al. [1997] presented a SoftAssign algorithm which assigned Gaussian weights to the points and applied deterministic annealing on the Gaussian variance. Granger and Pennec [2002] proposed an algorithm named Multi-scale EM-ICP where an annealing scheme on GMM variance was also used. Biber and Straßer [2003] developed the Normal Distributions Transform (NDT) method, where Gaussian models were defined for uniform cells in a spatial grid. Magnusson et al. [2009] experimentally showed that NDT was more robust to poor initial alignments than ICP.

Some methods extend ICP by robustifying the distance between points. For example, Sharp et al. [2002] proposed the additional use of invariant feature descriptor distance; Johnson and Sing [1999] exploited color distances to boost the performance.

Global Methods. To address the local minima problem, global registration methods have also been investigated. A typical family adopts stochastic optimization such as Genetic Algorithms [Silva et al., 2005; Robertson and Fisher, 2002], Particle Swam Optimization [Wachowiak et al., 2004], Particle Filtering [Sandhu et al., 2010] and Simulated Annealing schemes [Blais and Levine, 1995; Papazov and Burschka, 2011]. While the local minima issue is effectively alleviated, global optimality cannot be guaranteed and initializations still need to be reasonably good as otherwise the parameter space is too large for the heuristic search.

Another class of global registration methods introduces shape descriptors for coarse alignment. Local descriptors, such as Spin Images [Johnson and Hebert, 1999], Shape Contexts [Belongie et al., 2002], Integral Volume [Gelfand et al., 2005] and Point Feature Histograms [Rusu et al., 2009] are invariant under specific transformations. They can be used to build sparse feature correspondences, based on which the best transformation can be found with random sampling [Rusu et al., 2009], greedy algorithms [Johnson and Hebert, 1999], Hough Transforms [Woodford et al., 2014] or BnB algorithms [Gelfand et al., 2005; Bazin et al., 2012]. Global shape descriptors, such as Extended Gaussian Images (EGI) [Makadia et al., 2006], can be used to find the best transformation maximizing descriptor correlation. These methods are often robust and can efficiently register surfaces where the descriptor can be readily computed.

Random sampling schemes such as RANSAC [Fischler and Bolles, 1981] can also be used to register raw point clouds directly. Irani and Raghavan [1999] randomly sampled 2-point bases to align 2D point sets using similarity transformations. For 3D, Aiger et al. [2008] proposed a 4PCS algorithm that sampled coplanar 4-points, since congruent coplanar 4-point sets can be efficiently extracted with affine invariance.

Globally Optimal Methods. Registration methods that guarantee optimality have been published in the past, albeit in a smaller number. Most of them are based on BnB algorithms. For example, geometric BnB has been used for 2D image pattern matching [Breuel, 2003; Mount et al., 1999; Pfeuffer et al., 2012]. These methods share a similar structure with ours: given each transformation sub-domain, determine for each data point the uncertainty region, based on which the objective function bounds are derived and the BnB search is applied. However, despite uncertainty region computation with various 2D transformations has been extensively explored, extending them to 3D is often impractical due to the heightened complexity [Breuel, 2003].

For 3D registration, Li and Hartley [2007] proposed using a Lipschitzized L_2 error function that was minimized by BnB. However, this method makes unrealistic assumptions that the two point clouds are of equal size and that the transformation is a pure rotation. Olsson et al. [2009] obtained the optimal solution to simultaneous point-to-point, point-to-line and point-to-plane registration using BnB and bilinear relaxation of rotation quaternions. This method, although related to ours, requires known correspondences. Bustos et al. [2014] proposed searching $\mathrm{SO}(3)$ space for optimal 3D geometric matching, assuming known translation. Efficient run-times were achieved using stereographic projection techniques.

Some optimal 3D registration methods assume a small number of putative correspondences, and treat registration as a correspondence outlier removal problem. For example, to minimize the overall pairwise distance error, Gelfand et al. [2005] applied BnB to assign one best corresponding model point for each data point. A similar idea using pairwise consistency was proposed by Enqvist et al. [2009], where the inlier-set maximization was formulated as an NP-hard graph vertex cover problem and solved using BnB. Using angular error, Bazin et al. [2012] solved a similar correspondence inlier-set maximization problem via $\mathrm{SO}(3)$ space search assuming known translation. Enqvist and Kahl [2008] optimally solved camera pose in $\mathrm{SE}(3)$ via BnB. However, the key insight is that with pre-matched correspondences, their pairwise constraint (also used in [Enqvist et al., 2009]) enabled a single translation BnB in $\mathbb{R}^3$ to solve the $\mathrm{SE}(3)$ problem.

In this chapter, we optimally solve the 3D Euclidean registration problem with both rotation and translation. The proposed Go-ICP method is able to work directly on raw sparse or dense point clouds (which may be sub-sampled only for reasons of efficiency), without the need for a good initialization or putative correspondences. The method is related to the idea of $\mathrm{SO}(3)$ space search, as proposed in [Hartley and Kahl, 2007; Li and Hartley, 2007] and extended in [Ruland et al., 2012; Bazin et al., 2012; Yang et al., 2014], *etc*. We extend the 3-dimensional $\mathrm{SO}(3)$ search to

6-dimensional SE(3) search, which is much more challenging.

2.2 Problem Formulation

In this work, we define the L_2-norm registration problem in the same way as in the standard point-to-point ICP algorithm. Let two 3D point clouds $\mathcal{X} = \{\mathbf{x}_i\}, i = 1, ..., N$ and $\mathcal{Y} = \{\mathbf{y}_j\}, j = 1, ..., M$, where $\mathbf{x}_i, \mathbf{y}_j \in \mathbb{R}^3$ are point coordinates, be the *data* point cloud and the *model* point cloud respectively. The goal is to estimate a rigid motion with rotation $\mathbf{R} \in \mathrm{SO}(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$, which minimizes the following L_2-error E,

$$E(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^{N} e_i(\mathbf{R}, \mathbf{t})^2 = \sum_{i=1}^{N} \|\mathbf{R}\mathbf{x}_i + \mathbf{t} - \mathbf{y}_{j^*}\|^2 \tag{2.1}$$

where $e_i(\mathbf{R}, \mathbf{t})$ is the per-point residual error for $\mathbf{x}_i$. Given $\mathbf{R}$ and $\mathbf{t}$, the point $\mathbf{y}_{j^*} \in \mathcal{Y}$ is denoted as the optimal correspondence of $\mathbf{x}_i$, which is the closest point to the transformed $\mathbf{x}_i$ in $\mathcal{Y}$, i.e.,

$$j^* = \underset{j \in \{1,..,M\}}{\operatorname{argmin}} \|\mathbf{R}\mathbf{x}_i + \mathbf{t} - \mathbf{y}_j\|. \tag{2.2}$$

Note the short-hand notation used here: j^* varies as a function of $(\mathbf{R}, \mathbf{t})$ and also depends on $\mathbf{x}_i$.

Equation (2.1) and (2.2) actually form a well-known *chicken-and-egg* problem: if the true correspondences are known *a priori*, the transformation can be optimally solved in closed-form [Horn, 1987; Arun et al., 1987]; if the optimal transformation is given, correspondences can also be readily found. However, the joint problem cannot be trivially solved. Given an initial transformation $(\mathbf{R}, \mathbf{t})$, ICP iteratively solves the problem by alternating between estimating the transformation with (2.1), and finding closest-point matches with (2.2). Such an iterative scheme guarantees convergence to a local minimum [Besl and McKay, 1992].

(Non-)Convexity Analysis. It is easy to see from (2.1) that the transformation function (denote it by $T_x(p)$ for brevity) affinely transforms a point x with parameters p, thus the residual function $e(p) = d(T_x(p))$ is convex provided that *domain D_p is a convex set (Condition 1)* and $d(x) = \inf_{y \in \mathcal{Y}} \|x - y\|$ is convex. Moreover, it has been shown in [Boyd and Vandenberghe, 2004] and further in [Olsson et al., 2009] that $d(x)$ is convex if and only if *$\mathcal{Y}$ is a convex set (Condition 2)*. For registration with pure translation, Condition 1 can be satisfied as the domain D_p is $\mathbb{R}^3$. However, $\mathcal{Y}$ is often a discrete point set sampled from complex surfaces and is thus rarely a convex set, violating Condition 2. Therefore, $e(p)$ is nonconvex. Figure 2.1 shows a 1D example. For registration with rotation, even Condition 1 cannot be fulfilled, as the rotation space induced by the quadratic orthogonality constraints $\mathbf{R}\mathbf{R}^{\mathrm{T}} = \mathbf{I}$ is clearly not a convex set.

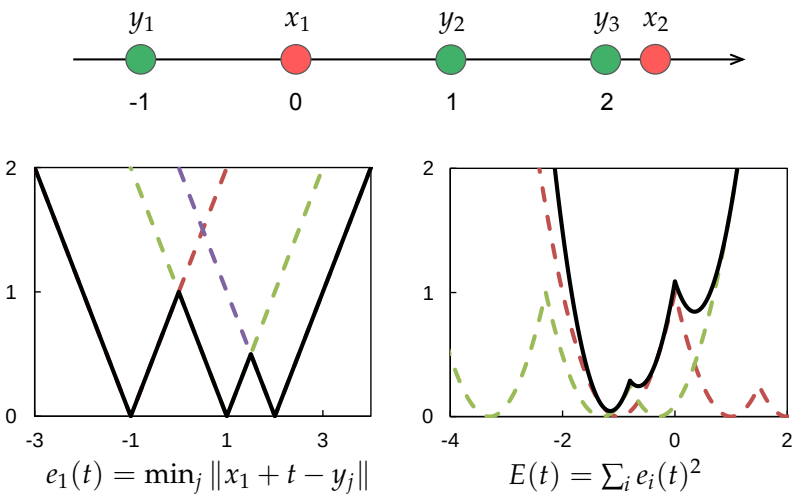
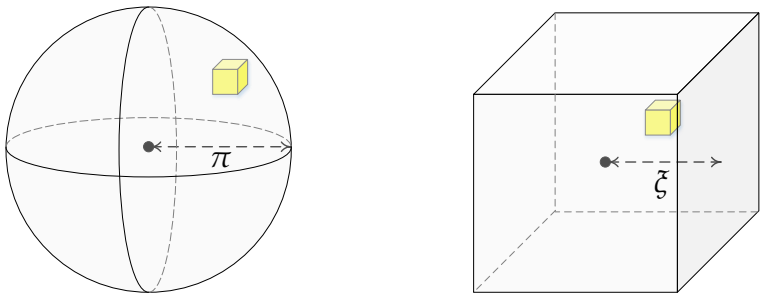


Figure 2.1: Nonconvexity of the registration problem. **Top**: two 1D point sets $\{x_1, x_2\}$ and $\{y_1, y_2, y_3\}$. **Bottom-left**: residual error (closest-point distance) for x_1 as a function of translation t; the three dashed curves are $\|x_1+t-y_j\|$ with $j=1,2,3$ respectively. **Bottom-right**: the overall L_2 registration error; the two dashed curves are $e_i(t)^2$ with $i=1,2$ respectively. The residual error functions are nonconvex, thus the L_2 error function is also nonconvex.

Outlier Handling. As is well known, L_2-norm least squares fitting is susceptible to outliers. A small number of outliers may lead to erroneous registration, even if the global optimum is achieved. There are many strategies to deal with outliers [Rusinkiewicz and Levoy, 2001; Champleboux et al., 1992; Fitzgibbon, 2003; Jian and Vemuri, 2005; Chetverikov et al., 2005]. In this work, a trimmed estimator is used to gain outlier robustness similar to [Chetverikov et al., 2005]. To streamline the presentation and mathematical derivation, we defer the discussion to Section 2.5.3. For now, we assume there are no outliers and focus on minimizing (2.1).

2.3 The Branch and Bound Algorithm

The BnB algorithm is a powerful global optimization technique that can be used to solve nonconvex and NP-hard problems [Lawler and Wood, 1966]. Although existing BnB methods work successfully for 2D registration, extending them to search $\mathrm{SE}(3)$ and solve 3D rigid registration has been much more challenging [Breuel, 2003; Li and Hartley, 2007]. In order to apply BnB to 3D registration, we must consider *i*) how to parametrize and branch the domain of 3D motions (Section 2.3.1), and *ii*) how to efficiently find upper bounds and lower bounds (Section 2.4).



(a) Rotation domain (b) Translation domain

Figure 2.2: SE(3) space parameterization for BnB. **Left**: the rotation space SO(3) is parameterized in a solid radius-π ball with the angle-axis representation. **Right**: the translation is assumed to be within a 3D cube $[-\xi, \xi]^3$ where ξ can be readily set. The octree data-structure is used to divide (branch) the domains and the yellow box in each diagram represents a sub-cube.

2.3.1 Domain Parametrization

Recall that our goal is to minimize the error E in (2.1) over the domain of all feasible 3D motions (the SE(3) group, defined by $\mathrm{SE}(3) = \mathrm{SO}(3) \times \mathbb{R}^3$). Each member of SE(3) can be minimally parameterized by 6 parameters (3 for rotation and 3 for translation).

Using the *angle-axis representation*, each rotation can be represented as a 3D vector $\mathbf{r}$, with axis $\mathbf{r}/\|\mathbf{r}\|$ and angle $\|\mathbf{r}\|$. We use $\mathbf{R}_\mathbf{r}$ to denote the corresponding rotation matrix for $\mathbf{r}$. The 3x3 matrix $\mathbf{R}_\mathbf{r} \in \mathrm{SO}(3)$ can be obtained by the matrix exponential map as

$$\mathbf{R}_\mathbf{r} = \exp([\,\mathbf{r}\,]_\times) = \mathbf{I} + \frac{[\,\mathbf{r}\,]_\times \sin\|\mathbf{r}\|}{\|\mathbf{r}\|} + \frac{[\,\mathbf{r}\,]_\times^2 (1-\cos\|\mathbf{r}\|)}{\|\mathbf{r}\|^2} \tag{2.3}$$

where $[\,\cdot\,]_\times$ denotes the skew-symmetric matrix representation

$$[\,\mathbf{r}\,]_\times = \begin{bmatrix} 0 & -r^3 & r^2 \\ r^3 & 0 & -r^1 \\ -r^2 & r^1 & 0 \end{bmatrix} \tag{2.4}$$

where r^i is the ith element in $\mathbf{r}$. Equation (2.3) is also known as the *Rodrigues' rotation formula* [Hartley and Zisserman, 2004a]. The inverse map is given by the matrix logarithm as

$$[\,\mathbf{r}\,]_\times = \log \mathbf{R}_\mathbf{r} = \frac{\|\mathbf{r}\|}{2\sin\|\mathbf{r}\|}(\mathbf{R}_\mathbf{r} - \mathbf{R}_\mathbf{r}^\mathrm{T}) \tag{2.5}$$

where $\|\mathbf{r}\| = \arccos\big((\mathrm{trace}(\mathbf{R}_\mathbf{r}) - 1)/2\big)$. With the angle-axis representation, the entire 3D rotation space can be compactly represented as a solid radius-π ball in $\mathbb{R}^3$. Rotations with angles less than (or, equal to) π have unique (or, two) corresponding angle-axis representations on the interior (or, surface) of the ball. For ease of manip-

ulation, we use the minimum cube $[-\pi, \pi]^3$ that encloses the π-ball as the rotation domain.

For the translation part, we assume that the optimal translation lies within a bounded cube $[-\xi, \xi]^3$, which may be readily set by choosing a large number for ξ.

During BnB search, initial cubes will be subdivided into smaller sub-cubes C_r, C_t using the *octree data-structure* and the process is repeated. Figure 2.2 illustrates our domain parametrization.

2.4 Bounding Function Derivation

For our 3D registration problem, we need to find the bounds of the L_2-norm error function used in ICP within a domain $C_r \times C_t$. Next, we will introduce the concept of an *uncertainty radius* as a mathematical preparation, then derive our bounds based on it.

2.4.1 Uncertainty Radius

Intuitively, we want to examine the uncertainty region of a 3D point $\mathbf{x}$ perturbed by an arbitrary rotation $\mathbf{r} \in C_r$ or a translation $\mathbf{t} \in C_t$. We aim to find a ball, characterized by an uncertainty radius, that encloses such an uncertainty region. We will use the first two lemmas of [Hartley and Kahl, 2009] in the following derivation. For convenience, we summarize both lemmas in a single Lemma shown below.

Lemma 2.1. *For any vector* $\mathbf{x}$ *and two rotations* $\mathbf{R}_\mathbf{r}$ *and* $\mathbf{R}_{\mathbf{r}_0}$ *with* $\mathbf{r}$ *and* $\mathbf{r}_0$ *as their angle-axis representations, we have*

$$\angle(\mathbf{R}_\mathbf{r}\mathbf{x}, \mathbf{R}_{\mathbf{r}_0}\mathbf{x}) \leqslant \angle(\mathbf{R}_\mathbf{r}, \mathbf{R}_{\mathbf{r}_0}) \leqslant \|\mathbf{r} - \mathbf{r}_0\|, \tag{2.6}$$

where $\angle(\mathbf{R}_\mathbf{r}, \mathbf{R}_{\mathbf{r}_0}) = \arccos\big((\mathrm{trace}(\mathbf{R}_\mathbf{r}^\mathrm{T}\mathbf{R}_{\mathbf{r}_0}) - 1)/2\big)$ *is the angular distance between rotations.*

The second inequality in (2.6) means that the angular distance between two rotations on the SO(3) manifold is less than the Euclidean vector distance of their angle-axis representations in $\mathbb{R}^3$. Based on this Lemma, uncertainty radii are given as follows.

Theorem 2.1. *(Uncertainty radius) Given a 3D point* $\mathbf{x}$*, a rotation cube* C_r *of half side-length* σ_r *with* $\mathbf{r}_0$ *as the center and examining the maximum distance from* $\mathbf{R}_\mathbf{r}\mathbf{x}$ *to* $\mathbf{R}_{\mathbf{r}_0}\mathbf{x}$*, we have* $\forall \mathbf{r} \in C_r$*,*

$$\|\mathbf{R}_\mathbf{r}\mathbf{x} - \mathbf{R}_{\mathbf{r}_0}\mathbf{x}\| \leqslant 2\sin(\min(\sqrt{3}\sigma_r/2, \pi/2))\|\mathbf{x}\| \doteq \gamma_r. \tag{2.7}$$

Similarly, given a translation cube C_t *with half side-length* σ_t *centered at* $\mathbf{t}_0$*, we have* $\forall \mathbf{t} \in C_t$*,*

$$\|(\mathbf{x} + \mathbf{t}) - (\mathbf{x} + \mathbf{t}_0)\| \leqslant \sqrt{3}\sigma_t \doteq \gamma_t. \tag{2.8}$$

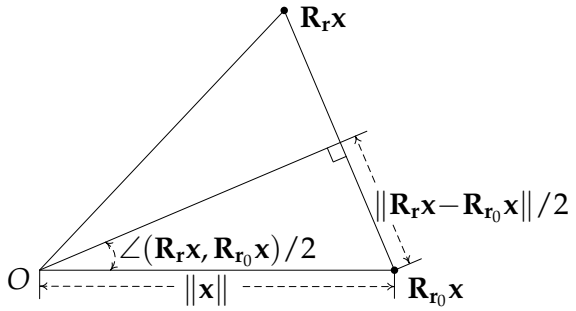


Figure 2.3: Distance computation from $\mathbf{R_r x}$ to $\mathbf{R}_{\mathbf{r}_0}\mathbf{x}$ used in the derivation of the rotation uncertainty radius.

Proof: Inequality (2.7) can be derived from

$$\begin{aligned}
&\|\mathbf{R_r x} - \mathbf{R}_{\mathbf{r}_0}\mathbf{x}\| && (2.9)\\
&= 2\sin(\angle(\mathbf{R_r x}, \mathbf{R}_{\mathbf{r}_0}\mathbf{x})/2)\|\mathbf{x}\| && (2.10)\\
&\leqslant 2\sin(\min(\angle(\mathbf{R_r}, \mathbf{R}_{\mathbf{r}_0})/2, \pi/2))\|\mathbf{x}\| && (2.11)\\
&\leqslant 2\sin(\min(\|\mathbf{r} - \mathbf{r}_0\|/2, \pi/2))\|\mathbf{x}\| && (2.12)\\
&\leqslant 2\sin(\min(\sqrt{3}\sigma_r/2, \pi/2))\|\mathbf{x}\| && (2.13)
\end{aligned}$$

where (2.10) is illustrated in Figure 2.3. Inequalities (2.11), (2.12) are based on Lemma 1, and (2.13) is from the fact that $\mathbf{r}$ resides in the cube.

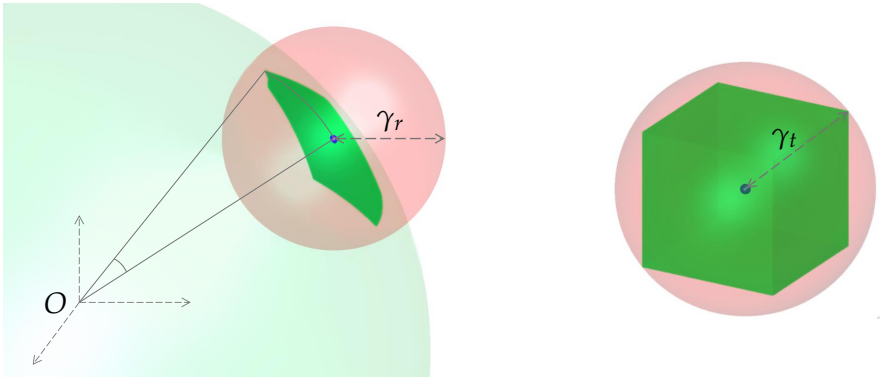
Inequality (2.8) can be trivially derived via $\|(\mathbf{x} + \mathbf{t}) - (\mathbf{x} + \mathbf{t}_0)\| = \|\mathbf{t} - \mathbf{t}_0\| \leqslant \sqrt{3}\sigma_t$. □

We call γ_r the rotation uncertainty radius, and γ_t the translation uncertainty radius. They are depicted in Figure 2.4. Note that γ_r is point-dependent, thus we use γ_{r_i} to denote the rotation uncertainty radius at $\mathbf{x}_i$ and the vector $\boldsymbol{\gamma}_r$ to represent all γ_{r_i}. Based on the uncertainty radii, the bounding functions are derived in the following section.

2.4.2 Bounding the L_2 Error

Given a rotation cube C_r centered at $\mathbf{r}_0$ and a translation cube C_t centered at $\mathbf{t}_0$, we will first derive valid bounds of the residual $e_i(\mathbf{R}, \mathbf{t})$ for a single point $\mathbf{x}_i$.

The upper bound of e_i can be easily chosen by evaluating the error at any $(\mathbf{r}, \mathbf{t}) \in C_r \times C_t$. Finding a suitable lower bound for the L_2 error is a harder task. From Section 2.4.1 we know that, with rotation $\mathbf{r} \in C_r$ (or, translation $\mathbf{t} \in C_t$), a transformed point $\mathbf{x}_i$ will lie in the uncertainty ball centered at $\mathbf{R}_{\mathbf{r}_0}\mathbf{x}_i$ (or, $\mathbf{x}_i + \mathbf{t}_0$) with radius γ_{r_i} (or, γ_t). For both rotation and translation, it therefore lies in the uncertainty ball centered at $\mathbf{R}_{\mathbf{r}_0}\mathbf{x}_i + \mathbf{t}_0$ with radius $\gamma_{r_i} + \gamma_t$. Now we need to consider the smallest residual error that is possible for $\mathbf{x}_i$. We have the following theorem, which is the cornerstone of the proposed method.



(a) Rotation uncertainty radius (b) Translation uncertainty radius

Figure 2.4: Uncertainty radii at a point. **Left:** rotation uncertainty ball for C_r (in red) with center $\mathbf{R}_{\mathbf{r}_0}\mathbf{x}$ (blue dot) and radius γ_r. **Right:** translation uncertainty ball for C_t (in red) with center $\mathbf{x}+\mathbf{t}_0$ (blue dot) and radius γ_t. In both diagrams, the uncertainty balls enclose the range of $\mathbf{R}_\mathbf{r}\mathbf{x}$ or $\mathbf{x}+\mathbf{t}$ (in green).

Theorem 2.2. *(Bounds of per-point residuals) For a 3D motion domain $C_r \times C_t$ centered at $(\mathbf{r}_0, \mathbf{t}_0)$ with uncertainty radii γ_{r_i} and γ_t, the upper bound $\overline{e_i}$ and the lower bound $\underline{e_i}$ of the optimal registration error $e_i(\mathbf{R}_\mathbf{r}, \mathbf{t})$ at $\mathbf{x}_i$ can be chosen as*

$$\overline{e_i} \doteq e_i(\mathbf{R}_{\mathbf{r}_0}, \mathbf{t}_0), \tag{2.14}$$

$$\underline{e_i} \doteq \max\left(e_i(\mathbf{R}_{\mathbf{r}_0}, \mathbf{t}_0) - (\gamma_{r_i}+\gamma_t), 0\right). \tag{2.15}$$

Proof: The validity of $\overline{e_i}$ is obvious: error e_i at the specific point $(\mathbf{r}_0, \mathbf{t}_0)$ must be larger than the minimal error within the domain, i.e., $e_i(\mathbf{R}_{\mathbf{r}_0}, \mathbf{t}_0) \geqslant \min_{\forall(\mathbf{r},\mathbf{t})\in(C_r\times C_t)} e_i(\mathbf{R}_\mathbf{r}, \mathbf{t})$. We now focus on proving the correctness of $\underline{e_i}$.

As defined in (2.2), the model point $\mathbf{y}_{j^*} \in \mathcal{Y}$ is closest to $(\mathbf{R}_\mathbf{r}\mathbf{x}_i + \mathbf{t})$. Let $\mathbf{y}_{j_0^*}$ be the closest model point to $\mathbf{R}_{\mathbf{r}_0}\mathbf{x}_i + \mathbf{t}_0$. Observe that, $\forall(\mathbf{r}, \mathbf{t}) \in (C_r \times C_t)$,

$$\begin{aligned}
&e_i(\mathbf{R}_\mathbf{r}, \mathbf{t}) \\
&= \|\mathbf{R}_\mathbf{r}\mathbf{x}_i+\mathbf{t}-\mathbf{y}_{j^*}\| && (2.16)\\
&= \|(\mathbf{R}_{\mathbf{r}_0}\mathbf{x}_i+\mathbf{t}_0-\mathbf{y}_{j^*})+(\mathbf{R}_\mathbf{r}\mathbf{x}_i-\mathbf{R}_{\mathbf{r}_0}\mathbf{x}_i)+(\mathbf{t}-\mathbf{t}_0)\| && (2.17)\\
&\geqslant \|\mathbf{R}_{\mathbf{r}_0}\mathbf{x}_i+\mathbf{t}_0-\mathbf{y}_{j^*}\|-(\|\mathbf{R}_\mathbf{r}\mathbf{x}_i-\mathbf{R}_{\mathbf{r}_0}\mathbf{x}_i\|+\|\mathbf{t}-\mathbf{t}_0\|) && (2.18)\\
&\geqslant \|\mathbf{R}_{\mathbf{r}_0}\mathbf{x}_i+\mathbf{t}_0-\mathbf{y}_{j^*}\|-(\gamma_{r_i}+\gamma_t) && (2.19)\\
&\geqslant \|\mathbf{R}_{\mathbf{r}_0}\mathbf{x}_i+\mathbf{t}_0-\mathbf{y}_{j_0^*}\|-(\gamma_{r_i}+\gamma_t) && (2.20)\\
&= e_i(\mathbf{R}_{\mathbf{r}_0}, \mathbf{t}_0)-(\gamma_{r_i}+\gamma_t), && (2.21)
\end{aligned}$$

where (2.17) trivially involves introducing two auxiliary elements $\mathbf{R}_{\mathbf{r}_0}\mathbf{x}$ and $\mathbf{t}_0$, (2.18) follows from the reverse triangle inequality[1], (2.19) is based on the uncertainty radii

[1] $|x+y| = |x-(-y)| \geqslant |x| - |-y| = |x| - |y|$

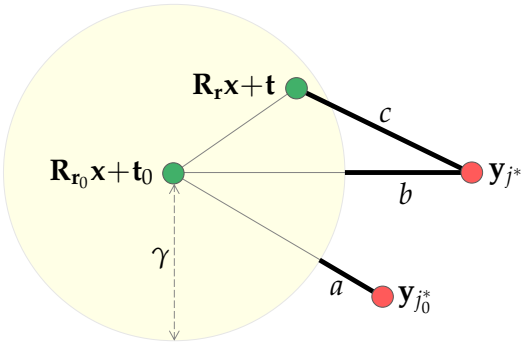


Figure 2.5: Deriving the lower bound. Any transformed data point $\mathbf{R_r x}+\mathbf{t}$ lies within the uncertainty ball (in yellow) centered at $\mathbf{R}_{\mathbf{r}_0}\mathbf{x}+\mathbf{t}_0$ with radius $\gamma = \gamma_r + \gamma_t$. Model points $\mathbf{y}_{j^*}$ and $\mathbf{y}_{j_0^*}$ are closest to $\mathbf{R_r x}+\mathbf{t}$ and $\mathbf{R}_{\mathbf{r}_0}\mathbf{x}+\mathbf{t}_0$ respectively. It is clear that $a \leq b \leq c$ where $a = \underline{e_i}$ and $c = e_i(\mathbf{R_r}, \mathbf{t})$. See text for more details.

in (2.7) and (2.8), and (2.20) is from the closest-point definition. Note that $\mathbf{y}_{j^*}$ is not fixed, but changes dynamically as a function of $(\mathbf{R_r}, \mathbf{t})$ as defined in (2.2).

According to the above derivation, the residual error $e_i(\mathbf{R_r}, \mathbf{t})$ after perturbing a data point $\mathbf{x}_i$ by a 3D rigid motion composed of a rotation $\mathbf{r} \in C_r$ and a translation $\mathbf{t} \in C_t$ will be at least $e_i(\mathbf{R}_{\mathbf{r}_0}, \mathbf{t}_0) - (\gamma_{r_i} + \gamma_t)$. Given that a closest point distance should be non-negative, a valid lower bound $\underline{e_i}$ for $C_r \times C_t$ is $\max\left(e_i(\mathbf{R}_{\mathbf{r}_0}, \mathbf{t}_0) - (\gamma_{r_i} + \gamma_t), 0\right) \leqslant \min_{\forall(\mathbf{r},\mathbf{t}) \in (C_r \times C_t)} e_i(\mathbf{R_r}, \mathbf{t})$. □

The geometric explanation for $\underline{e_i}$ is as follows. Since $\mathbf{y}_{j_0^*}$ is closest to the center $\mathbf{R}_{\mathbf{r}_0}\mathbf{x}_i + \mathbf{t}_0$ of the uncertainty ball with radius $\gamma = \gamma_{r_i} + \gamma_t$, it is also closest to the surface of the ball and $\underline{e_i}$ is the closest distance between point cloud $\mathcal{Y}$ and the ball. Thus, no matter where the transformed data point $\mathbf{R_r x}_i + \mathbf{t}$ lies inside the ball, its closest distance to point cloud $\mathcal{Y}$ will be no less than $\underline{e_i}$. See Figure 2.5 for a geometric illustration.

Summing the squared upper and lower bounds of per-point residuals in (2.14) and (2.15) for all M points, we get the L_2-error bounds in the following corollary.

Corollary 2.1. *(Bounds of L_2 error) For a 3D motion domain $C_r \times C_t$ centered at $(\mathbf{r}_0, \mathbf{t}_0)$ with uncertainty radii γ_{r_i} and γ_t, the upper bound $\overline{E}$ and the lower bound $\underline{E}$ of the optimal L_2 registration error E^* can be chosen as*

$$\overline{E} \doteq \sum_{i=1}^{M} \overline{e_i}^2 = \sum_{i=1}^{M} e_i(\mathbf{R}_{\mathbf{r}_0}, \mathbf{t}_0)^2, \tag{2.22}$$

$$\underline{E} \doteq \sum_{i=1}^{M} \underline{e_i}^2 = \sum_{i=1}^{M} \max\left(e_i(\mathbf{R}_{\mathbf{r}_0}, \mathbf{t}_0) - (\gamma_{r_i} + \gamma_t), 0\right)^2. \tag{2.23}$$

Algorithm 2.1: Go-ICP – the Main Algorithm: BnB search for optimal registration in SE(3)

Input: Data and model points; threshold ϵ; initial cubes $\mathcal{C}_r, \mathcal{C}_t$.
Output: Globally minimal error E^* and corresponding $\mathbf{r}^*, \mathbf{t}^*$.

Put $\mathcal{C}_r$ into priority queue Q_r.
Set $E^* = +\infty$.
loop
  Read out a cube with lowest lower-bound $\underline{E}_r$ from Q_r.
  Quit the loop if $E^* - \underline{E}_r < \epsilon$.
  Divide the cube into 8 sub-cubes.
  foreach sub-cube C_r **do**
    Compute $\overline{E}_r$ for C_r and corresponding optimal $\mathbf{t}$ by calling Algorithm 2.2 with $\mathbf{r}_0$, zero uncertainty radii, and E^*.
    if $\overline{E}_r < E^*$ **then**
      Run ICP with the initialization $(\mathbf{r}_0, \mathbf{t})$.
      Update E^*, $\mathbf{r}^*$, and $\mathbf{t}^*$ with the results of ICP.
    end
    Compute $\underline{E}_r$ for C_r by calling Algorithm 2.2 with $\mathbf{r}_0$, γ_r and E^*.
    if $\underline{E}_r \geqslant E^*$ **then**
      Discard C_r and continue the loop;
    end
    Put C_r into Q_r.
  end
end

2.5 The Go-ICP Algorithm

Now that the domain parametrization and bounding functions have been specified, we are ready to present the Go-ICP algorithm concretely.

2.5.1 Nested BnBs

Given Corollary 2.1, a direct 6D space BnB (i.e., branching each 6D cube into $2^6 = 64$ sub-cubes and bounding errors for them) seems to be straightforward. However, we find it prohibitively inefficient and memory consuming, due to the huge number of 6D cubes and point cloud transformation operations.

Instead, we propose using a nested BnB search structure. An outer BnB searches the rotation space of SO(3) and solves the bounds and corresponding optimal translations by calling an inner translation BnB. In this way, we only need to maintain two queues with significantly fewer cubes. Moreover, it avoids redundant point cloud rotation operations for each rotation region, and takes the advantage that translation operations are computationally much cheaper.

The bounds for both the BnBs can be readily derived according to Section 2.4.2.

Algorithm 2.2: BnB search for optimal translation given rotation

Input: Data and model points; threshold ϵ; initial cube $\mathcal{C}_t$; rotation $\mathbf{r}_0$; rotation uncertainty radii γ_r, so-far-the-best error E^*.
Output: Minimal error E_t^* and corresponding $\mathbf{t}^*$.

Put $\mathcal{C}_t$ into priority queue Q_t.
Set $E_t^* = E^*$.
loop
  Read out a cube with lowest lower-bound $\underline{E}_t$ from Q_t.
  Quit the loop if $E_t^* - \underline{E}_t < \epsilon$.
  Divide the cube into 8 sub-cubes.
  foreach sub-cube C_t **do**
    Compute $\overline{E}_t$ for C_t by (2.26) with $\mathbf{r}_0$,$\mathbf{t}_0$ and γ_r.
    if $\overline{E}_t < E_t^*$ **then**
      Update $E_t^* = \overline{E}_t$, $\mathbf{t}^* = \mathbf{t}_0$.
    end
    Compute $\underline{E}_t$ for C_t by (2.27) with $\mathbf{r}_0$,$\mathbf{t}_0$,γ_r and γ_t.
    if $\underline{E}_t \geqslant E_t^*$ **then**
      Discard C_t and continue the loop.
    end
    Put C_t into Q_t.
  end
end

In the outer rotation BnB, for a rotation cube C_r the bounds can be chosen as

$$\overline{E}_r = \min_{\forall \mathbf{t} \in \mathcal{C}_t} \sum_i e_i(\mathbf{R}_{\mathbf{r}_0}, \mathbf{t})^2, \tag{2.24}$$

$$\underline{E}_r = \min_{\forall \mathbf{t} \in \mathcal{C}_t} \sum_i \max\left(e_i(\mathbf{R}_{\mathbf{r}_0}, \mathbf{t}) - \gamma_{ri}, 0\right)^2, \tag{2.25}$$

where $\mathcal{C}_t$ is the initial translation cube. To solve the lower bound $\underline{E}_r$ in (2.25) with the inner translation BnB, the bounds for a translation cube C_t can be chosen as

$$\overline{E}_t = \sum_i \max\left(e_i(\mathbf{R}_{\mathbf{r}_0}, \mathbf{t}_0) - \gamma_{ri}, 0\right)^2, \tag{2.26}$$

$$\underline{E}_t = \sum_i \max\left(e_i(\mathbf{R}_{\mathbf{r}_0}, \mathbf{t}_0) - (\gamma_{ri} + \gamma_t), 0\right)^2. \tag{2.27}$$

By setting all the rotation uncertainty radii γ_{ri} in (2.26) and (2.27) to zero, the translation BnB solves $\overline{E}_r$ in (2.24). A detailed description is given in Algorithm 2.1 and Algorithm 2.2.

Search Strategy and Stop Criterion. In both BnBs, we use a best-first search strategy. Specifically, each of the BnBs maintains a priority queue; the priority of a cube is opposite to its lower bound. Once the difference between so-far-the-best error E^*

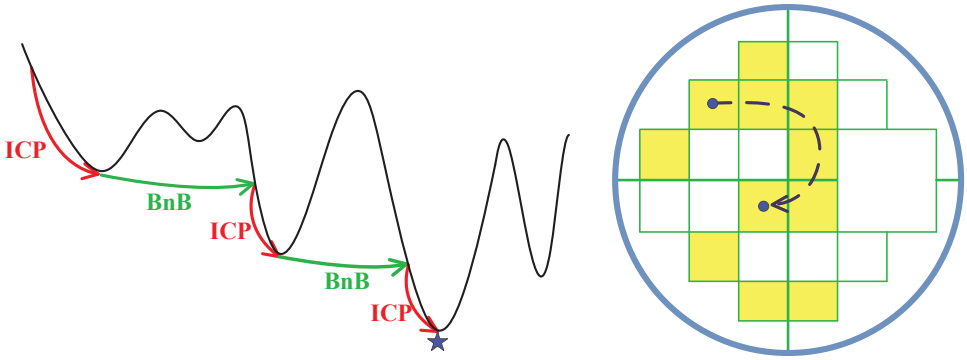


Figure 2.6: Collaboration of BnB and ICP. **Left**: BnB and ICP collaboratively update the upper bounds during the search process. **Right**: with the guidance of BnB, ICP only explores un-discarded, promising cubes with small lower bounds marked up by BnB.

and the lower bound $\underline{E}$ of the current cube is less than a threshold ϵ, the BnB stops. Another possible strategy is to set $\epsilon = 0$ and terminate the BnBs when the remaining cubes are sufficiently small.

2.5.2 Integration with the ICP Algorithm

Lines 10–11 of Algorithm 2.1 show that whenever the outer BnB finds a cube C_r that has an upper bound lower than the current best function value, it will call conventional ICP, initialized with the center rotation of C_r and the corresponding best translation.

Figure 2.6 illustrates the collaborative relationship between ICP and BnB. Under the guidance of global BnB, ICP converges into local minima one by one, with each local minimum having a lower error than the previous one, and ultimately reaches the global minimum. Since ICP monotonically decreases the current-best error E^* (cf. [Besl and McKay, 1992]), the search path of the local ICP is confined to un-discarded, promising sub-cubes with small lower bounds, as illustrated in Figure 2.6.

In this way, the global BnB search and the local ICP search are intimately integrated in the proposed method. The former helps the latter jump out of local minima and guides the latter's next search; the latter accelerates the former's convergence by refining the upper bound, hence improving the efficiency.

2.5.3 Outlier Handling with Trimming

In statistics, trimming is a strategy to obtain a more robust statistic by excluding some of the extreme values. It is used in Trimmed ICP [Chetverikov et al., 2005] for robust point cloud registration. Specifically, in each iteration, only a subset $\mathcal{S}$ of the data points that have the smallest closest distances are used for motion computation.

Therefore, the registration error will be

$$E^{Tr} = \sum_{i \in \mathcal{S}} e_i(\mathbf{R}, \mathbf{t})^2. \tag{2.28}$$

To robustify our method with trimming, it is necessary to derive new upper and lower bounds of (2.28). We have the following result.

Corollary 2.2. *(Bounds of the trimmed L_2 error) The upper bound $\overline{E^{Tr}}$ and lower bound $\underline{E^{Tr}}$ of the registration error with trimming for the domain $C_r \times C_t$ can be chosen as*

$$\overline{E^{Tr}} \doteq \sum_{i \in \mathcal{P}} \overline{e_i}^2, \tag{2.29}$$

$$\underline{E^{Tr}} \doteq \sum_{i \in \mathcal{Q}} \underline{e_i}^2. \tag{2.30}$$

where $\overline{e_i}$, $\underline{e_i}$ are bounds of the per-point residuals defined in (2.14), (2.15) respectively, and $\mathcal{P}$, $\mathcal{Q}$ are the trimmed point sets having smallest values of $\overline{e_i}$, $\underline{e_i}$ respectively, with $|\mathcal{P}| = |\mathcal{Q}| = |\mathcal{S}| = K$.

Proof: The upper bound in (2.29) is chosen trivially. To see the validity of the lower bound in (2.30), observe that $\forall(\mathbf{r}, \mathbf{t}) \in C_r \times C_t$,

$$\underline{E^{Tr}} = \sum_{i \in \mathcal{Q}} \underline{e_i}^2 \leq \sum_{i \in \mathcal{S}} \underline{e_i}^2 \leq \sum_{i \in \mathcal{S}} e_i(\mathbf{R_r}, \mathbf{t})^2 = E^{Tr}. \tag{2.31}$$

□

Based on this corollary, the corresponding bounds in the nested BnB can be readily derived. As proved in [Chetverikov et al., 2005], iterations of Trimmed ICP decrease the registration error monotonically to a local minimum. Thus it can be directly integrated into the BnB procedure.

Fast Trimming. A straightforward yet inefficient way to do trimming is to sort the residuals outright and use the K smallest ones. In this work, we employ the Introspective Selection algorithm [Musser, 1997] which has $O(N)$ performance in both the worst case and average case.

Other Robust Extensions. In the same spirit as trimming, other ICP variants such as [Champleboux et al., 1992; Masuda and Yokoya, 1994] can be handled. The method can also be adapted to LM-ICP [Fitzgibbon, 2003], where the new lower-bound is simply a robust kernelized version of the current one. It may also be extended to ICP variants with L_p-norms [Bouaziz et al., 2013], such as the robustness-promoting L_1-norm.

2.6 Experiments

We implemented the method in C++ and tested it on a standard PC with an Intel i7 3.4GHz CPU. In the experiments reported below, the point clouds were pre-normalized such that all the points were within the domain of $[-1,1]^3$. Although the goal was to minimize the L_2 error in (2.1), the root-mean-square (RMS) error is reported for better comprehension.

Closest-point distance computation. To speed up the closest distance computation, a kd-tree data structure can be used. We also provide an alternative solution that is used more often in the experiments – a 3D Euclidean Distance Transform (DT) [Fitzgibbon, 2003] used to compute closest distances for fast bound evaluation[2]. A DT approximates the closest-point distances in the real-valued space by distances of uniform grids, and pre-computes them for constant-time retrieval (details about our DT implementation can be found in the XXXXXXXXXXXXXXXXXXXX). Despite the DT can introduce approximation errors thus the convergence gap may not be exactly ϵ, in the following experiments our method works very well with a $300\times300\times300$ DT for optimal registration. Naturally, higher resolutions can be used when necessary.

2.6.1 Optimality

To verify the correctness of the derived bounds and the optimality of Go-ICP, we first use a convergence condition similar to [Hartley and Kahl, 2009] for the BnBs. Specifically, we set the threshold of a BnB to be 0 and specify a smallest cube size at which the BnB stops dividing a cube. In this way, we can examine the uncertainty in the parameter space after the BnB stops. Both the DT and kd-tree are tested in these experiments.

2.6.1.1 Synthetic Points

We first tested the method on a synthetically generated scene with simple objects. Specifically, five 3D shapes were created: an irregular tetrahedron, a cuboid with three different side-lengths, a regular tetrahedron, a regular cube, and a regular octahedron. Note that the latter 4 shapes have self-symmetries. All the shapes were then placed together, each with a random transformation, to generate clustered scenes. Zero-mean Gaussian noise with standard deviation $\sigma=0.01$ was added to the scene points. We created such a scene as shown in Figure 2.7, and applied Go-ICP to register the vertices of each shape to the scene points.

To test the rotation BnB, we set the parameter domain to be $[-\pi,\pi]^3\times[-1,1]^3$ and the minimal volume of a rotation cube to 1.5E$-$5 ($\sim$1 degree uncertainty). The lower bound of a rotation cube was set to be the global lower bound of the invoked

[2]Local ICP is called infrequently so we simply use a kd-tree for it. The refined upper-bounds from the found ICP solutions are evaluated via the DT for consistency.

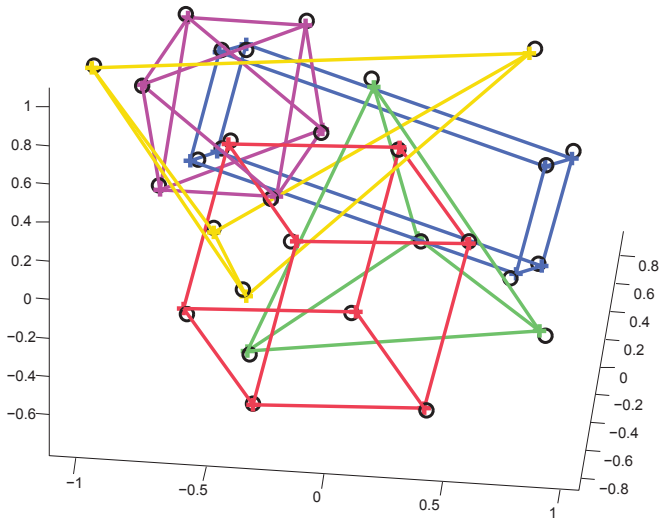


Figure 2.7: A clustered scene (black circles) and the registration results of Go-ICP for the five shapes.

translation BnB. Thus the threshold of translation BnB is not very important and we set it to a small value ($0.0001 \times N$ where N is the data point number). The initial errors E_t^* of translation BnBs were set to infinity.

In all tests, Go-ICP produced correct results with both the DT and kd-tree. The remaining rotation cubes using the DT and kd-tree respectively are almost visually indistinguishable, and Figure 2.8 shows the results using the DT. It is interesting to see that the remaining cubes formed 1 cluster for the irregular tetrahedron, 4 clusters for the cuboid, 12 clusters for the regular tetrahedron, and 24 clusters for the regular cube and octahedron. These results conform to the geometric properties of these shapes and validated the derived bounds. Investigating shape self-similarity would be a practical application of the algorithm. Moreover, Figure 2.9 shows some typical remaining rotation domains on 2D slices of the rotation π-ball[3]. The non-convexity of the problem can be clearly seen from the presence of many local minima. It can also be seen that the remaining rotation domains using a DT and kd-tree are highly consistent, and the optima are well contained by them.

The translation BnB can be easily verified by running it with rotations picked from the remaining rotation cubes. The threshold was set to be 0, and the minimal side-length of a translation cube was set to be 0.01. The last figure of Figure 2.8 shows a typical result.

[3]We chose the slices passing two randomly-selected optimal rotations plus the origin. Due to shape symmetry, there may exist more than two optimal rotations on one slice.

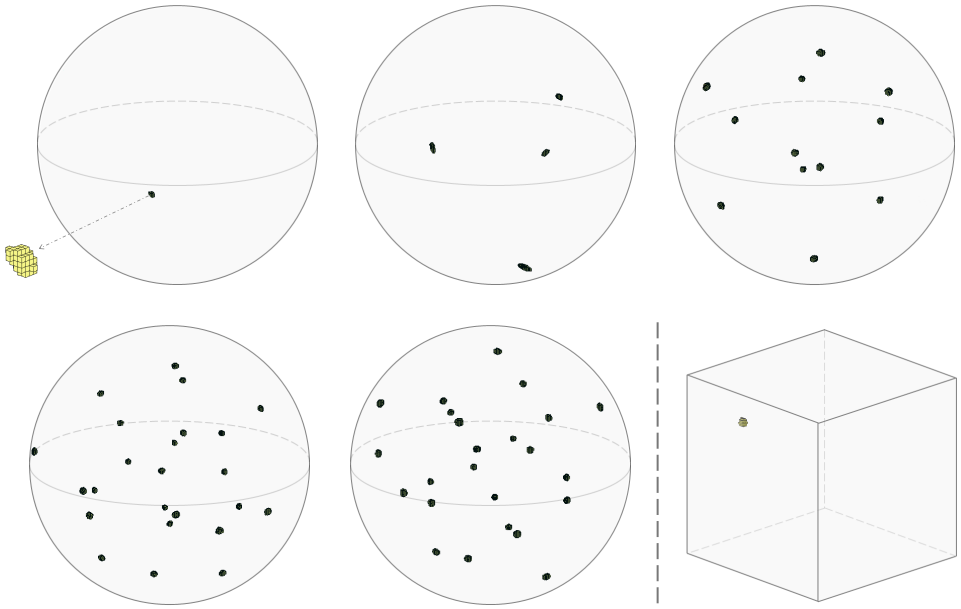


Figure 2.8: Remaining cubes of BnBs. The first five figures show the remaining cubes in the rotation π-ball of the rotation BnBs, for an irregular tetrahedron, a cuboid with three different side-lengths, a regular tetrahedron, a regular cube, and a regular octahedron respectively. The last figure shows a typical example of remaining cubes of a translation BnB, for the irregular tetrahedron. (Best viewed when zoomed in)

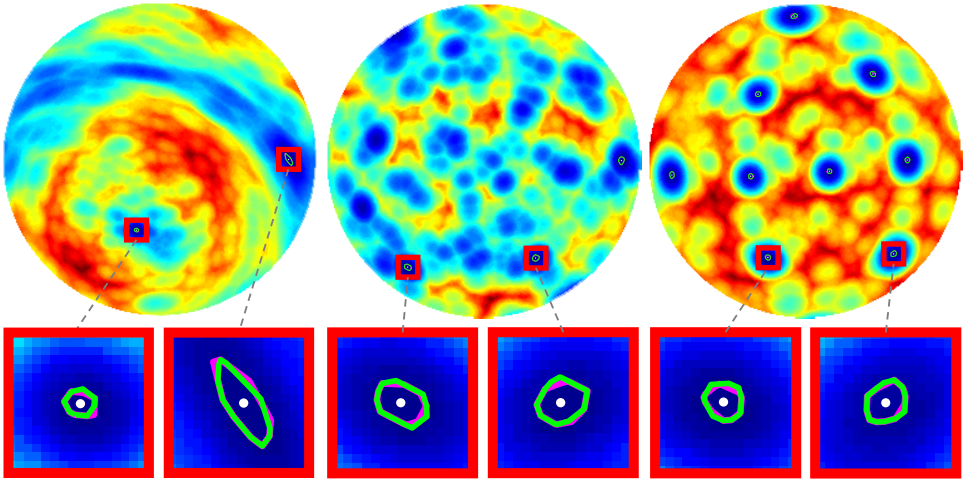


Figure 2.9: Remaining rotation domains of the outer rotation BnB on 2D slices of the π-ball, for the synthetic points. Results using the DT and the kd-tree are within magenta and green polygons, respectively. The white dots denote optimal rotations. From left to right: a cuboid, a regular tetrahedron and a regular cube. The colors on the slices indicate registration errors evaluated via inner translation BnB: red for high error and blue for low error. (Best viewed when zoomed in)

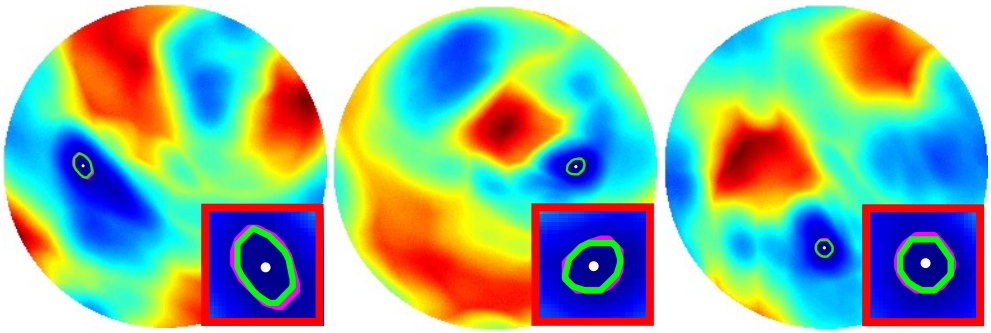


Figure 2.10: Remaining rotation domains of the outer rotation BnB on 2D slices of the π-ball, for the bunny point clouds. The three slices pass through the optimal rotation and the X-, Y-, Z-axes respectively. See also the caption of Figure 2.9. (Best viewed when zoomed in)

2.6.1.2 Real Data

Similar experiments were conducted on real data. We applied our method to register a bunny scan bun090 from the Stanford 3D dataset[4] to the reconstructed model. Since the model and data point clouds are of similar spatial extents, we set the parameter domain to be $[-\pi, \pi]^3 \times [-0.5, 0.5]^3$ which is large enough to contain the optimal solution. We randomly sampled 500 data points and did similar tests to those on the synthetic points. The translation BnB threshold was set to $0.001 \times N$, and the remaining rotation cubes from the outer rotation BnB were similar to the first figure in Figure 2.8 (i.e., , one cube cluster). Figure 2.10 shows the results on three slices of the rotation π-ball.

Additionally, we recorded the bound and cube evolutions in the rotation BnB which are presented in Figure 2.11. It can be seen that BnB and ICP collaboratively update the global upper bound. Corresponding transformations for each global upper bound found by BnB and ICP are shown in Figure 2.12. Note that in the fourth image the pose has been very close to the optimal one, which indicates that ICP may fail even if reasonably good initialization is given.

Although the convergence condition used in this section worked successfully, we found that using a small threshold ϵ of the bounds to terminate a BnB also works well in practice. It is more efficient and produces satisfactory results. In the following experiments, we used this strategy for the BnBs.

2.6.2 "Partial" to "Full" Registration and Camera Global Motion Estimation

If a global, full point cloud model of an object or a scene is known a prior, then given a partial point cloud of it scanned by a 3D camera, the global motion of the camera can be estimated by registering the partial point cloud onto the full point

[4] http://graphics.stanford.edu/data/3Dscanrep/

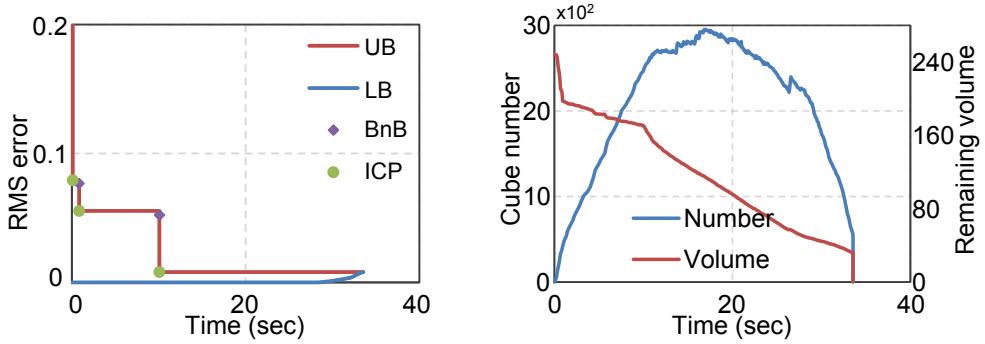


Figure 2.11: Evolution of the bounds (left) and cubes (right) in the rotation BnB with a DT on the bunny point-sets. See text for details.

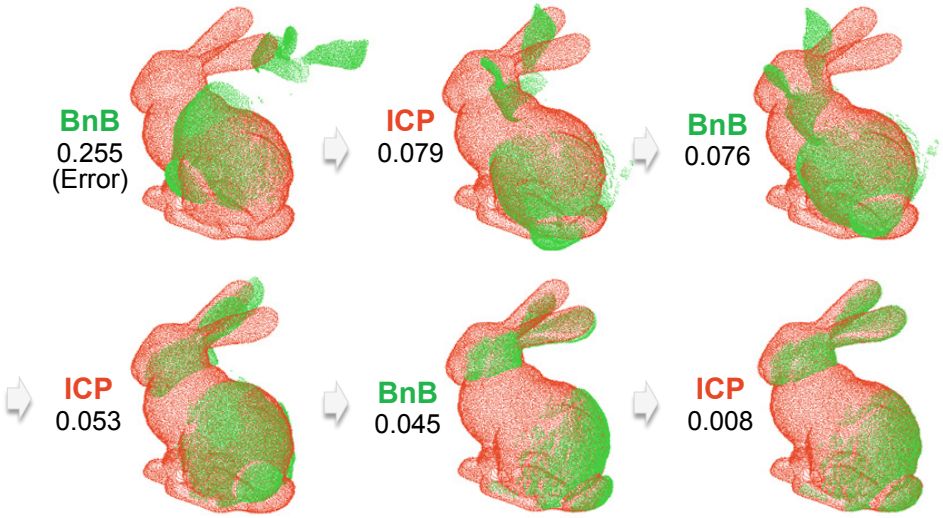


Figure 2.12: Evolution of Go-ICP registration for the bunny dataset. The model point cloud and data point cloud are shown in red and green respectively. BnB and ICP collaboratively update the registration: ICP refines the solution found by BnB and BnB guides ICP into the convergence basins of multiple local minima with increasingly lower registration errors.

cloud model. In this section, we will use such partial and full point clouds to test the proposed 3D registration algorithm and estimate camera global motion. We will first analyze the performance of the proposed method on point clouds of some relatively small objects, then will perform camera global motion estimation in relatively large scenes.

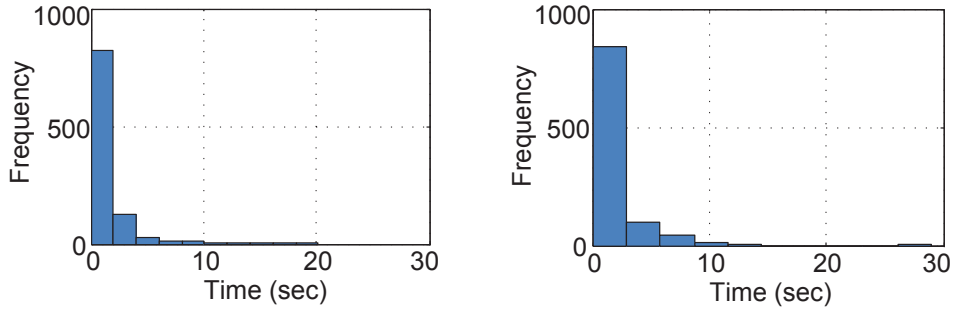


Figure 2.13: Running time histograms of Go-ICP with DTs for the bunny (left) and dragon (right) point clouds.

2.6.2.1 Performance Analysis with Small Objects

The bunny and dragon models from the Stanford 3D dataset were used to test the performance of Go-ICP by registering partially scanned point clouds to full 3D model point clouds. All 10 partial scans of the bunny dataset were used as data point clouds. For the dragon model, we selected 10 scans generated from different viewpoints as data point clouds. The reconstructed bunny and dragon models were used as model point clouds.

For each of these 20 scans, we first performed 100 tests with random initial rotations and translations. The transformation domain to explore for Go-ICP was set to be $[-\pi, \pi]^3 \times [-0.5, 0.5]^3$. We sampled $N = 1000$ data points from each scan, and set the convergence threshold ϵ to be $0.001 \times N$.

As expected, *Go-ICP achieved 100% correct registration on all the 2 000 registration tasks on the bunny and dragon models*, with both the DT and kd-tree. All rotation errors were less than 2 degrees and all translation errors were less than 0.01. With a DT, the mean/longest running times of Go-ICP, in the 1 000 tests on 1 000 data points and 20 000–40 000 model points, were 1.6s/22.3s for bunny and 1.5s/28.9s for dragon. Figure 2.13 shows the running time histograms The running times with a kd-tree were typically 40–50 times longer than that with the DT. The solutions from using the DT and the kd-tree respectively were highly consistent (the largest rotation difference was below 1 degree).

We then analyzed the running time of the proposed method under various settings using the DT. We analyzed the influence of each factor by varying it while keeping others fixed. Default factor settings: number of data points $N = 1000$, no added Gaussian noise (i.e., standard deviation $\sigma = 0$) and convergence threshold $\epsilon = 0.001 \times N$.

Effect of Number of Points. In this experiment, the running time was tested for different numbers of points. Since the DT was used for closest-point distance retrieval, the number of model points does not significantly affect the speed of our

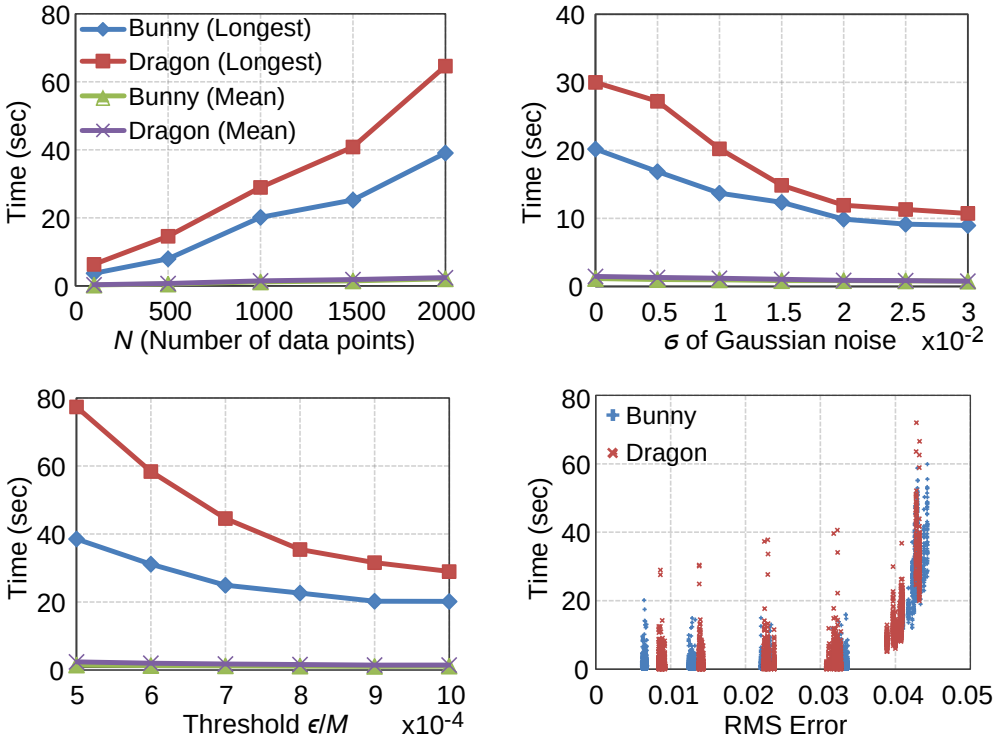
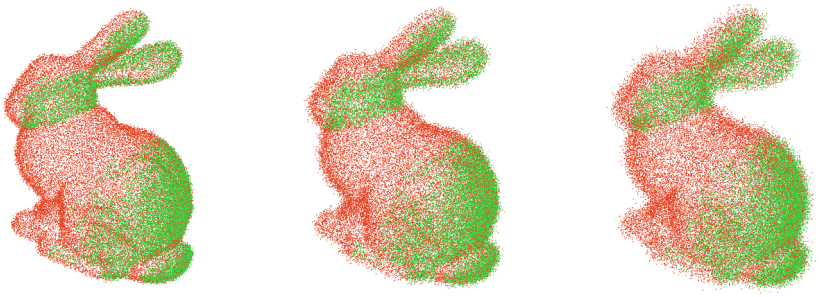


Figure 2.14: Running time of the Go-ICP method with DTs on the bunny and dragon point clouds with respect to different factors. The evaluation was conducted on 10 data point clouds with 100 random poses (i.e., , 1 000 pairwise registrations).

method. To test the running time with respect to different numbers of data points, we randomly sampled the data point clouds. As presented in Figure 2.14, the running time manifested a linear trend since closest-point distance retrieval was $O(1)$ and the convergence threshold varied linearly with the number of data points.

Effect of Noise. We examined how the noise level impacted the running time by adding Gaussian noise to both the data and model point clouds. The registration results on the corrupted bunny point clouds are shown in Figure 2.15. We found that, as shown in Figure 2.14, the running time decreased as the noise level increased (until $\sigma = 0.02$). This is because the Gaussian noise (especially that added to the model points) smoothed out the function landscape and widened the convergence basin of the global minimum, which made it easier for Go-ICP to find a good solution.

Effect of Convergence Threshold. We further investigated the running time with respect to the convergence threshold of the BnB loops. We set the threshold ϵ to depend linearly on N, since the registration error is a sum over the N data points. Figure 2.14 shows that the smaller the threshold is, the slower our method performs. In our experiments, $\epsilon = 0.001 \times N$ was adequate to get a 100% success rate for the



$\sigma = 0.01$ $\sigma = 0.02$ $\sigma = 0.03$

Figure 2.15: Registration with different levels of Gaussian noise.

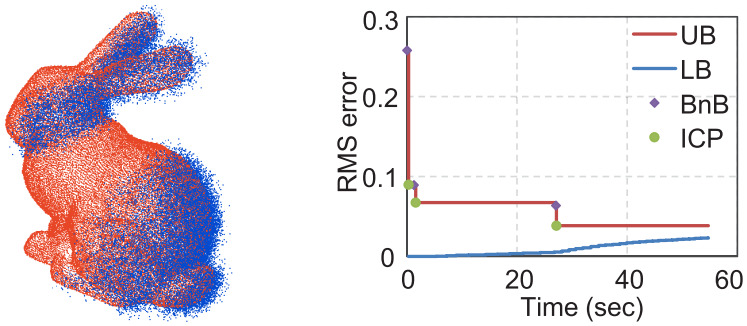


Figure 2.16: Registration with high optimal error. **Left:** Gaussian noise was added to the data point cloud to increase the RMS error. **Right:** the global minimum was found at about 25s with a DT; the remainder of the time was devoted solely to increasing the lower bound.

bunny and dragon point clouds. For cases when the local minima are small or close to the global minimum, the threshold can be set smaller.

Effect of Optimal Error. We also tested the running time *w.r.t.* the optimal registration error. To increase the error, Gaussian noise was added to the data point cloud *only*. As shown in Figure 2.14, the running time remained almost constant when the RMS error was less than 0.03. This is because the gap between the global lower bound and the optimal error was less than ϵ. Therefore, the running time depended primarily on when the global minimum was found, that is, the termination depended on the *decrease of the upper bound*. However, it takes longer to converge if the final RMS error is higher. Figure 2.16 shows the bounds evolution for bunny when the RMS error was increased to ~ 0.04. As can be seen, the global minimum was found at about 25s, with the remainder of the time devoted to *increasing the lower bound*.

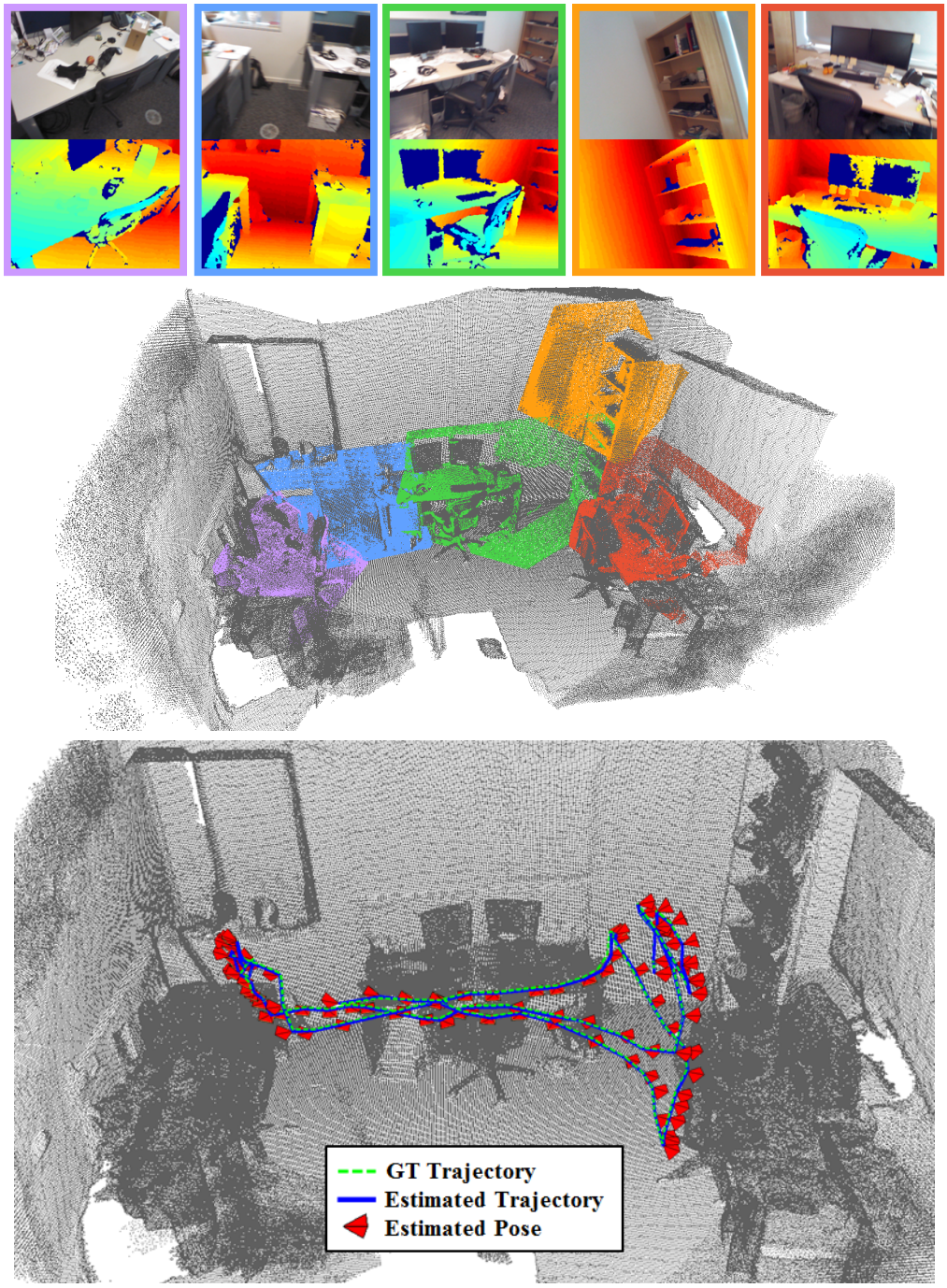


Figure 2.17: Camera localization experiment. **Top**: 5 (out of 100) color and depth image pairs of the scene. (The color images were not used) **Middle**: Corresponding registration results. Note that the scene contains many similar structures, and the depth images only cover small portions of the scene, which make the 3D registration tasks very challenging. **Bottom**: Ground truth camera trajectories and the estimated results. (We did not estimate the relative motion between two cameras; the camera locations are connected with lines for better visualization)

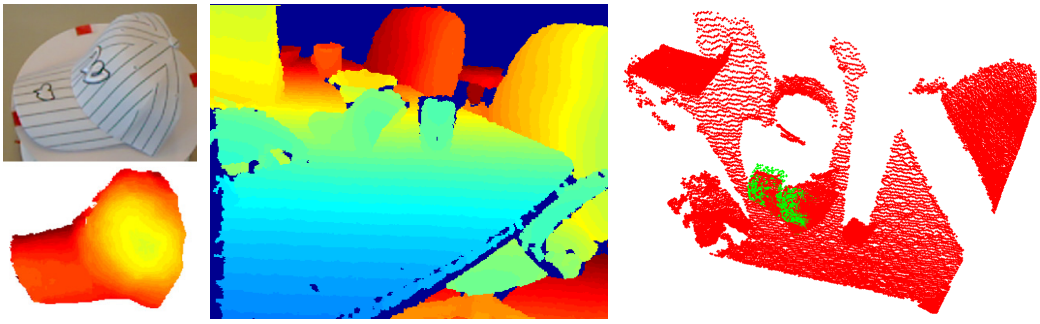


Figure 2.18: 3D object localization experiment. **Left**: a labelled object and its depth image to generate the data point cloud. **Middle**: a scene depth image to generate the model point cloud. **Right**: the registration result.

2.6.2.2 Camera Global Motion Estimation in a Large Scene

As previously mentioned, the global motion of the camera can be estimated by registering the partial point cloud onto the full point cloud model. This section tests the proposed algorithm in such scenarios. In the following experiments, the transformation domain for exploration was set to be $[-\pi, \pi]^3 \times [-1, 1]^3$.

We first tested our method on one sequence of the camera localization dataset [Shotton et al., 2013]. The size of the office is approximately $5.5m^2$, and the sequence contains 1 000 depth images taken by a Kinect depth camera moving smoothly in the office over different locations. Figure 2.17 shows a sample color and depth image pair, and a 3D model of the office scene. Note that the scene contains many similar structures, and the depth images only cover small portions of the scene. Our goal was to estimate the camera poses by registering the point clouds of the depth images onto the 3D scene model, which were challenging tasks. We evenly sampled the sequence to 100 depth images. Each depth image was then evenly sampled to $400 \sim 600$ points. We set our method to seek a solution with the registration error smaller than $0.0001 \times N$, and the method registered the 100 point clouds with the mean/longest running time of 32s/178s using a DT. The rotation errors and translation errors were all below 5 degrees and 10cm. Five typical results are presented in Figure 2.17. The last row of Figure 2.17 compares the ground-truth camera trajectories provided by the dataset and the results by our method. It can be seen that the trajectory from out method conforms quite well to the ground truth.

To further test the proposed method, we used the RGB-D Object Dataset [Lai et al., 2011] to perform the object localization experiment in a relatively large scene. As shown in Figure 2.18, a baseball cap was used as the object being localized and its depth image was taken to generate data point cloud, and a scene depth image was used to generate the model point cloud. Since the two depth images are taken from different views points, the model point clouds only cover a part of the data point cloud. Therefore, we need to handle the outliers in the data points whose correspondences are missing, and trimming was employed as described in Section 2.5.3.

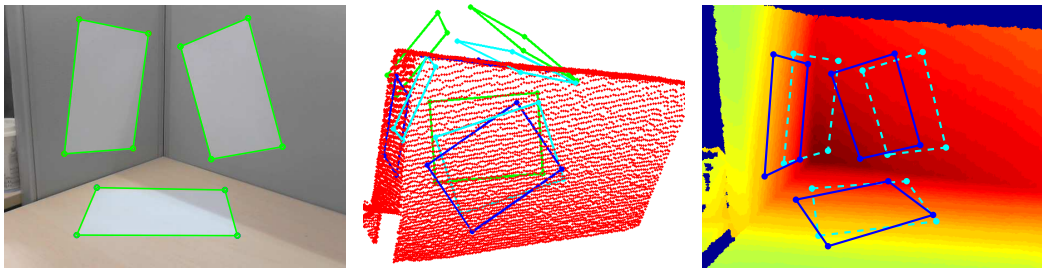


Figure 2.19: Sparse-to-dense 3D point cloud registration. **Left**: the color image with extracted line segments for single view 3D reconstruction. **Middle**: the initial 3D registration (in green), the result of ICP (in cyan) and the result of Go-ICP (in blue) (the lines are for visualization purposes only). **Right**: the depth image with a projection of the registered 3D points from ICP (in cyan) and Go-ICP (in blue).

We randomly sampled $N = 100$ points from the cap model, and set the trimming percentage and threshold to be $\rho = 10\%$ and $\epsilon = 0.00003 \times K$ respectively. Go-ICP successfully registered the cap point cloud in 42 seconds with a DT. It can be seen from Figure 2.18 that the pose of the cap accurately estimated.

The proposed method can also be used to register a sparse point cloud from, e.g., the reconstructed feature points from a 2D color camera. Figure 2.19 shows an example where 12 points are reconstructed, and the goal was to register them onto a dense point cloud from a depth camera. We found that ICP often failed to find the correct registration when the pose difference between the cameras was reasonably large. To the best of our knowledge, few methods can perform such *sparse-to-dense registration* reliably without human intervention, due to the difficulty of building putative correspondences. Setting ϵ to be $0.00001 \times N$, Go-ICP with a DT found the optimal solution in less than 1s. Note that the surfaces are not exactly perpendicular to each other.

2.6.3 "Partial" to "Partial" Registration and Camera Relative Motion Estimation

In this section, we tested the proposed method on partially overlapping point clouds. The data points in regions that are not overlapped by the other model point cloud should be treated as outliers, as their correspondences are missing. Trimming was employed to deal with outliers.

We used 10 point cloud pairs shown in Figure 2.20 to test Go-ICP with trimming. These point clouds were generated by different scanners and with different noise levels. The bunny, dragon and buddha models are from the Standford 3D dataset. The chef and dinosaur models are from [Mian et al., 2006]. The denture was generated with a structured light 3D scanner[5]. The owl status is from [Bouaziz et al., 2013] and

[5]http://www.david-3d.com/en/support/downloads

Figure 2.20: Registration with partial overlap. Go-ICP with the trimming strategy successfully registered the 10 point cloud pairs with 100 random relative poses for each of them. The point clouds in red and blue are denoted as point cloud A and point cloud B, respectively. The trimming settings and running times are presented in Table 2.1.

Table 2.1: Running time (in seconds) of Go-ICP with DTs for the registration of the partially overlapping point clouds in Figure 2.20. 100 random relative poses were tested for each point cloud pair and 1 000 data points were used. ρ is the trimming percentage.

	$A \rightarrow B$		$B \rightarrow A$	
	ρ	mean/max time	ρ	mean/max time
Bunny	10%	0.81 / 10.7	10%	0.49 / 7.25
Dragon	20%	2.99 / 43.5	40%	8.72 / 72.4
Buddha	10%	0.71 / 11.3	10%	0.60 / 14.8
Chef	20%	0.45 / 4.47	30%	0.52 / 3.79
Dinosaur	10%	2.03 / 23.5	10%	1.65 / 26.1
Owl	40%	12.5 / 87.5	40%	13.4 / 75.0
Denture	30%	6.74 / 74.7	30%	4.24 / 68.1
Room	30%	9.82 / 73.3	30%	18.4 /107.3
Bowl	20%	3.19 / 20.3	30%	3.52 / 25.3
Loom	30%	8.64 / 67.2	20%	5.96 / 44.6

the room scans are from [Shotton et al., 2013]. The bowl and loom point clouds were collected by us with a Kinect. The overlapping ratios of the point cloud pairs are between 50% $\sim$ 95%.

For each of the 10 point cloud pairs, we generated 100 random relative poses and registered the two point clouds to each other. This lead to 2 000 registration tasks. The translation domain to explore for Go-ICP was set to be $[-\pi, \pi]^3 \times [-0.5, 0.5]^3$.

We chose the trimming percentages ρ as in Table 2.1, sampled $N = 1000$ data points for each registration, and set all the convergence thresholds to $\epsilon = 0.001 \times K$ where $K = (1 - \rho) \times N$. Our method correctly registered the point clouds in all these tasks. All the rotation errors were less than 5 degrees and translation errors were less than 0.05 compared to the manually-set ground truths. The running times using DTs are presented in Table 2.1. In general, it takes the method a longer time compared to the outlier-free case due to 1) the emergence of additional local minima induced by the outliers and 2) the time-consuming trimming operations.

Choosing trimming percentages. In these experiments, each parameter ρ was chosen by visually observing the two point clouds and roughly guessing their non-overlapping ratios. The results were not very sensitive to ρ (e.g., setting ρ as 5%, 10% and 20% all led to a successful registration for bunny). If no rough guess is available, one can gradually increase ρ until a measure such as the inlier number or RMS error attains a set value, or apply the automatic overlap estimation proposed in [Chetverikov et al., 2005].

2.7 Conclusion

We have introduced a globally optimal solution to Euclidean registration in 3D, under the L_2-norm closest-point error metric originally defined in ICP. The method is based on the Branch-and-Bound (BnB) algorithm, thus global optimality is guaranteed regardless of the initialization. The key innovation is the derivation of registration error bounds based on the $\mathrm{SE}(3)$ geometry.

The proposed Go-ICP algorithm is especially useful when an exactly optimal solution is highly desired or when a good initialization is not reliably available. For practical scenarios where real-time performance is not critical, the algorithm can be readily applied or used as an optimality benchmark.

In the future work, we would like to investigate tighter bounds to further improve the efficiency. We also plan to test other outlier handling strategies such as those mentioned in Section 2.5.3).

Chapter 3

2D Camera Motion Estimation via Optimal Inlier-set Maximization

2D Color camera relative pose estimation, or essential matrix estimation, is a basic building block for Structure from Motion (SfM). Given two views of a rigid scene from a calibrated perspective camera, the task is to estimate the relative pose or motion between the two views. Essential matrix can be estimated with image point correspondences using epipolar geometry. In reality, correspondence outliers are ubiquitous. For instance, natural or man-made scenes often contain similar structures, flat (and ambiguous) regions, repetitive patterns *etc.*, making flawless feature matching nearly impossible.

To deal with outliers in the context of multiple-view geometry, RANSAC [Fischler and Bolles, 1981] and its variants have played a major role. These methods, which are based on random sampling, cannot provide an optimality guarantee, and the inlier sets they find often vary from time to time. Moreover, in most RANSAC algorithms (*e.g.* [Goshen and Shimshoni, 2008; Raguram et al., 2013]), to ensure efficiency, an algebraic solver (*e.g.* the 5-point method [Nistér, 2004; Li and Hartley, 2006]) and the 8-point method [Longuet-Higgins, 1981; Hartley, 1997]) is often adopted to compute tentative estimation, followed by a thresholding stage using geometric reprojection error or Sampson error. The apparent inconsistency here, *i.e.* algebraic solver versus geometric threshold, can lead to inferior estimate.

In contrast, this chapter seeks a consistent, and globally optimal solution to essential matrix estimation, based on meaningful geometric error. By optimal, we adopt the consensus set maximization idea of RANSAC, *i.e.* to find the maximal-sized inlier set that is compatible with the input image measurements. To distinguish inliers from outliers, we use angular reprojection error. With a calibrated camera, it is natural to use angular reprojection error, because a calibrated camera behaves just like an angle measurement device, and every image point (represented by a unit vector) gives the actual viewing angle.

To achieve globally maximal inlier-set, a naive way would be exhaustively enumerating all possible combinations of inliers/outliers. However, this soon becomes intractable as combinations grow exponentially with point number. No efficient solver to this combinational problem exists to our knowledge. Our idea in this chap-

ter is: rather than searching over all *discrete* combinations of inliers, we search the entire *continuous* parameter space of essential matrices. To this end, it is necessary to find a suitable domain representation (parametrization) of the space, with which the bounds can be easily derived and efficiently evaluated.

The proposed method is based on systematically searching two (reduced) rotation spaces using branch-and-bound (BnB). It is inspired by the rotation search technique proposed by Hartley and Kahl [2007], which has been used in several vision problems [Heller et al., 2012; Bazin et al., 2012; Yang et al., 2013b]. To minimize the L_∞-norm of angular errors, Hartley and Kahl [2007] uses BnB to recursively search $\mathrm{SO}(3)$ with elegant bounding. However, L_∞-optimization is known to be extremely vulnerable to outliers, and Hartley and Kahl [2007] assumes outlier-free correspondences. In contrast, our method works in the presence of outliers.

3.1 Related Work

Our method is closely related to [Hartley and Kahl, 2007], and extends [Hartley and Kahl, 2007] to optimal inlier-set maximization which is non-trivial. A key insight for [Hartley and Kahl, 2007] to applying rotation search to essential matrix estimation is that, given rotation, the translation can be optimally solved with convex optimization (SOCP/LP). However, optimally solving the translation maximizing inlier-set cardinality is not trivial. The optimal essential matrix problem considered here is more challenging. Method of Bazin et al. [2012] achieves inlier-set maximization with rotation search, however translation is assumed to be known. In contrast, we optimally solve the problem by searching the essential manifold with BnB, based on a novel parametrization scheme.

There have been some research efforts devoted to optimal essential matrix estimation with inlier-set maximization criterion [Enqvist et al., 2011; Enqvist and Kahl, 2009]. Most closely related to our method is [Enqvist et al., 2011] in which a brute-force search method is proposed using triangulation feasibility test. The solution is exhaustively searched over the discretized parameter space formed by two unit spheres, and GPU implementation is used to speed up the computation. In [Enqvist and Kahl, 2009], double pairs of correspondences are used, from which camera pose is found by searching the two epipoles via BnB. An approximation is made to solve an otherwise NP-hard problem (minimum vertex cover), which compromises the global optimality guarantee. The closed-form bounding functions we use in this chapter are inspired by [Enqvist et al., 2011] (with necessary extension); however, we introduce other innovations in both parametrization scheme and optimization technique. By our method, an exact optimality can be achieved.

Some approaches use branch-and-bound methods for finding globally optimal fundamental matrix [Li, 2009; Zheng et al., 2011]. In particular, inlier-set is optimally maximized by Li [2009] with an algebraic error. Geometrically meaningful error is investigated by Zheng et al. [2011], but the goal is optimal error minimization assuming no outlier. These works discuss uncalibrated cases only, where the underlying

Euclidean constraints of essential matrices are not exploited.

Another line of related work is outlier removal using convex optimization [Sim and Hartley, 2006; Ke and Kanade, 2007; Li, 2007a; Olsson et al., 2010]. These methods are able to detect potential outliers with respect to a given threshold. However, the goal is not inlier-set maximization and outliers may be removed at the expense of losing some true inliers. Moreover, in SfM they assume known rotation to formulate the problem to be (quasi-)convex. Our work is also related to the study of SfM without pre-built correspondences [Dellaert et al., 2000; Makadia et al., 2007], in a sense that we all compute the motion yielding most agreeable correspondences.

3.2 Essential Manifold Parametrization

A rigid motion comprises rotation and translation. As such, an essential matrix $\mathbf{E}$ relates to a 3D rotation $\hat{\mathbf{R}} \in \mathrm{SO}(3)$ and a 3D translation $\hat{\mathbf{t}} \in \mathbb{R}^3$ from the first camera to the second one by $\mathbf{E} = [\hat{\mathbf{t}}]_\times \hat{\mathbf{R}}$ where $[\cdot]_\times$ denotes the skew-symmetric matrix representation. Essential matrix can only be determined up to an unknown scale. To resolve this scale indeterminacy one can set the length of $\hat{\mathbf{t}}$, *i.e.* $\|\hat{\mathbf{t}}\|$ to be fixed (*e.g.* to be 1). Therefore, we have $\hat{\mathbf{t}} \in \mathbb{S}^2$, *i.e.* a 2-sphere embedded in $\mathbb{R}^3$. In this way the essential manifold can be parameterized with 5 degrees of freedom (dofs) in $\mathrm{SO}(3) \times \mathbb{S}^2$. In this chapter, we advocate different coordinate system and parametrization scheme to facilitate our BnB algorithm.

In solving the relative pose problem, one has the freedom to arbitrarily choose a coordinate system as the world frame. Different from a common practice which sets the first camera matrix to be $[\mathbf{I} \mid \mathbf{0}]$, we fix the first camera's center at the origin, *i.e.* $\mathbf{C} \equiv \mathbf{0}$, and fix the second camera's center at $\mathbf{C}' \equiv [0, 0, 1]^\mathrm{T}$ on the Z-axis.[1] We use $\mathbf{R}$ to denote the *absolute orientation* of the first camera (relative to the world frame), and $\mathbf{R}'$ for the second camera. Then, it is easy to see that, under this configuration the camera relative motion $(\hat{\mathbf{R}}, \hat{\mathbf{t}})$ and the essential matrix can be written as

$$\hat{\mathbf{R}} = \mathbf{R}'\mathbf{R}^\mathrm{T} \tag{3.1}$$

$$\hat{\mathbf{t}} = -\mathbf{R}'\mathbf{C}' = -\mathbf{R}'[0, 0, 1]^\mathrm{T} \tag{3.2}$$

$$\mathbf{E} = [-\mathbf{R}'\mathbf{C}']_\times \mathbf{R}'\mathbf{R}^\mathrm{T} = \mathbf{R}'[-\mathbf{C}']_\times \mathbf{R}^\mathrm{T} = \mathbf{R}' \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{R}^\mathrm{T}. \tag{3.3}$$

Using two absolute rotations $(\mathbf{R}, \mathbf{R}') \in \mathrm{SO}(3) \times \mathrm{SO}(3)$ to represent essential matrix is clearly an over-parametrization, because the essential manifold has only five dofs. The excess one dof can further be removed, as we will show next.

Observe that, under our special camera setup, any rotation about Z-axis (*i.e.* the axis joining the two camera centers) applied to both cameras will leave the essential

[1] Note that, the second camera's center can be set on either X-, Y-, or Z-axis; the resultant parametrization using X- or Y-axis can be similarly derived. We opt for Z-axis for the convenience of closed-form bounding function evaluation (*cf.* Section 3.4.3).

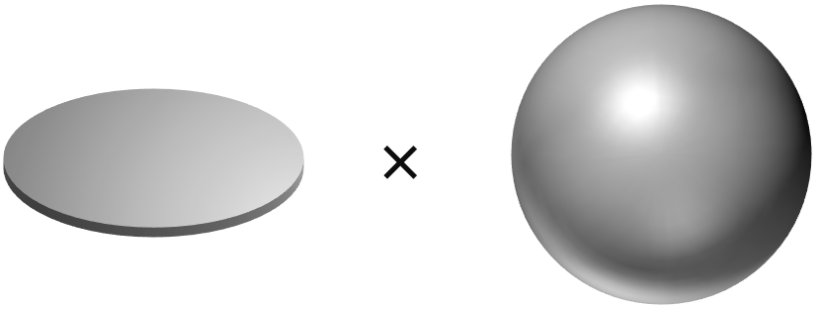


Figure 3.1: The essential manifold is parameterized as the product space of a solid 2D disk $\mathbb{D}^2_\pi$ and a solid 3D ball $\mathbb{B}^3_\pi$, corresponding to rotations of the first and second camera respectively. (Note that the disk is thickened to aid in visualization)

matrix invariant. In other words, they form an *equivalence class* which is a member of the 2D rotation group $\mathrm{SO}(2)$. In order to "factor out" these Z-axis rotations, we apply *group quotient operator* to one of the two $\mathrm{SO}(3)$ groups as $\mathrm{SO}(3)/\mathrm{SO}(2)$. In this way we can represent the essential space as $\mathrm{SO}(3) \times \big(\mathrm{SO}(3)/\mathrm{SO}(2)\big)$, i.e. the product space of $\mathrm{SO}(3)$ – rotation space for one camera, and $\mathrm{SO}(3)/\mathrm{SO}(2)$ for the other camera. Note that there are still equivalence classes remaining, and each of them corresponds to four relative pose configurations [Hartley and Zisserman, 2004b; Tron and Daniilidis, 2014]. It is necessary to leave these equivalence classes there, as only one (unknown) configuration out of the four depicts the true relative pose.

We adopt the angle-axis representation for 3D rotations, with which any rotation is representable as a point in a solid radius-π ball in 3-space, *i.e.* $\mathbb{B}^3_\pi$. Thus $\mathrm{SO}(3)$ can be parameterized as $\mathbb{B}^3_\pi$. The remaining problem is how to parameterize $\mathrm{SO}(3)/\mathrm{SO}(2)$. It is known in topology [Lee, 2010] that $\mathrm{SO}(3)/\mathrm{SO}(2)$ is homeomorphic to $\mathbb{S}^2$. Instead of this, we directly parameterize $\mathrm{SO}(3)/\mathrm{SO}(2)$ using angle-axis representation of camera rotation, as detailed in the following.

With angle-axis representation, it is easy to verify that in our setup, the X-Y plane of $\mathbb{B}^3_\pi$ effectively encodes all "Z-axis-free" rotations we need. This is because the X-Y plane of $\mathbb{B}^3_\pi$ contains all rotations whose Z-axis components are zero while X-axis and Z-axis components are arbitrary. Concretely, let $\mathbf{v}$ be the angle-axis vector of $\mathbf{R}$, *i.e.* $\mathbf{R} = \exp([\,\mathbf{v}\,]_\times)$, we avoid the freedom of Z-axis rotation by setting v^3, the 3rd element of vector $\mathbf{v}$, to be 0. Thus our search space for the first rotation $\mathbf{R} = \exp([v^1, v^2, 0]_\times)$ is reduced to the 2D disk $\mathbb{D}^2_\pi$ on the equator plane of the π-ball. Now, we have "squeezed" a 3D radius-π ball to a flat 2D radius-π disk in the X-Y plane.

Without loss of generality, we assume the first camera's rotation $\mathbf{R}$ is of 2-dof and "Z-axis free"; we denote this as $\mathbf{v} \in \mathbb{D}^2_\pi$. Let $\mathbf{R}' = \exp([\,\mathbf{v}'\,]_\times)$, then the essential manifold is parameterized by 5D vectors $(\mathbf{v}, \mathbf{v}') \in \mathbb{D}^2_\pi \times \mathbb{B}^3_\pi$. See Figure 3.1 for an illustration. To recover a 3×3 essential matrix $\mathbf{E}$ from $(\mathbf{v}, \mathbf{v}')$, one simply needs to recover rotations matrices $(\mathbf{R}, \mathbf{R}')$ from $(\mathbf{v}, \mathbf{v}')$, then compute $\mathbf{E}$ with (3.3).

Comparison to previous work. Some previous works such as [Helmke et al., 2007; Subbarao et al., 2008] base their parametrization on Singular Value Decomposition (SVD) of essential matrix. Although these representations also originate from $\mathrm{SO}(3) \times \mathrm{SO}(3)$, they do not provide the geometric interpretation of their parameters, and are not suitable for our BnB search. Recently, an concurrent independent work of Tron and Daniilidis [2014] chooses the same coordinate system as ours and uses the essential matrix formulation in (3.3). One difference between [Tron and Daniilidis, 2014] and our work is that, [Tron and Daniilidis, 2014] computes geodesic distance between two equivalence classes of two 6D $\mathrm{SO}(3) \times \mathrm{SO}(3)$ elements, while we propose an explicit parametrization of the 5D manifold $\mathrm{SO}(3) \times \big(\mathrm{SO}(3)/\mathrm{SO}(2)\big)$.

3.3 Optimization Criteria

With the parametrization described above, we are ready to formally define the optimality, and formulate the problem we will solve.

Let $(\mathbf{x}, \mathbf{x}')$ be a putative feature correspondence pair represented as unit 3D vectors, both corresponding to an unknown 3D scene point $\mathbf{X} \in \mathbb{R}^3$. Note $(\mathbf{x}, \mathbf{x}')$ may be subject to outliers and measurement noise. We represent the two cameras by their absolute orientations $\mathbf{R}$ and $\mathbf{R}'$, which jointly encode the essential matrix $\mathbf{E} = \mathbf{E}(\mathbf{R}, \mathbf{R}')$. The epipolar equation $\mathbf{x}'^{\mathrm{T}}\mathbf{E}\mathbf{x} = 0$ gives an algebraic error metric for measuring the optimality of an essential matrix. In this work, we will use the geometrically meaningful angular reprojection error, which is defined as

$$\begin{aligned} \measuredangle(\mathbf{R}^{\mathrm{T}}\mathbf{x}, \mathbf{R}'^{\mathrm{T}}\mathbf{x}') &\doteq \min_{\mathbf{X}} \max\big(\angle(\mathbf{R}^{\mathrm{T}}\mathbf{x}, \mathbf{X}), \angle(\mathbf{R}'^{\mathrm{T}}\mathbf{x}', \mathbf{X} - \mathbf{C}')\big) \\ &= \min_{\mathbf{X}} \max\big(\angle(\mathbf{x}, \mathbf{R}\mathbf{X}), \angle(\mathbf{x}', \mathbf{R}'(\mathbf{X} - \mathbf{C}'))\big) \end{aligned} \tag{3.4}$$

where $\angle(\cdot\,,\cdot)$ denotes the angle between two vectors, and $\mathbf{C}' \equiv [0, 0, 1]^{\mathrm{T}}$. We use the symbol $\measuredangle(\cdot\,,\cdot)$ to denote the angular reprojection error, which is the maximum of the two angular residuals.

With this angular error definition, there are two options to define the optimality of essential matrix $\mathbf{E}(\mathbf{R}, \mathbf{R}')$, corresponding to the following two problems.

Problem 3.1 (Inlier-set cardinality maximization)**.** *Given feature correspondences* $(\mathbf{x}_i, \mathbf{x}'_i)$ *and a prescribed angular error tolerance* ϵ*, the optimal essential matrix* $\mathbf{E}(\mathbf{R}, \mathbf{R}')$ *maximizes the cardinality of the inlier set (or consensus set) as*

$$\max_{\mathbf{R},\mathbf{R}'} |\mathcal{I}|\,, \;\; s.t. \;\; \forall i \in \mathcal{I},\; \measuredangle(\mathbf{R}^{\mathrm{T}}\mathbf{x}_i, \mathbf{R}'^{\mathrm{T}}\mathbf{x}'_i) \leq \epsilon \tag{3.5}$$

where $\mathcal{I}$ *denotes the inlier set and* $|\cdot|$ *represents cardinality. A pair of correspondences* $(\mathbf{x}_i, \mathbf{x}'_i)$ *is considered to be an inlier with respect to* ϵ *if* $\measuredangle(\mathbf{R}^{\mathrm{T}}\mathbf{x}_i, \mathbf{R}'^{\mathrm{T}}\mathbf{x}'_i) \leq \epsilon$*.*

Problem 3.2 (Angular reprojection error minimization)**.** *Given feature correspondences*

$(\mathbf{x}_i, \mathbf{x}'_i)$, *the optimal essential matrix* $\mathbf{E}(\mathbf{R}, \mathbf{R}')$ *is found by*

$$\min_{\mathbf{R},\mathbf{R}'} \|\mathbf{e}\|, \;\; s.t. \;\; e_i = \measuredangle(\mathbf{R}^{\mathrm{T}}\mathbf{x}_i, \mathbf{R}'^{\mathrm{T}}\mathbf{x}'_i) \tag{3.6}$$

where $\|\cdot\|$ *is a certain norm.*

Solving Problem 3.2 gives rise to an exact essential matrix minimizing angular error; however the result is sensitive to outliers. The goal of this work is to optimally solve Problem 3.1 with an exact inlier-set cardinality, thus it is intrinsically robust. Note that the solution to Problem 3.1 may not be unique. To solve essential matrix both robustly and exactly, one can solve Problem 3.2 with existing methods (*e.g.* [Hartley and Kahl, 2007]) after obtaining the true inliers with the proposed method.

Although global optimization for Problem 3.2 is studied in [Hartley and Kahl, 2007], solving the cardinality maximization problem globally optimally is still extremely difficult due to its obvious combinatorial and discrete nature. In the following, we approach the problem as a continuous optimization, and solve it by BnB search over the continuous parameter domain – the 5D product space $\mathbb{D}^2_\pi \times \mathbb{B}^3_\pi$.

3.4 Branch and Bound over $\mathbb{D}^2_\pi \times \mathbb{B}^3_\pi$

Recall that the goal is to globally maximize the inlier-cardinality as shown in (3.5). We treat this problem as continuous optimization and solve it via 5D space BnB. A high-level description of our method is given below. For the ease of manipulation, we use a 5D cube $\mathbb{C}^5_\pi$ with half side-length π to enclose the $\mathbb{D}^2_\pi \times \mathbb{B}^3_\pi$ space[2]. The initial cube $\mathbb{C}^5_\pi$ can be divided into smaller cubes. For each such cube, we compute the lower-bound (*LB*) as well as the upper-bound (*UB*) of the inlier-set cardinality for all rotations within it. *LB* and *UB* will be compared with the best value found so far, then this cube will be discarded or sub-divided. In the following we will denote a cube by $C_\sigma(\bar{\mathbf{R}}, \bar{\mathbf{R}}')$, where σ is its half side-length, and $\bar{\mathbf{R}}, \bar{\mathbf{R}}'$ are the center rotations of the corresponding 2D square and 3D cube respectively.

As is true for any BnB algorithm, the key to success is to find effective and efficient bounds. Below we will explain how we achieve this.

3.4.1 Lower-bound Computation

Finding a lower-bound for the cardinality maximization problem is relatively easy. It can be done simply by evaluating the cardinality function at a single point within the cubical domain. Obviously, the cardinality obtained in this way is necessarily a lower-bound, as it must not be greater than the true maximal cardinality with rotation in that cube.

The following procedure computes a lower-bound for a cube $C_\sigma(\bar{\mathbf{R}}, \bar{\mathbf{R}}')$ with respect to a prescribed angular error tolerance ϵ.

[2]The points outside $\mathbb{D}^2_\pi \times \mathbb{B}^3_\pi$ represent the same transformation at some point inside. This does not matter for our purpose and it brings no difficulty for the optimization. If one sub-cube in $\mathbb{C}^5_\pi$ falls entirely outside $\mathbb{D}^2_\pi \times \mathbb{B}^3_\pi$ during the BnB search, it can be ignored safely.

1. Check all candidate correspondences $(\mathbf{x}_i, \mathbf{x}'_i)$, with center rotations $\bar{\mathbf{R}}, \bar{\mathbf{R}}'$.
2. Count how many *feasibility inequalities* $\angle(\mathbf{R}^{\mathrm{T}}\mathbf{x}_i, \mathbf{X}_i) \leq \epsilon$ and $\angle(\mathbf{R}'^{\mathrm{T}}\mathbf{x}'_i, \mathbf{X}_i - \mathbf{C}') \leq \epsilon$ can be satisfied with some $\mathbf{X}_i$.
3. Report the above count as a lower-bound for this cube.

Step 2 of the procedure is done by solving a series of *feasibility test problems*. How to perform such tests will be explained in Section 3.4.3.

3.4.2 Upper-bound Computation via Relaxation

In solving maximization (as opposed to minimization) with BnB, it is in general more difficult to find a proper upper-bound (than to find a lower-bound).

The following procedure gives our solution to finding suitable upper-bound of the cardinality function for a given cube $C_\sigma(\bar{\mathbf{R}}, \bar{\mathbf{R}}')$ and tolerance ϵ.

1. Check all correspondences $(\mathbf{x}_i, \mathbf{x}'_i)$ with center rotations $\bar{\mathbf{R}}, \bar{\mathbf{R}}'$.
2. Count how many *relaxed feasibility inequalities* $\angle(\bar{\mathbf{R}}^{\mathrm{T}}\mathbf{x}_i, \mathbf{X}_i) \leq \epsilon + \sqrt{2}\sigma$ and $\angle(\bar{\mathbf{R}}'^{\mathrm{T}}\mathbf{x}'_i, \mathbf{X}_i - \mathbf{C}') \leq \epsilon + \sqrt{3}\sigma$ can be satisfied with some $\mathbf{X}_i$.
3. Report the above count as an upper-bound for this cube.

Note that, in Step 2 we solve a *relaxed* feasibility test problem, as the thresholds in the right side of the inequalities have been enlarged (relaxed), leading to more correspondences to be claimed as inliers, hence increasing the inlier cardinality.

To show that the upper-bound is valid (*i.e.* no solution in the cube yields larger inlier-set cardinality), a lemma and its proof are given below.

Lemma 3.1. *For a 5D cubic domain $C_\sigma(\bar{\mathbf{R}}, \bar{\mathbf{R}}')$, solving the above relaxed feasibility problem gives a valid upper-bound of the inlier-set cardinality.*

Proof. Our proof follows from two lemmas of [Hartley and Kahl, 2007], which show that, for any vector $\mathbf{x} \in \mathbb{R}^3$, given two arbitrary rotations $\mathbf{R}, \bar{\mathbf{R}}$ (with $\mathbf{v}$ and $\bar{\mathbf{v}}$ as their angle-axis representations), one must have $\angle(\mathbf{R}\mathbf{x}, \bar{\mathbf{R}}\mathbf{x}) \leq \angle(\mathbf{R}, \bar{\mathbf{R}}) \leq \|\mathbf{v} - \bar{\mathbf{v}}\|$.

Let's first fix $\mathbf{R}'$, and consider a 2D square domain of $\mathbf{R}$ centered at $\bar{\mathbf{R}}$ with half side-length σ. Suppose $\mathbf{R}^*$ is the optimal rotation, among all rotations within this domain, such that the corresponding inlier-set $\mathcal{I}$ is maximized. Therefore $\mathbf{R}^*$ must be feasible for inlier points, *i.e.* $\forall i \in \mathcal{I}$ one has $\measuredangle(\mathbf{R}^{*\mathrm{T}}\mathbf{x}_i, \mathbf{R}'^{\mathrm{T}}\mathbf{x}'_i) \leq \epsilon \Rightarrow \angle(\mathbf{R}^{*\mathrm{T}}\mathbf{x}_i, \mathbf{X}_i) \leq \epsilon$ with some $\mathbf{X}_i$. Then for the center rotation $\bar{\mathbf{R}}$ we have

$$\begin{aligned} \angle(\bar{\mathbf{R}}^{\mathrm{T}}\mathbf{x}_i, \mathbf{X}_i) &\leq \angle(\mathbf{R}^{*\mathrm{T}}\mathbf{x}_i, \mathbf{X}_i) + \angle(\bar{\mathbf{R}}^{\mathrm{T}}\mathbf{x}_i, \mathbf{R}^{*\mathrm{T}}\mathbf{x}_i) \\ &\leq \epsilon + \angle(\bar{\mathbf{R}}, \mathbf{R}^*) \\ &\leq \epsilon + \|\bar{\mathbf{v}} - \mathbf{v}^*\| \\ &\leq \epsilon + \sqrt{2}\sigma. \end{aligned} \tag{3.7}$$

This result implies that, if we relax the right side of the feasibility inequality from ϵ to $\epsilon + \sqrt{2}\sigma$ and evaluate inlier cardinality with respect to the center rotation, then the obtained cardinality will be no less than the optimal cardinality obtained within this cube, *i.e.* the one corresponding to $\mathbf{R}^*$.

For the other rotation $\mathbf{R}'$ (which is a 3-dof rotation) and vector $\mathbf{v}'$, a similar result can be obtained, except that in this case one has $\sqrt{3}\sigma$ for a 3D cubic domain instead of $\sqrt{2}\sigma$. Combining both rotations we have: for each point i in the optimal inlier-set with rotations in $C_\sigma(\bar{\mathbf{R}}, \bar{\mathbf{R}}')$, both $\angle(\bar{\mathbf{R}}^{\mathrm{T}}\mathbf{x}_i, \mathbf{X}_i) \leq \epsilon + \sqrt{2}\sigma$ and $\angle(\bar{\mathbf{R}}'^{\mathrm{T}}\mathbf{x}'_i, \mathbf{X}_i - \mathbf{C}') \leq \epsilon + \sqrt{3}\sigma$ must be satisfied with some $\mathbf{X}_i$. This completes the proof and the upper-bound is valid. □

3.4.3 Efficient Bounding with Closed-form Feasibility Test

Solving upper-bound and lower-bound necessitates the feasibility test task. This task is: given a pair of camera rotations $\mathbf{R}, \mathbf{R}'$ (along with $\mathbf{C} \equiv \mathbf{0}, \mathbf{C}' \equiv [0, 0, 1]^{\mathrm{T}}$), test whether or not a correspondences pair $(\mathbf{x}, \mathbf{x}')$ is an inlier with respect to the given angular reprojection error threshold ϵ. It can be formally formulated as

Problem 3.3 (Feasibility test for determining inliers)**.** *The inliers can be determined by the flowing feasibility test:*

Given	$\mathbf{x}, \mathbf{x}', \mathbf{R}, \mathbf{R}', \mathbf{C}, \mathbf{C}', \varepsilon, \varepsilon'$
does there exist	$\mathbf{X}$
such that	$\angle(\mathbf{R}^{\mathrm{T}}\mathbf{x}, \mathbf{X} - \mathbf{C}) \leq \varepsilon$
and	$\angle(\mathbf{R}'^{\mathrm{T}}\mathbf{x}', \mathbf{X} - \mathbf{C}') \leq \varepsilon'$

where $\varepsilon = \varepsilon' = \epsilon$ for the feasibility test in lower-bound computation, and $\varepsilon = \epsilon + \sqrt{2}\sigma$, $\varepsilon' = \epsilon + \sqrt{3}\sigma$ for the relaxed one in upper-bound computation.

One way to do such a test is by *two-view triangulation* [Hartley and Sturm, 1997]. It has been shown in [Ke and Kanade, 2007; Kahl and Hartley, 2008] that this problem can be solved by Second Order Cone Programming (SOCP). We have tested this method experimentally using a commercial SOCP solver (MOSEK). It worked successfully on very small numbers of feature points but with high computational demand, preventing us from doing larger experiments. We were therefore motivated to seek a faster solution.

Built upon previous work [Enqvist et al., 2011], our bounds are derived with efficient feasibility test in closed-form. The intuition is: to verify whether or not $(\mathbf{x}, \mathbf{x}')$ is compatible with a tentatively given essential matrix, one does not have to recover the corresponding 3D point $\mathbf{X}$. Instead, it is sufficient to check whether or not the epipolar relationship of the two points is satisfied. See Figure 3.2 for an illustration. A similar idea was proposed in [Hartley and Kahl, 2007], where a Linear Programming solver is used for feasibility tests.

Our method avoids using convex programming. It is a direct application of the following theorem which is a simple extension of that in [Enqvist et al., 2011]. Recall

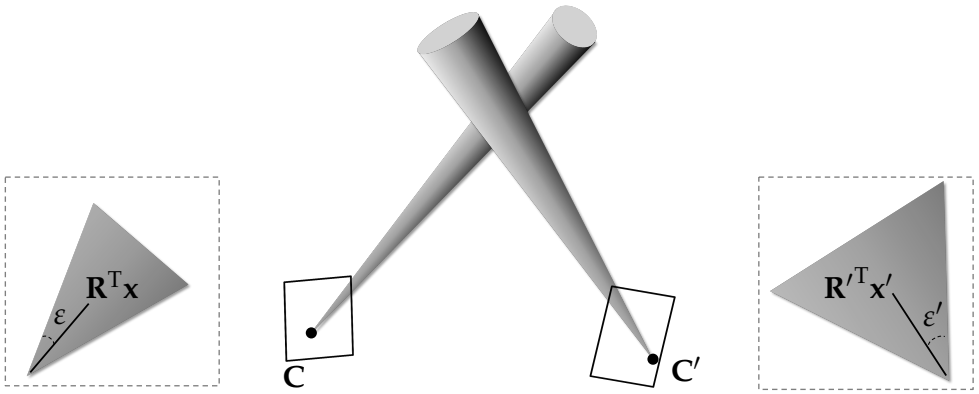


Figure 3.2: Illustration of the feasibility test. Given camera relative motion parameters, a correspondence pair is an inlier if and only if the two cones shown in the figure intersect. These two cones have the camera centers as their vertices, and angular error thresholds as their half apex angles.

that, the first camera is centered the origin and the second one is on Z-axis. If we represent the unit vectors $\mathbf{R}^{\mathrm{T}}\mathbf{x}$ and $\mathbf{R}'^{\mathrm{T}}\mathbf{x}'$ in spherical coordinates, they become

$$\mathbf{R}^{\mathrm{T}}\mathbf{x} = \begin{bmatrix} \sin\theta\cos\varphi \\ \sin\theta\sin\varphi \\ \cos\theta \end{bmatrix}, \quad \mathbf{R}'^{\mathrm{T}}\mathbf{x}' = \begin{bmatrix} \sin\theta'\cos\varphi' \\ \sin\theta'\sin\varphi' \\ \cos\theta' \end{bmatrix}. \tag{3.8}$$

Theorem 3.1. *Given a pair of correspondences* $\mathbf{x}, \mathbf{x}'$*, rotation matrices* $\mathbf{R}, \mathbf{R}'$ *and camera centers* $\mathbf{C} \equiv \mathbf{0}, \mathbf{C}' \equiv [0,0,1]^{\mathrm{T}}$*, representing* $\mathbf{R}^{\mathrm{T}}\mathbf{x}$ *and* $\mathbf{R}'^{\mathrm{T}}\mathbf{x}'$ *in spherical coordinates as* (θ, φ) *and* (θ', φ')*, we have: Problem 3.3 is feasible if and only if*

$$\begin{cases} \theta \leq \theta' + \varepsilon + \varepsilon' \\ |\varphi - \varphi'| \leq \omega \end{cases}, \tag{3.9}$$

where ω *is given below*

$$\omega = \begin{cases} \arcsin(\frac{\sin\varepsilon}{\sin\theta}) + \arcsin(\frac{\sin\varepsilon'}{\sin\theta'}), & if\ \theta < \theta' \\ \arccos(\frac{\cos(\varepsilon+\varepsilon') - \cos\theta\cos\theta'}{\sin\theta\sin\theta'}), & if\ \theta \in [\theta', \theta'+\varepsilon+\varepsilon'] \\ \pi, & \textit{if any of the above is undefined} \end{cases}. \tag{3.10}$$

Proof of this theorem can be found in [Enqvist et al., 2011]. The geometric intuition behind Theorem 3.1 is easy to discern. Consider the limit case when $\varepsilon \to 0$ and $\varepsilon' \to 0$ (thus $\omega \to 0$), then $|\varphi - \varphi'| \leq \omega \Rightarrow \varphi = \varphi'$ says that the two viewing rays of the two points lie in the same *half-plane* containing the baseline, and $\theta < \theta' + \varepsilon + \varepsilon' \Rightarrow \theta < \theta'$ entails that the two viewing rays intersect in this half-plane.

Based on this theorem, both lower-bound and upper-bound for a cube can be made in closed-form. The evaluation is efficient with elementary computation (and counting), using basic trigonometric functions.

Algorithm 3.1: BnB search in $\mathbb{D}_\pi^2 \times \mathbb{B}_\pi^3$ for optimal essential matrix maximizing the inlier set

Input: Images point pairs $(\mathbf{x}_i, \mathbf{x}_i'), i = 1, \ldots, M$; angular error threshold ϵ.
Output: Optimal essential matrix $\mathbf{E}^*$ and corresponding inlier set $\mathcal{I}^*$ of size N^*.

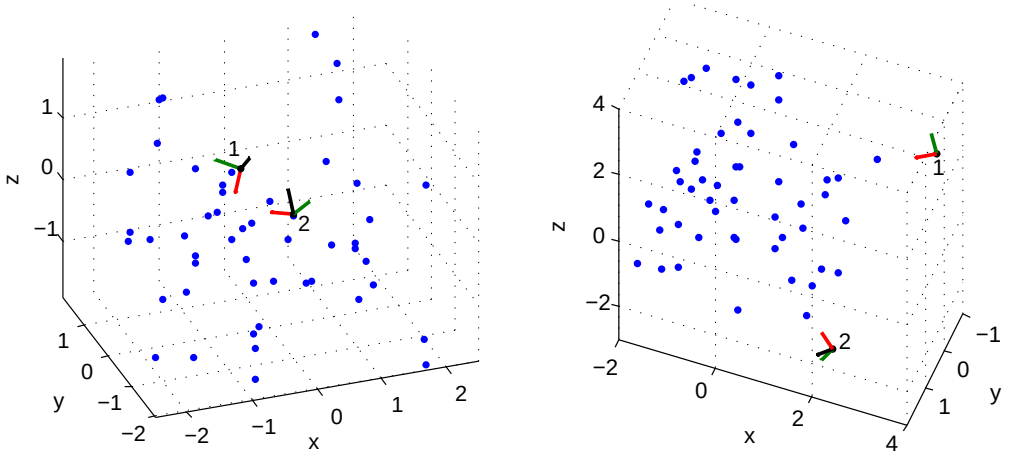
Divide $[-\pi, \pi]^5$ into small sub-cubes and push them into priority queue Q.
Set $N^* = 4$. *%we need to find at least $N^* = 5$ points*
loop
  Read out a cube with the highest upper-bound UB from Q.
  Quit the loop if $UB = N^*$.
  Divide it into $2^5 = 32$ sub-cubes with equal side length.
  foreach sub-cube $C_\sigma(\bar{\mathbf{R}}, \bar{\mathbf{R}}')$ **do**
    Set its lower-bound LB and upper-bound UB to be 0.
    foreach correspondence pair $(\mathbf{x}_i, \mathbf{x}_i')$ **do**
      LB++, if Problem 3.3 is feasible with $\bar{\mathbf{R}}, \bar{\mathbf{R}}', \epsilon, \epsilon$.
      UB++, if Problem 3.3 is feasible with $\bar{\mathbf{R}}, \bar{\mathbf{R}}', \epsilon+\sqrt{2}\sigma, \epsilon+\sqrt{3}\sigma$.
    end
    if $LB > N^*$ **then**
      Update $N^* = LB$, $\mathbf{E}^* = \mathbf{E}(\bar{\mathbf{R}}, \bar{\mathbf{R}}')$ and also $\mathcal{I}^*$.
    end
    Discard this cube if $UB \leqslant N^*$; otherwise put it into Q.
  end
end

Degeneracy. Note that when a feature point (θ, φ) either falls on Z-axis or is sufficiently close to it ($\theta < \epsilon$ or $\theta < \epsilon'$), the above functions for ω are not defined. In such cases, the feasibility test always returns true.

3.4.4 The Main Algorithm

Armed with the above developments of domain parametrization, lower and upper bounds, and closed-form feasibility test, we are now ready to present our main algorithm. Although it appears to be a bit technically heavy, the central idea and the implementation are rather simple: for each parameter domain, *i.e.* a 5D cube, count the number of feature correspondences that pass the feasibility test (or, relaxed feasibility test) as the lower-bound (or, upper-bound) of the cardinality, and try to update the solution and discard this cube accordingly. Algorithm 3.1 summarizes the algorithm in pseudo-code form.

Initial Cubes. Before the BnB loop we divide the initial cube $[-\pi, \pi]^5$ into smaller cubes as it is less likely that a large cube can be discarded. In our implementation we use $6^5 = 7776$ initial cubes with equal side length.



(a) Wide FOV (b) Narrow FOV

Figure 3.3: Typical configurations of the synthesized cameras and 3D points for the wide-FOV (*left*) and narrow-FOV (*right*) cases.

Search Strategy. The BnB algorithm uses the *best-first-search* strategy. Concretely, it maintains a priority queue of the active cubes, whose priorities are set to be their upper-bounds. In this way, the BnB algorithm always explores the most promising cube first.

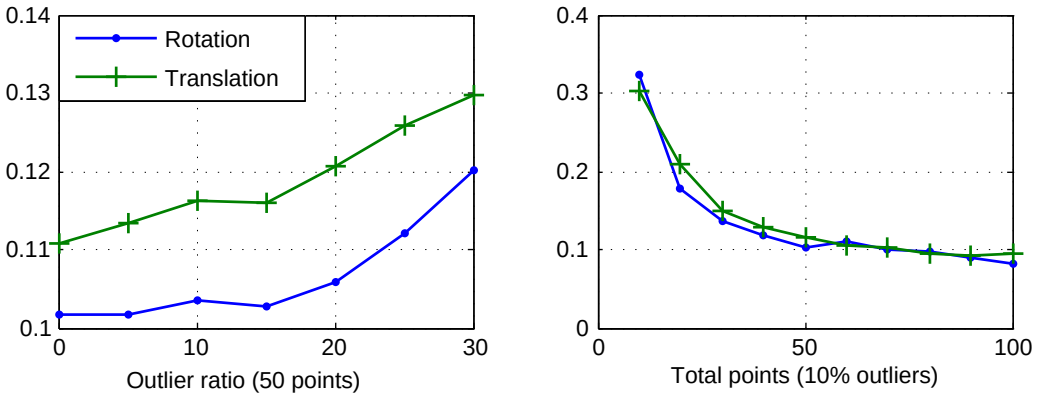
Proof of Convergence. The convergence of the algorithm is easy to see, as when the side-lengths of all cubes asymptotically diminish to zero, the gap between the upper-bound and lower-bound will be zero too.

3.5 Experiments

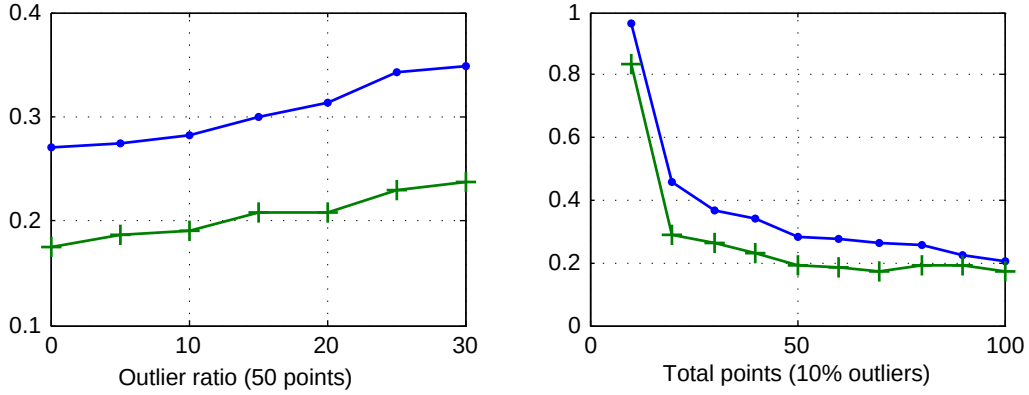
In this section, we report the experimental results on synthetic scenes and real imageries. Our method is implemented in C++, and tested on a standard PC with Intel i7 3.4GHz 4-core CPU and 8GB memory.

3.5.1 Synthetic Scene Test: Normal Cases

The main goal of experiments on synthetic data is to verify the correctness of the proposed method, including the essential manifold parameterization and the BnB algorithm. In these experiments, we set the angular error threshold to be 0.002 radians (about 0.115 degrees). Inlier number is the main index for essential matrix evaluation as our goal is to optimally maximize it. Nevertheless, we will also report the estimation error of essential matrix. For better comprehension, we use classic parametrization $\mathbf{E} = [\hat{\mathbf{t}}]_\times \hat{\mathbf{R}}$, and evaluate error of $\hat{\mathbf{R}}$ and $\hat{\mathbf{t}}$. Rotation error is the angle between $\hat{\mathbf{R}}$ and ground truth rotation. As $\hat{\mathbf{t}}$ is obtained up to a scale, we define translation error as the angle between $\hat{\mathbf{t}}$ and ground truth translation. Note that, as



(a) Wide FOV

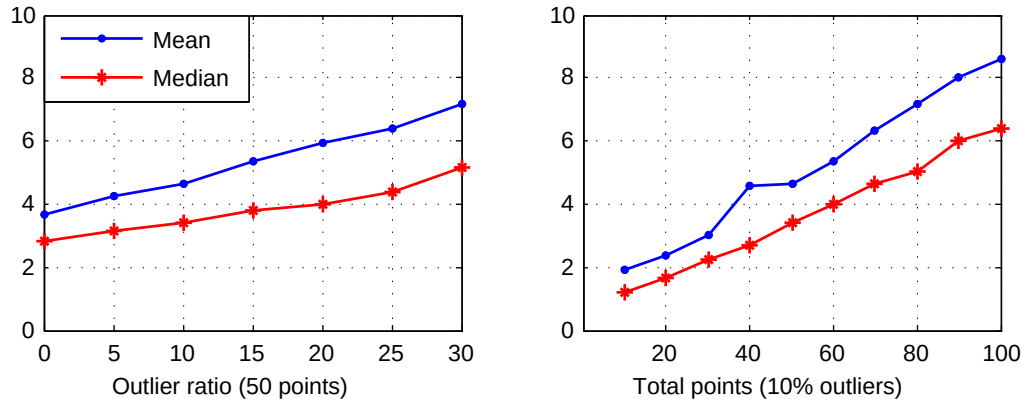


(b) Narrow FOV

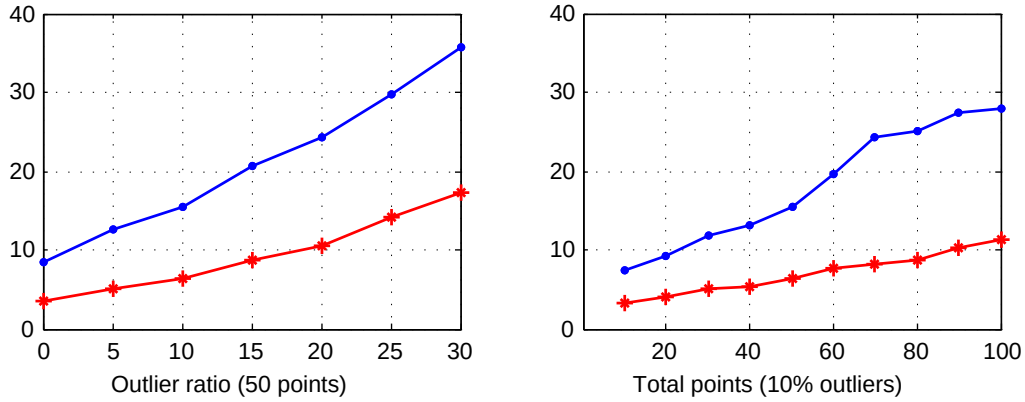
Figure 3.4: Average rotation and translation errors (both in degrees) for 50 runs of our method in synthetic wide-FOV (*top*) and narrow-FOV (*bottom*) tests with respect to different outlier ratios and total points.

discussed in Section 3.3, these results can be further improved by minimizing the reprojection error of obtained inliers (which is not used here).

Wide Field-Of-View (Omnidirectional Camera). In this test synthetic data with random points and two omnidirectional cameras which have 360° field of view were used. We synthesized 50 configurations of different points and camera poses. The points were generated in a cube centered at the origin with side length 4, and camera centers were generated from a Gaussian distribution centered at the origin with $\sigma = 0.5$. Gaussian noise with $\sigma = 0.001$ was added to all the projected image points. To generate outliers, we randomly perturbed the image points in the first camera by over 10 degrees. We tested our method first on different numbers of outliers with fixed total points (50), and then on different numbers of points with fixed outlier ratio (10%). As expected, *our method succeeded in all the tests in terms of finding out all the true inliers.* Average rotation and translation errors of the 50 configurations are



(a) Wide FOV

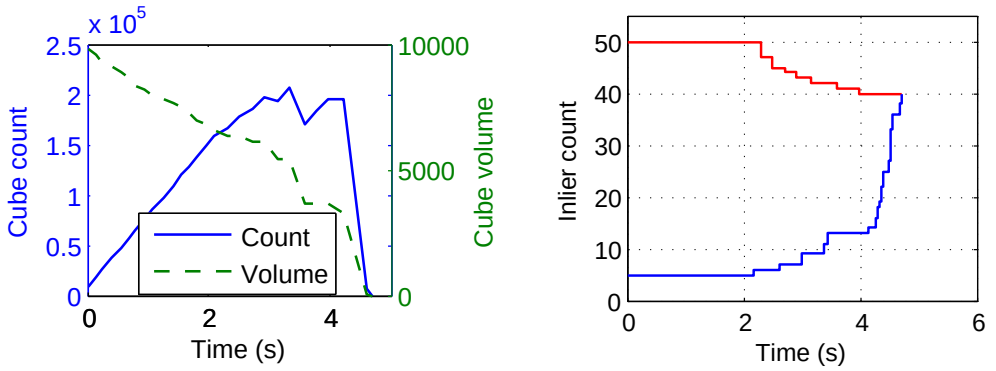


(b) Narrow FOV

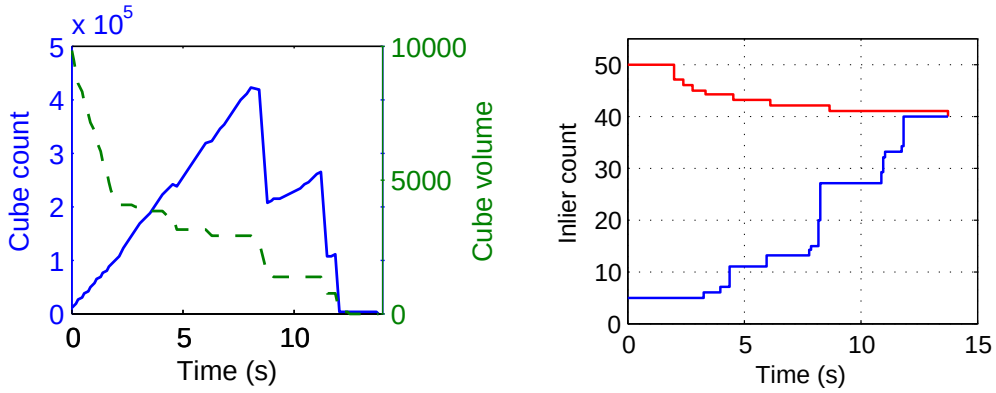
Figure 3.5: Average running time (in seconds) for 50 runs of our method in synthetic wide-FOV (*top*) and narrow-FOV (*bottom*) tests with respect to different outlier ratios and total points.

shown in Figure 3.4. Clearly, the error increases with outlier ratio and decreases with total point number. Average running time is shown in Figure 3.5. In general, it took the method longer time to converge when higher levels of outliers were present. To visualize the behavior of BnB, we present typical evolution curves of active cubes and global bounds as a function of time in Figure 3.6.

Narrow Field-Of-View. We then tested our method under narrow field of views. We synthesized the situation where the points are confined in approximately 60° FOV of two regular pinhole cameras. The points were generated in a cube centered at the origin with side length 4, and cameras were randomly placed at a distance of about 4 facing the origin. Other settings were the same with that in wide-FOV tests. Again, our method successfully found out all the true inliers. The estimation error, running time, and typical BnB evolution are also shown in Figure 3.4, Figure 3.5



(a) Wide FOV



(b) Narrow FOV

Figure 3.6: Typical cube and bound evolutions of BnB in synthetic wide-FOV (*top*) and narrow-FOV (*bottom*) tests using 50 points with 20% (*i.e.* 10) outliers.

and Figure 3.6 respectively. It is clear that solving the problem with narrow-FOV is generally more difficult than that with wide-FOV, as evidenced by the larger rotation and translation errors as well as the longer running time of our method.

3.5.2 Synthetic Scene Test: Special Cases

We tested some special cases on wide FOV configuration, aiming to test the performance of the proposed method under special or extreme situations.

Large Outlier Ratio. To test the performance under large outlier ratio, we generated 50 points with 25 (50%), 30 (60%), 35 (70%) outliers respectively in the wide FOV configuration. Our method successfully found the true inliers in 11s, 26s and 81s respectively.

Pure Translational Motion. In this experiment, two cameras with pure (and random) translation as the ground-truth transformation and 50 points were synthesized.

We ran our method on these points, and the angle between the two estimated rotations $\mathbf{R}$ and $\mathbf{R}'$ is about 0.11 degrees, which indicates that our method successfully identified the equal rotation case.

All Scene Points on a Plane. We synthesized a planar case where all 50 points lie on a plane. This is a well-known degenerate case for fundamental matrix estimation, however it should not affect essential matrix estimation, as explained in [Nistér, 2004]. Our experiment in this case obtained a positive result and we successfully recovered the correct essential matrices with and without outliers. The rotation and translation errors are all below 0.15 degrees.

3.5.3 Real Image Test

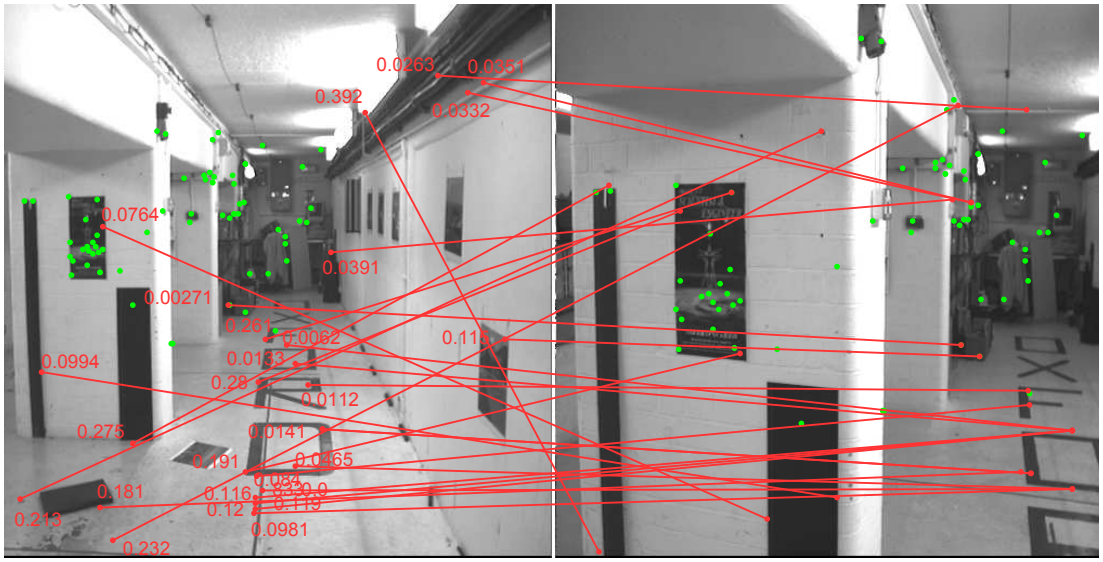
Images from both narrow-FOV and wide-FOV cameras were then used to evaluate the real-life performance of our method. We also tested RANSAC and LO-RANSAC [Chum et al., 2003] (with Option 4 of local optimization described in [Chum et al., 2003]) methods. In both RANSAC implementations, the 8-point method[3] was used and angular error threshold is adopted to distinguish outliers; the outlier ratio and probability parameter η were set to be 30% and 0.99 respectively. Note that, the goal of this work is not to replace the popular RANSAC and its variants in essential matrix estimation, but to provide a complementary (yet important) optimal method.

Narrow Field-Of-View. We tested our method on two image pairs from the Corridor and Valbonne data sets[4]. 94 and 106 SIFT matches were generated respectively for the two pairs as shown in Figure 3.7. The angular error threshold was set to 0.0015 radians. We parallelized the BnB search with 8 threads, and our method converged in 221s and 453s respectively. Apparently, it takes quite more time than on synthetic data of the same size. However, this is reasonable as will be analyzed as follows. On a 600×600 image from a 60°-FOV camera, a small pixel difference, say 3 pixels, yields about 0.3-degree angle difference. To tell outliers from inliers at this accuracy of both camera orientations, the 5D cube would have to be divided into $(\frac{180}{0.3/\sqrt{2}})^2\times(\frac{180}{0.3/\sqrt{3}})^3 \approx 8\times10^{14}$ blocks for a complete search method, and this is also a very difficult task for our BnB. The number of detected inliers is 66 for Corridor image and 89 for Church image, indicating 29.8% and 16% outlier ratios respectively. The detected inliers and outliers are shown in Figure 3.7. For some outliers, we show their angular errors (optimally solved via bi-section and SOCP [Kahl and Hartley, 2008]) with the obtained essential matrix.

We then repeated both RANSAC and LO-RANSAC 1,000 times with the same angular error threshold; the resulting inlier numbers are shown in the first two columns of Table 3.1. The heuristic and stochastic nature of random sampling scheme can be

[3]The 5-point method (with Hartley and Li [2012]'s solver) was also tested. It performed comparably with or slightly worse than the 8-point method; the latter one is thus presented.

[4]http://www.robots.ox.ac.uk/~vgg/data/data-mview.html



(a) Corridor

(b) Church

Figure 3.7: Results on narrow-FOV images. Green and red dots are respectively inlier and outlier correspondences found by our method. For outliers we labeled their angular reprojection errors in radius. (Best viewed on screen and with zoom-in)

clearly seen, as the mean performances of the 1,000 runs are not satisfactory. Moreover, both the two methods failed to detect the same inlier number as ours. This can be explained by the fact that algebraic solution of essential matrix is not consistent with the meaningful geometric error metric. In future, we plan to compare our

Table 3.1: Inlier-set maximization performance of different methods. The first column lists the images and correspondence numbers. The second and third columns show the maximal and mean inlier number detected by RANSAC and LO-RANSAC in their 1,000 runs. The last column shows the inlier number from our method and the running time (with 8 threads).

Images (#points)	RANSAC max/mean #inliers	LO-RANSAC max/mean #inliers	Our method #inliers (time)
Corridor (94)	63 / 32.5	65 / 50.0	66 (221s)
Church (106)	82 / 32.8	87 / 35.5	89 (453s)
Building (202)	160 / 146.9	161 / 151.7	163 (52s)
Office (151)	124 / 91.4	126 / 104.2	126 (43s)

method with RANSAC methods in high-noise situations where algebraic solutions can be severely biased.

Wide Field-Of-View (Fisheye Camera). In order to test our method in real-life wide-FOV case, a camera with a fisheye lens was used to capture images of the scene with up to 190° FOV. The camera was calibrated with the method of Scaramuzza et al. [2006].

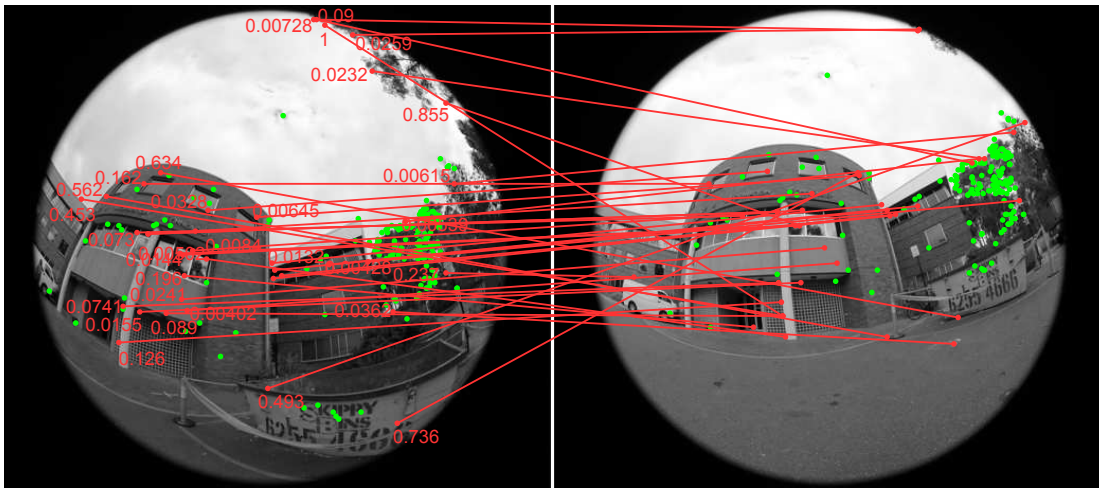
Figure 3.8 shows two typical pairs referred as Building and Office. The angular threshold was set to be 0.003 radians for these images. Our method converged in 52s and 43s for the two image pairs respectively as shown in Table 3.1, and the results indicate 19.3% and 16.6% outlier ratios. In general, the angular errors of outliers are larger than that in the narrow-FOV case (see Figure 3.8), and our method ran faster on wide-FOV images. This result is consistent with our synthetic experiments and similar discoveries reported in previous works [Daniilidisl and Spetsakisz, 1997; Hartley and Kahl, 2007; Enqvist and Kahl, 2009; Heller et al., 2012].

3.6 Conclusion

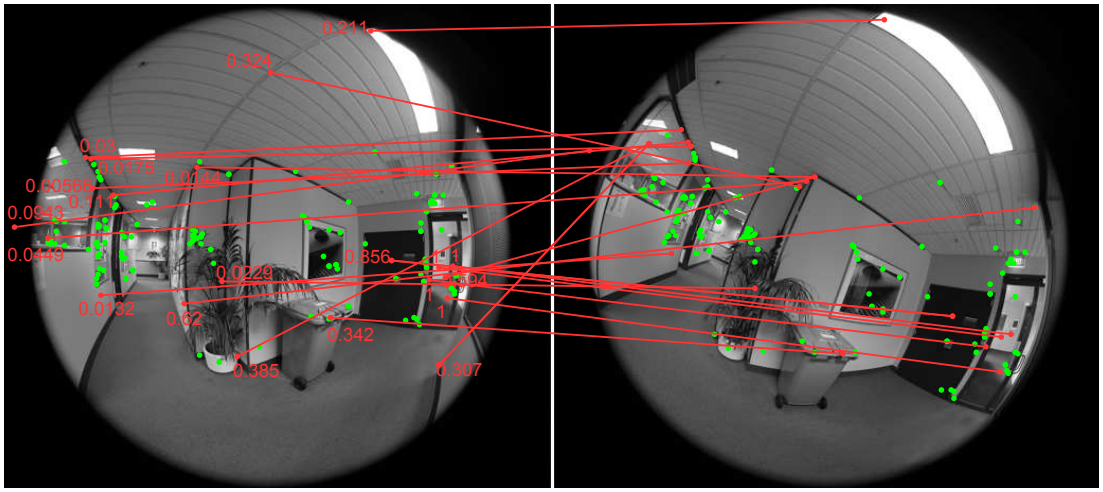
A branch-and-bound global optimization method is proposed for essential matrix estimation via inlier-set cardinality maximization under geometric (angular) error. An explicit and geometrically meaningful parametrization of the 5D essential manifold, *i.e.* $\mathbb{D}^2_\pi \times \mathbb{B}^3_\pi$, is used to perform the BnB search. Based on previous works [Hartley and Kahl, 2007] and [Enqvist et al., 2011], closed-form bounding functions of inlier-set cardinality are derived, leading to efficient bound evaluation in the 5D space BnB.

Currently, the proposed method is slow especially for cameras with a small field of view. Nevertheless, due to its optimality, the method can be used as a benchmark for method evaluation, or be applied in situations where robustness or accuracy is highly desired while speed is not crucial.

To make the method faster and more practical, there are some strategies we would like to investigate in future. For example, a possible one is to get an initial essential



(a) Building



(b) Office

Figure 3.8: Results on wide-FOV images taken with a fisheye camera. Green and red dots are respectively inlier and outlier correspondences found by our method. For outliers we labeled their angular reprojection errors in radius. (Best viewed on screen and with zoom-in)

matrix estimate using RANSAC, then search the parameter space with the proposed BnB in a small region around this estimate. Taking advantage of prior knowledge on motion to confine the parameter space is a metric of continues optimization in contrast to discrete combinatorial optimization. Since our BnB algorithm can be easily parallelized, another idea would be porting it onto modern GPU where a significant speedup can be expected.

Chapter 4

2D Camera and 3D Camera Relative Pose Estimation from Scene Constraints

The popularization of consumer depth cameras has greatly boosted the development of 3D-based entertainment, augmented/mixed reality, 3D reconstruction *etc*. Depth cameras can provide the three-dimensional perception of the scene, while conventual 2D color cameras can provide the color of the visual world. 3D geometry and color are complementary information, and can be fused for advanced perception. As the depth camera and color camera are at different locations in the 3D space, the depth and color images they captured cannot be fused directly. To achieve color and depth data fusion, the relative pose between the two cameras is required to register the two images. Some consumer depth cameras come with a rigidly attached color camera, with their relative pose being fixed as a manufacturer setting (e.g., a Microsoft Kinect device). Despite this, in practice one may need a different color camera (e.g., a high-definition CCD camera as shown in Figure 4.6), or different relative poses between depth and color cameras for different scenarios (e.g., for hand-held cameras). Therefore, relative pose estimation is an important task.

Relative pose estimation of a color camera and a depth camera is not an easy task. It can not be achieved by conventional relative pose estimation techniques for color cameras described in Chapter 3, i.e., computing the motion using feature correspondences. This is because a color image and a depth image bear different types of information of the scene, and no suitable cross-modality feature extraction and matching technique exist at present. In fact, feature matching is a difficult task even by manual feature point selection: a salient image point on one image may not be salient enough for manual selection on the other image.

Up until now, color and depth camera relative pose estimation has been mostly achieved as a camera extrinsic calibration task, in a way that is very similar to the conventional procedure of calibrating a regular color camera. Typically, this involves the user waving a plate with a checkerboard pattern in front of the camera(s). However, these methods require multiple color and depth image pairs of the checkerboard plane for off-line calibration, after which the relative pose should remain fixed when

in actual use, as otherwise the calibration should be again. Moreover, the calibration procedure is cumbersome which necessitates a lot of human intervention to label the feature correspondences (on color images) or plane regions (on depth images) across multiple images.

In this chapter, we propose a new method to estimate the color and depth camera relative pose with *single shot* (i.e., with one pair of color and depth images). The color camera and depth camera can be placed at different locations according to the actual needs, as long as they share a common field of view. The estimation can be on-site, or even can be applied for post-processing. We make the assumption that each camera has already been intrinsically calibrated, thus we deal with extrinsic calibration only. This assumption is not too restrictive as the intrinsic parameters can be readily obtained using a separate intrinsic calibration procedure or from manufacturers' specifications. Additionally, in many practical applications, the intrinsic parameters of the cameras are often fixed while the extrinsic parameters can be subject to change.

Another feature of the proposed method is that it works in a correspondence-free style, and does not need a special calibration pattern (e.g., a checkerboard plane). Instead, it makes use of a small set of known scene constraints, such as known distances/angles and distances/angle equivalences. Based on these geometric constraints from the scene, we formulate the relative pose estimation problem as a 2D-3D image registration problem. We leverage the metric information of the scene to minimize the geometric error from the registration results. In this way, we not only free ourselves from building image correspondences, but also directly optimize the image registration quality. Additionally, we propose a singe-view 3D reconstruction algorithm using these geometric constraints from the scene. The algorithm is applied to obtain an initial solution to the 2D-3D registration problem.

4.1 Related Work

Relative pose estimation of a generically configured color and depth camera pair has attracted considerable attention from computer vision [Zhang and Zhang, 2011], mixed and augmented reality [Pilet et al., 2006]Gomez et al. [2005] and robotics communities [Zhang and Pless, 2004]. In this section we briefly review the most relevant prior work to our method.

As previously mentioned, color and depth camera relative pose estimation has been mostly done as an office camera extrinsic calibration process. For example, the camera calibration works by Herrera C et al. [2012] and Zhang and Zhang [2011] are closely related to ours. Herrera C et al. [2012] presented a method to calibrate the intrinsic and extrinsic parameters of two color cameras and a depth camera by using a planar pattern surface. The calibration procedure is similar to the conventional plane-based color camera calibration [Zhang, 2000], i.e., a checkerboard is waved before the cameras and imaged from various poses. The user needs to give the correspondences across the color images and mark the plane region on the depth images. The calibration method of Zhang and Zhang [2011] is similar, although

they additionally make use of the correspondences between the color image and the depth image to improve accuracy. Smisek et al. [2011] calibrated Kinect cameras using correspondences between the RGB image and the infrared image.

Among other related works, Zhang and Pless [2004] proposed a practical procedure to extrinsically calibrate an RGB camera with a 2D Laser-Rangefinder (LRF), where a checkerboard pattern was moved freely in front of both sensors. Extrinsic calibration was achieved by solving a set of linear constraints which were subsequently refined by iterative minimization of the reprojection error. Likewise, Vasconcelos et al. [2012] also studied the calibration of a color camera with a 2D LRF and they showed that a set of three pairs of planes and lines provides a minimal configuration to solve the calibration problem linearly. Scaramuzza et al. [2007] proposed a method to estimate the relative pose between a color camera and a 3D LRF. However, this method requires manually selecting correspondences between the color image and the depth image. To this end, they convert a range image to a so-called bearing angle image on which natural features of a scene are highlighted to facilitate manual feature selection. Alismail et al. [2012] used a calibration target consisting of a single circle to estimate the extrinsic parameters of a camera-Lidar system. The method detects the circles (projected as ellipses) and their physical centers on multiple color images, reconstruct them to 3D, and register them onto the point clouds from Lidar to obtain the relative pose.

The relative pose estimation problem we consider in this chapter also has a close relationship with hand-eye calibration, which has been intensively studied in computer vision and robotics [Tsai and Lenz, 1989; Horaud and Dornaika, 1995; Dai et al., 2009]. These methods works by solving the well-known conjugate equation $\mathbf{AX} = \mathbf{XB}$ to estimate the relative pose (i.e., $\mathbf{X}$), which necessitates moving the camera system and estimating the ego motions of the cameras (i.e., $\mathbf{A}$ and $\mathbf{B}$). In contrast, no ego-motion estimation is required by our method. In addition, these methods have commonly used an algebraic error to solve the problem, while we minimize geometrically-meaningful errors.

Our method works in a single-shot fashion, i.e., it only requires one color image and one depth image to estimation the relative pose. To our knowledge, little work that uses a single shot has been published, except for the work by Geiger et al. [2012b]. However, although Geiger et al. [2012b] used one image pair to calibrate the cameras, they actually set up multiple checkerboard patterns in a large scene. This is essentially similar to a multi-shot configuration. Their calibration process further involved an explicit segmentation of the planar regions corresponding to the checkerboard. In contrast, our single-shot method does not use checkerboard patterns, nor does it require any plane detection and segmentation process. Instead, it exploits the geometric constraints in the scene to solve the problem. Our method is more flexible in that it can estimate the relative pose in a relatively small scene. Furthermore, we minimize the geometric error of scene constraints given by image registration, thus directly optimizing the image registration quality.

4.2 Color and Depth Camera Relative Pose Estimation from Scene Constraints

4.2.1 Problem Statement

The ultimate goal of color and depth camera relative pose estimation is to bring the obtained 2D color image and 3D depth image into perfect geometric alignment (registration). This process can be intuitively understood as either,

- to color each pixel in the depth image with the correct color, or, conversely,
- to assign each pixel in the color image a correct depth value.

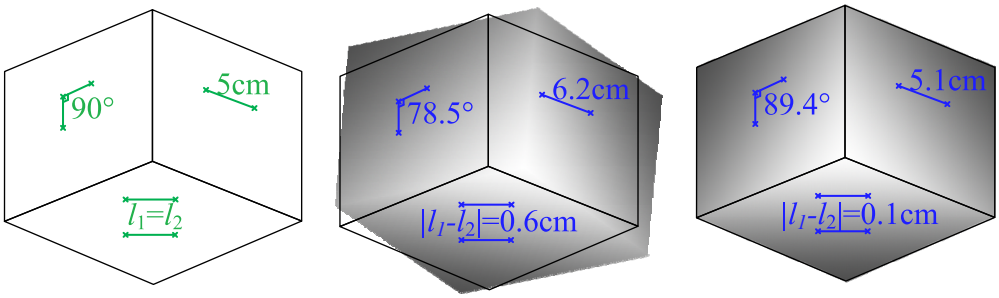
Given a perfect registration, these two statements are equivalent. Mathematically, the underlying task estimates the relative geometric transformation between the two cameras. The transformation involves a rotation matrix, $\mathbf{R}$, and a translation vector $\mathbf{t}$, forming a 6-dof (degrees-of-freedom) rigid transformation $\Theta = \{\mathbf{R}, \mathbf{t}\}$.

When the two cameras are observing the same scene, for each scene point we can simply obtain its 3D coordinates from the depth image. If we are able to identify the corresponding pairs of 2D image points and depth points, then we can obtain the mapping relationship between the 3D scene and its 2D image coordinates, which is defined in terms of the (unknown) relative pose parameters Θ. But this approach, while conceptually simple and straightforward, is not an easy task in practice. The main difficulty comes from the necessary requirement of knowing cross-modality feature correspondences between the RGB image and the depth image. Moreover, most existing methods require multiple shots of a calibration pattern to obtain the relative pose.

4.2.2 The Proposed Approach

In this work, we solve the color and depth camera relative pose estimation problem in a "single shot" and "correspondence free" style. Our method aims to optimize the final "warping" quality directly by minimizing the geometric registration error between a color camera and a depth camera. We evaluate the warping quality from scene knowledge. Under perfect relative pose estimates, scene knowledge observed from the color image should have identical measurements in the corresponding 3D point clouds from the depth image. Thus we can potentially achieve single-shot estimation by minimizing the discrepancy between this prior scene knowledge and its estimation. In addition, we can build a 3D estimation for the RGB image under assumptions about the smoothness and continuity of the scene. We call this process "*inverse projection*", which estimates 3D positions from 2D image coordinates.

Our method works by capturing a single shot of a scene, provided that certain constraints about the scene are easy to access. The relative pose estimation task is achieved by solving a 2D-3D registration problem. Of course, in the absence of scene constraints, doing such a 2D-3D registration is generally impossible due to the information loss in the projection from the 3D to 2D. Nevertheless, our knowledge



(a) (b) (c)

Figure 4.1: Illustration of the evaluation of scene knowledge. (a) An image of a scene containing three planes. Partial knowledge (ground truth) of the scene is labeled, which includes a known distance, distance equivalency and a known angle. Initial alignment of the color and depth images demonstrates a large discrepancy in evaluating the scene knowledge, as shown in (b). The goal of our method is to find the optimal rigid transformation Θ^* between the color and depth camera with which the alignment yields minimal errors, as shown in (c).

(including qualitative assessments) of the scene can help to provide feedback on the quality of the registration. For example, we invite the reader to look at the schematic example in Figure 4.1(b) where it is not difficult to suspect (or to guess) that this situation is *very likely* to be not registered well while Figure 4.1(c) gives a much better registration. In this example, scene constraints including known distances, distance equivalency, and known angles are involved (as discussed in the next subsection).

In the following sub-sections, we first introduce our inverse projection method to estimate a 3D position for a 2D image point. We then illustrate how to incorporate different scene constraints in evaluating the registration performance. Finally, we present our geometric-registration-error-minimization based method that directly optimizes the warping quality between the two images.

4.2.3 Inverse Projection Estimation

Given a rigid body transformation between the RGB and depth cameras, $\Theta = (\mathbf{R}, \mathbf{t})$, we can transform the point clouds $\mathcal{X}_D = \{(x_i^d, y_i^d, z_i^d)\}$ from the depth camera to the coordinates of the color camera, and project it onto the image plane using the intrinsic matrix $\mathbf{K}^c$. This procedure can be expressed as

$$\lambda_i^{cd}[u_i^{cd}, v_i^{cd}, 1]^T = \mathbf{K}^c[\mathbf{R}, \mathbf{t}][x_i^d, y_i^d, z_i^d, 1]^T, \tag{4.1}$$

where $[u_i^{cd}, v_i^{cd}]$ gives the color image point corresponding to the 3D point $[x_i^d, y_i^d, z_i^d]$ and λ_i^{cd} is the unknown projective depth. This mapping relationship can be compactly expressed as:

$$(u_i^{cd}, v_i^{cd}) = g(\Theta) \circ (x_i^d, y_i^d, z_i^d), \tag{4.2}$$

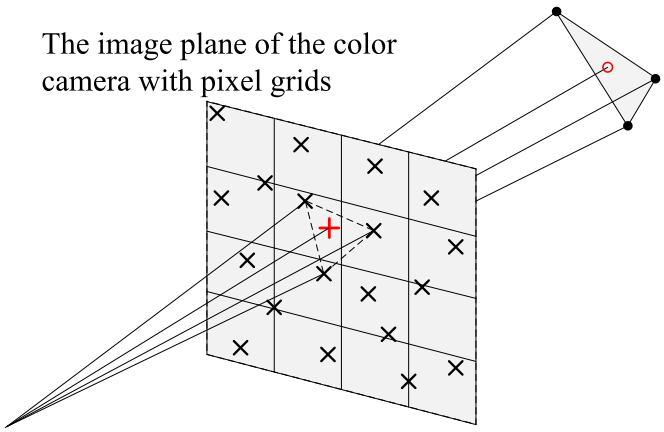


Figure 4.2: Inverse projection estimation: estimating 3D position for an image point on the color image with available 3D point clouds from depth camera. Black crosses and dots: the projected points from the depth image and their corresponding 3D points. Red plus and circle: an image point and its (estimated) corresponding 3D point.

where g denotes the transformation from a 3D point in the depth camera to the color image coordinate.

Now that we have obtained 3D positions for some image pixels on the color image, the question is: *can we obtain 3D positions for all the pixels in the color image?* This is generally impossible as the projection from 3D continuous world to 2D discrete grid is an information-loss procedure. Nonetheless, we can make a local (piecewise) smoothness assumption, under which the inverse projection $g^{-1}(\Theta)$ may be well defined, and we can recover (x_j^d, y_j^d, z_j^d) through $g^{-1}(\Theta) \circ (u_j^c, v_j^c)$ based on the local structure around particular (u_j^c, v_j^c).

We use triangulation to estimate this inverse projection $g^{-1}(\Theta)$, assuming a locally smooth surface. Specifically, we obtain surface triangles for the dense 3D point clouds $\{(x_i^d, y_i^d, z_i^d)\}$ from the depth camera. Given a current estimation of Θ, the triangles Δ_k are first rigidly transformed and then projected onto the image plane of the RGB camera. Note that the projected triangles do not necessarily correspond to the 2D Delaunay triangulation of the projected 3D points because, due to self-occlusion of the scene, there are possibly triangles overlaid on top of each other. For an image point (u_i^c, v_i^c), we first find the projected triangles containing this point, then back-project the image point onto the 3D space. We then obtain the estimated 3D positions from triangles containing the image point. If there are multiple 3D points corresponding to the image point, we choose the one with the smallest depth, thus effectively handling the occlusion. The procedure of inverse projection is illustrated in Figure 4.2.

Finally, the inverse projection can be written as

$$(x_i^d, y_i^d, z_i^d) = g^{-1}(\Theta) \circ (u_i^c, v_i^c), \tag{4.3}$$

where (u_i^c, v_i^c) is any point on the image plane of the RGB camera.

4.2.4 Scene Constraints

Once we have the inverse projection $g^{-1}(\Theta)$, we can evaluate metric information, and compare it with prior knowledge about the scene. This prior knowledge can include (but is not limited to) three broad types of constraints which we discuss here.

Known-distance Constraints: Suppose we have two feature points from the color image denoted as (u_i^c, v_i^c) and (u_j^c, v_j^c), and we know their Euclidean distance, l_{ij}. By applying the inverse projection, we obtain their would-be distance, which is $\|g^{-1}(\Theta) \circ (u_i^c, v_i^c) - g^{-1}(\Theta) \circ (u_j^c, v_j^c)\|$. Then, the discrepancy from the known distance, given by

$$e_k(\Theta) = |\|g^{-1}(\Theta) \circ (u_i^c, v_i^c) - g^{-1}(\Theta) \circ (u_j^c, v_j^c)\| - l_{ij}|, \tag{4.4}$$

measures how good the tentative registration is. This known-distance constraint fixes the distance between points on two lines. Given enough known distance constraints, we are able to recover the 3D coordinates of the points. In general there needs to be at least one known distance constraint to fix the global scale of the 2D-3D registration.

Distance-equivalency Constraints: If, for instance, we know that the distance between one pair of feature points, (u_i^c, v_i^c) and (u_j^c, v_j^c) should be the same as the distance between another pair of points, (u_k^c, v_k^c) and (u_l^c, v_l^c), then the observed discrepancy is expressed as $e_d(\Theta) = |\|g^{-1}(\Theta) \circ (u_i^c, v_i^c) - g^{-1}(\Theta) \circ (u_j^c, v_j^c)\| - \|g^{-1}(\Theta) \circ (u_k^c, v_k^c) - g^{-1}(\Theta) \circ (u_l^c, v_l^c)\||$. This is another useful constraint but it cannot be applied alone as solely using distance-equivalency constraints could result in the trivial solution of all distances being zero.

Angular Constraints: Besides the scene constraints from distance measurements, we can also evaluate angular constraints about the scene such as the preservation of orthogonal and parallel lines in the images. Discrepancy from an orthogonal constraint can be expressed as $e_o(\Theta) = (g^{-1}(\Theta) \circ (u_i^c, v_i^c) - g^{-1}(\Theta) \circ (u_j^c, v_j^c))^T (g^{-1}(\Theta) \circ (u_k^c, v_k^c) - g^{-1}(\Theta) \circ (u_l^c, v_l^c))$ while discrepancy from a parallel constraint can be expressed as $e_p(\Theta) = [g^{-1}(\Theta) \circ (u_i^c, v_i^c) - g^{-1}(\Theta) \circ (u_j^c, v_j^c)]_\times (g^{-1}(\Theta) \circ (u_k^c, v_k^c) - g^{-1}(\Theta) \circ (u_l^c, v_l^c))$, where for $\mathbf{a} = [a_1\ a_2\ a_3]^T$, $[\mathbf{a}]_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$. Other angle-based constraints such as the preservation of known angles, and of pairs of angles being the same (angle equivalency) can also be evaluated in a similar way. Note that angular constraints include a degenerate result, i.e., all the 3D points are at the camera center, which results in a trivial solution in 2D-3D registration.

As discussed in the next subsection, minimizing the total error with respect to the rigid body transformation, Θ, for all known image constraints enables us to solve the problem with a single shot of an RGB and depth system.

4.2.5 Geometric Error Minimization

Given an RGB color image, assume that certain metric information about the scene is available (e.g., inter-point distances between some pairs of image features, parallel or orthogonal constraints between lines, distance equivalency and so on – a set of rather mild and general conditions on the images), then for any tentative 2D-3D registration (parameterized by Θ), we can always quantitatively measure registration quality using the discrepancy between the estimation and the *a priori* knowledge. Minimizing this discrepancy directly leads to the optimal relative pose, as well as a direct optimization of the warping quality. This is the main idea behind our proposed method.

Mathematically, our method formulates the color and depth camera relative pose estimation problem of as searching for an optimal rigid transformation $\Theta^* = \{\mathbf{R}^*, \mathbf{t}^*\}$ that minimizes geometric error:

$$\Theta^* = \underset{\Theta \in \mathrm{SE}(3)}{\operatorname{argmin}} \sum_i e_i(\Theta)^2, \tag{4.5}$$

where $e_i(\Theta)$ is the discrepancy between a measurement and its corresponding prior knowledge under the transformation Θ.

Due to the fact that there is no explicit form of the inverse-projection function, we are not able to employ analytic gradient-based methods for solving the minimization problem. Instead, the implicit inverse-projection function means that evaluating the scene knowledge with respect to the rigid transformation results in a complex, nonlinear, optimization problem and numerical gradient-based methods such as the Levenberg-Marquardt algorithm [Moré, 1978] can be used. The desired rigid transformations reside on the Riemannian manifold $\mathrm{SE}(3)$, which is homeomorphic to $\mathrm{SO}(3) \times \mathbb{R}^3$ [Tron et al., 2008]. The constraints on $\mathrm{SO}(3)$ can be involved in parameterizing the rotation, and we use the angle-axis representation in our algorithm implementation.

Alternatively, other gradient-free algorithms such as the Nelder-Mead simplex downhill algorithm [Nelder and Mead, 1965] can be used. The Nelder-Mead algorithm can also be adapted with a "simplex downhill on manifold" optimization scheme from [Dreisigmeyer, 2006] to better exploit the geometry of the $\mathrm{SE}(3)$ manifold.

To solve the non-linear minimization problem in evaluating scene constraints, we need a good initial guess such as the one described in the next section.

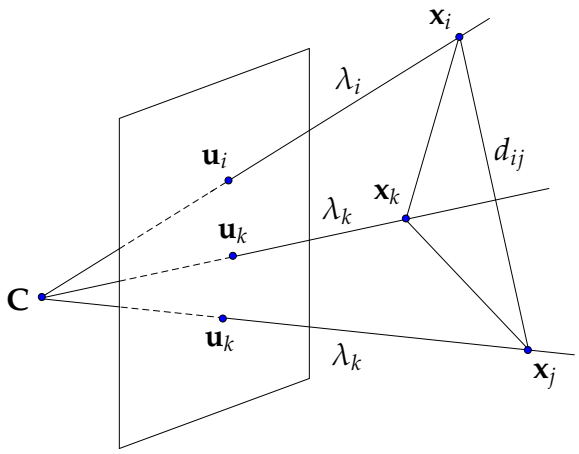


Figure 4.3: Single view reconstruction with scene constraints.

4.3 Initial Relative Pose Estimation

In this section, we present a simple solution to estimate the relative pose, which can be used to initialize the nonlinear optimization. Note that, although simple, this method is a self-contained solution and is interesting in its own right. The method takes advantage of the scene constraints to reconstruct the extracted color image points and then to register the reconstructed 3D points with the dense 3D point clouds from the depth camera to obtain a solution for the rigid body transformation. Note that the objective function for 3D registration is different from directly evaluating scene constraints as in (4.5).

4.3.1 Single View 3D Reconstruction

Generally, a single view 3D reconstruction is impossible without any scene information. But, with partial scene constraints such as known distances, distance equivalency, and known angles, we are able to reconstruct the 3D scene from measurements on a single-view image.

Under the color camera coordinate, the perspective imaging process is expressed as $\lambda_i[u_i^c \; v_i^c \; 1]^T = \mathbf{K}\,[\mathbf{I}\;\mathbf{0}]\,[X_i^c \; Y_i^c \; Z_i^c \; 1]^T$, where $[u_i^c \; v_i^c]^T$ is the image measurement, $[X_i^c \; Y_i^c \; Z_i^c]^T$ is the corresponding 3D position and λ_i is the unknown projective depth. This equation actually gives a direction constraint on the 3D position, say, $[X_i^c \; Y_i^c \; Z_i^c]^T = \lambda_i \mathbf{K}^{-1}[u_i^c \; v_i^c \; 1]^T$, i.e., the 3D point lies on the ray with direction $\mathbf{K}^{-1}[u_i^c \; v_i^c \; 1]^T$ with an unknown projective depth λ_i to be determined.

With scene constraints such as known distances, distance equivalency and known angles, we have further constraints on the projective depths. Thus it is possible to recover the scene structure. Take the known distance constraint as an example (Figure 4.3), the distance between two 3D points is measured as:

$$d_{ij} = \|\lambda_i \mathbf{K}^{-1}[u_i^c \; v_i^c \; 1]^T - \lambda_j \mathbf{K}^{-1}[u_j^c \; v_j^c \; 1]^T\|_2, \tag{4.6}$$

which gives constraint on the projective depths λ_i and λ_j. By defining

$$a_{ij} = (\mathbf{K}^{-1}[u_i^c \; v_i^c \; 1]^T)^T + (\mathbf{K}^{-1}[u_j^c \; v_j^c \; 1]^T), \tag{4.7}$$

(4.6) gives the following bilinear equation on λ_i and λ_j,

$$d_{ij}^2 = \lambda_i^2 a_{ii} + \lambda_j^2 a_{jj} - 2\lambda_i \lambda_j a_{ij}, \tag{4.8}$$

which can be equivalently expressed as:

$$[\lambda_i \; \lambda_j] \begin{bmatrix} a_{ii} & -a_{ij} \\ -a_{ij} & a_{jj} \end{bmatrix} \begin{bmatrix} \lambda_i \\ \lambda_j \end{bmatrix} = d_{ij}^2, \forall (i,j) \in \mathcal{N}, \tag{4.9}$$

where $\mathcal{N}$ defines the set of all measured point pairs.

We define a vector $\mathbf{\Lambda} = [\lambda_1, \lambda_2, \cdots, \lambda_n]^T$, which contains all the projective depths to determine. $\mathbf{Y} = \mathbf{\Lambda}\mathbf{\Lambda}^T$ is defined as the Gram matrix and $\text{rank}(\mathbf{Y}) = 1$. Define $\mathbf{A}_{ij} \in \mathbb{R}^{n \times n}$ as being element-wise zero except for $\mathbf{A}_{ij}^{ii} = a_{ii}$, $\mathbf{A}_{ij}^{ij} = \mathbf{A}_{ij}^{ji} = -a_{ij}$, $\mathbf{A}_{ij}^{jj} = a_{jj}$. Then, the bilinear constraint on the projective depth can be expressed as:

$$\text{tr}(\mathbf{A}_{ij}\mathbf{Y}) = d_{ij}^2, \; \forall (i,j) \in \mathcal{N}. \tag{4.10}$$

Finally the problem of single view 3D reconstruction from known distance constraints is formulated as:

$$\begin{aligned} &\text{Find } \mathbf{\Lambda}, \\ &\text{such that } \; \text{tr}(\mathbf{A}_{ij}\mathbf{Y}) = d_{ij}^2, \forall (i,j) \in \mathcal{N}, \\ &\mathbf{Y} = \mathbf{\Lambda}\mathbf{\Lambda}^T, \\ &\text{rank}(\mathbf{Y}) = 1. \end{aligned} \tag{4.11}$$

The quadratic constraint and the rank constraint are both non-convex, thus the entire optimization problem is non-convex. We take a similar strategy to [Li, 2010] to "convexify" the constraints, proposing to minimize the trace norm of $\mathbf{Y}$ rather than enforcing the rank-constraint implicitly. Finally we reach a trace norm minimization problem as:

$$\begin{aligned} &\min \text{trace}(\mathbf{Y}) \\ &\text{such that } \; \text{tr}(\mathbf{A}_{ij}\mathbf{Y}) = d_{ij}^2, \; \forall (i,j) \in \mathcal{N}. \end{aligned} \tag{4.12}$$

This is a standard Semi-Definite Programming (SDP) problem and we can use off-the-shelf solvers such as SDPT3 [Toh et al., 1999] to solve it efficiently. Once we obtain $\mathbf{Y}$, we can solve $\mathbf{\Lambda}$ by the singular value decomposition (SVD). Finally the 3D structure is recovered as $\mathbf{X}_i^c = \lambda_i \mathbf{K}^{-1}[u_i^c \; v_i^c \; 1]^T$. As an example, a single view 3D reconstruction for Figure 4.7(a) is illustrated in Figure 4.8(a).

Minimal Configuration: For each ray, there is an unknown projective depth λ_i. For n points in a complete connected graph with known distances, we have $n(n-1)/2$ constraints, therefore when $n(n-1)/2 \geq n$, we will have enough constraints to solve

λ_i. Thus $n = 3$ gives the minimal configuration. However, under this configuration, multiple solutions exist. To retrieve a unique solution, at least 4 points with known distances should be involved.

Other Scene Constraints: In the previous paragraph, we have taken the known distance constraint as an example to demonstrate how to recover 3D points from single-view 2D image measurements. In principle, other constraints can also be incorporated into the same framework as follows:

- The distance equivalency constraint is $d_{ij} = d_{kl}$, which gives a linear equation of $\mathbf{Y}$ as $\text{tr}(\mathbf{A}_{ij}\mathbf{Y}) = \text{tr}(\mathbf{A}_{kl}\mathbf{Y})$;
- The orthogonal constraint of lines L_{ij} and L_{kl} is expressed as $(\lambda_i\mathbf{K}^{-1}[u_i^c\ v_i^c\ 1]^T - \lambda_j\mathbf{K}^{-1}[u_j^c\ v_j^c\ 1]^T)^T(\lambda_k\mathbf{K}^{-1}[u_k^c\ v_k^c\ 1]^T - \lambda_l\mathbf{K}^{-1}[u_l^c\ v_l^c\ 1]^T) = 0$, which gives a linear equation of $\mathbf{Y}$ as $a_{ik}\mathbf{Y}_{ik} + a_{jl}\mathbf{Y}_{jl} - a_{il}\mathbf{Y}_{il} - a_{jk}\mathbf{Y}_{jk} = 0$;
- The parallel constraint of lines L_{ij} and L_{kl} is expressed as $\left[\lambda_k K^{-1}[u_k^c\ v_k^c\ 1]^T - \lambda_l\mathbf{K}^{-1}[u_l^c\ v_l^c\ 1]^T\right]_\times \left[\lambda_i K^{-1}[u_i^c\ v_i^c\ 1]^T - \lambda_j\mathbf{K}^{-1}[u_j^c\ v_j^c\ 1]^T\right] = 0$, which gives three linear equations of $\mathbf{Y}$.

All these constraints can be incorporated into the above trace norm minimization formulation naturally. Different types of scene knowledge constrain the 3D reconstruction to different extents. For example, using only the distance equivalency constraint, we can only achieve reconstruction up to a global scale, where a trivial solution as all depths being zero is included. Angle-based constraints, such as orthogonal or parallel constraints, result in 3D reconstruction up to a global scale and rotation. For single-view reconstruction from scene constraints, at least one known distance constraint is required to obtain a global scale.

Related Work: Note that our single view 3D reconstruction has connections with the Perspective-n-Point (PnP) problem [Lepetit et al., 2009], where the camera motion is the main focus to solve. Meanwhile, Zhang et al. [1998] used domain knowledge such as distances and angles to upgrade the affine structure into a Euclidean space by minimizing the sum of Mahalanobis distances, which is solved as a general nonlinear least-squares problem. Wilczkowiak et al. [2005] exploited geometric constraints through parallelepipeds for calibration and 3D modeling. Nevertheless, our method is based on recent progress in compressive sensing theory [Recht et al., 2010] and provides a more efficient implementation.

4.3.2 Point Cloud Registration

Now that we have sparse point clouds $\{\mathbf{X}_i^c\}$ from a single view 3D reconstruction, the initial transformation Θ_0 can be obtained by registering $\{\mathbf{X}_i^c\}$ to the dense point clouds $\{\mathbf{X}_j^d\}$ from the depth camera. The well-known Iterative Closest Point (ICP)

algorithm [Besl and McKay, 1992] can be used to get the solution as[1]

$$\Theta_0 = \underset{\Theta=(\mathbf{R},\mathbf{t})\in \mathrm{SE}(3)}{\operatorname{argmin}} \sum_i \min_j \|(\mathbf{R}^{-1}\mathbf{X}_i^c - \mathbf{R}^{-1}\mathbf{t}) - \mathbf{X}_j^d\|^2. \tag{4.13}$$

The ICP algorithm is simple and efficient. However, it can be easily stuck into local minima when the displacement of the two camera is reasonably large. Furthermore, when the scene contains similar local structures, there may be many local minima with low registration errors. To tackle these issues, we can use the globally optimal point cloud registration method (Go-ICP) proposed in Chapter 2. Go-ICP uses the same cost function as in ICP, and leverages a branch-and-bound scheme to guarantee the optimality (see Chapter 2 for more details). In our experiments, we found it to be very suitable for the sparse-to-dense 3D point cloud registration task in this chapter.

At this point, we have described the scene-constraint based color and depth camera relative pose estimation approach, as well as an initialization method which is also based on scene constraints. We will experimentally evaluate the proposed approach in the next section.

4.4 Experiments

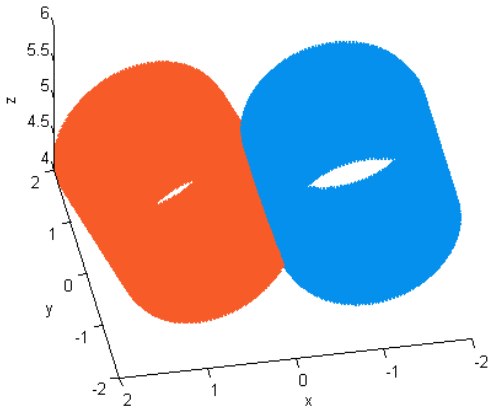
In this section, we present experimental results on generically-configured RGB-D camera rigs. We first give a synthetic scenario with two cylinders to illustrate the generality and performance of our method on minimizing geometric error. Then three sheets of A4 paper in the real world are used to extrinsically calibrate a generically configured RGB-D camera rig.

Performances of relative pose estimation and alignment were evaluated both qualitatively (by warping the depth image onto the color image) and quantitatively (by comparing with ground truth or measuring the geometric error from scene constraints). All the experiments were run on a computer with 2.4GHz Intel Core i5 CPU.

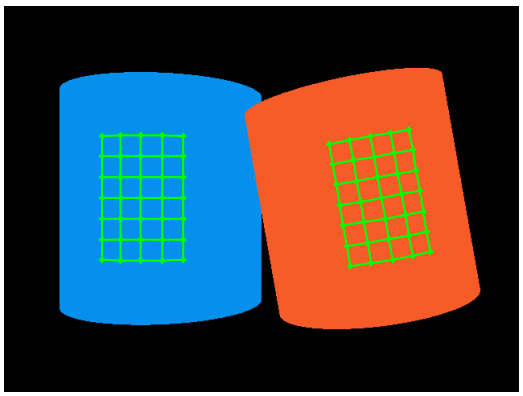
4.4.1 Tests on Synthetic Data

In the first experiment, a scene containing two non-parallel cylinders was synthesized as shown in Figure 4.4(a). We then synthesized a single-shot of an RGB-D camera. The color image was computed via a simple pinhole model, and the depth map was generated by the Z-buffer technique. The synthesized RGB-D image pair is shown in Figure 4.4(b) (with square grids overlaid) and Figure 4.4(c). We took the true side-length of the grid as constraints about the scene. In this experiment, we directly use

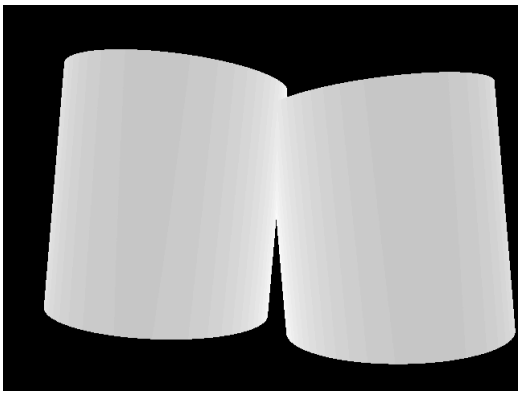
[1]Note that, in the problem of (4.5), transformation Θ is used to register the depth image onto the color image; in contrast, the sparse point cloud from the color camera is registered onto the dense point cloud from the depth camera in (4.13). To unify the parameter symbol, Θ is also used in (4.13), however its inverse transformation $(\mathbf{R}^{-1}, -\mathbf{R}^{-1}\mathbf{t}) \doteq (\bar{\mathbf{R}}, \bar{\mathbf{t}})$ is applied on the color point cloud. In practice, one can directly estimate $(\bar{\mathbf{R}}, \bar{\mathbf{t}})$ and then compute Θ trivially.



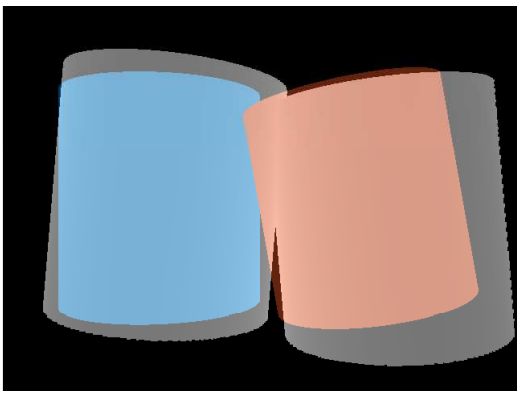
(a) A synthesized scene



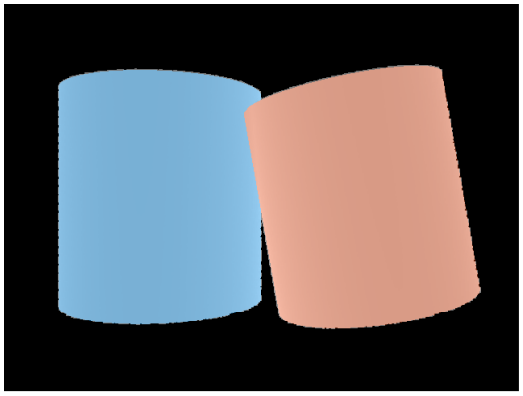
(b) Synthesized color image



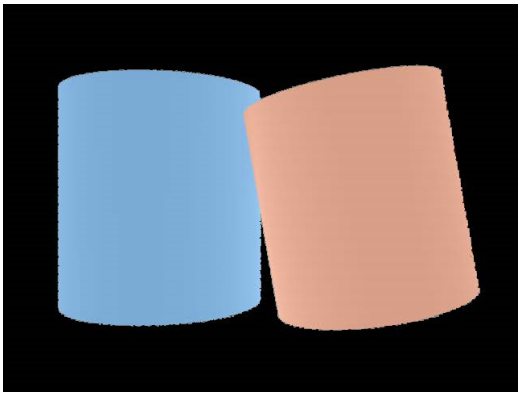
(c) Synthesized depth image



(d) Initial registration

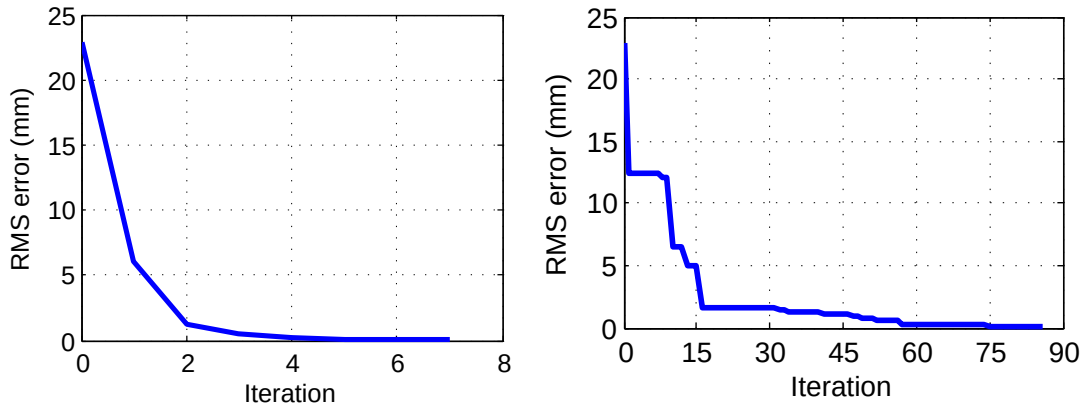


(e) Final registration



(f) GT registration

Figure 4.4: Experiments on a synthetic scene. A scene containing two cylinders shown in (a) was synthesized. The color image shown in (b) was created by projecting the points onto the image plane using a pinhole model. The side-length of the labeled grids are known and used in our method. The depth image shown in (c) was computed with the Z-buffer technique. Initial alignment of the color and depth images are shown in (d). The alignment result with our method and the ground truth are shown in (e) and (f) respectively. (**Best viewed on screen**)



(a) Optimization using LM (b) Optimization using NM

Figure 4.5: Convergence curve for the synthetic cylinder scene (RMS error *w.r.t.* iteration).

$\Theta_0 = (\mathbf{I}, \mathbf{0})$ as the initial parameter; both the Levenberg-Marquardt (LM) algorithm and the Nelder-Mead (NM) algorithm were applied to minimize the geometric error.

Quantitative Evaluation. The objective function minimizing the sum of squared errors was used for optimization, while the root mean square (RMS) errors were recorded during each iteration as the performance measure for better comprehension. Convergence curves of the proposed method using the LM and NM algorithm are shown in Figure 4.5. By our optimization, the RMS error is reduced from the initial 23mm to about 0.06mm. The optimization converged in 7 iterations with about 25 seconds using the LM algorithm, and 84 iterations with about 110 seconds using the NM algorithm. Table 4.1 presents the estimated relative pose by our method compared against the ground-truth relative pose. It can be seen that the results are high consistent with the ground truth. Table 4.2 further shows the relative pose error. The rotation error is evaluate as $\arccos\big((\text{trace}(\tilde{\mathbf{R}}^{\mathrm{T}}\mathbf{R}_{gt}) - 1)/2\big)$, where $\tilde{\mathbf{R}}$ and $\mathbf{R}_{gt}$ are the estimated and ground-truth rotation matrix respectively. Our method recovers the relative pose accurately, with rotation error below 0.3 degrees and translation error below 0.01m.

Table 4.1: Estimation results of our method compared with the ground truth on the synthetic cylinder scene. The rotation is expressed with angle-axis representation.

	Angle (°)	Axis	Translation (m)
Ground truth	5.067	-0.100 -0.128 -0.987	-0.113 -0.086 0.500
Our result (LM)	5.106	-0.096 -0.128 -0.986	-0.112 -0.078 0.503
Our result (NM)	5.100	-0.105 -0.127 -0.985	-0.114 -0.093 0.501

Table 4.2: Estimation error of our method on the synthetic cylinder scene.

	Rotation error (°)	Translation error (m)
Our result (LM)	0.021	0.008
Our result (NM)	0.027	0.007

Qualitative Evaluation. The color and depth image registration results are shown in Figure 4.4(d) and Figure 4.4(f), which are the registration results before, and after our optimization, respectively. The results from the LM and NM algorithms are almost visually indistinguishable. Visually inspected, our method yields satisfactory registration (e.g., the edges are well aligned).

4.4.2 Tests on a Real-world Scene

In the real-world task, we used the depth sensor on a Kinect device as our depth camera, and attached it to a high-resolution color camera; see Figure 4.6. Note that, our method can be adapted to other types of depth imaging sensors (e.g., a 3D LIDAR or a ToF camera).

We set up a scene containing three sheets of A4 paper with different orientations, as shown in Figure 4.7(a). These sheets of paper could just as well have been objects from an indoor scene such as a laptop screen, a book, a table or similar rigid objects with well-defined vertices. We then extracted the four corners of each sheet of A4 paper and the metric scene constraint was available as an international standard: the

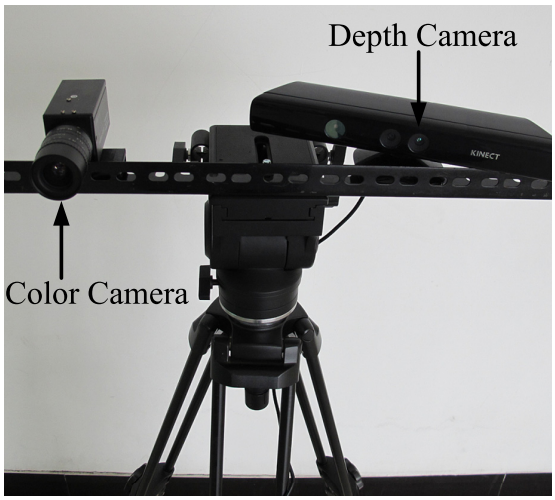
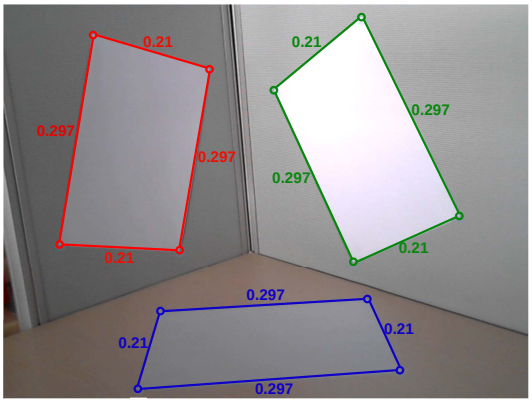
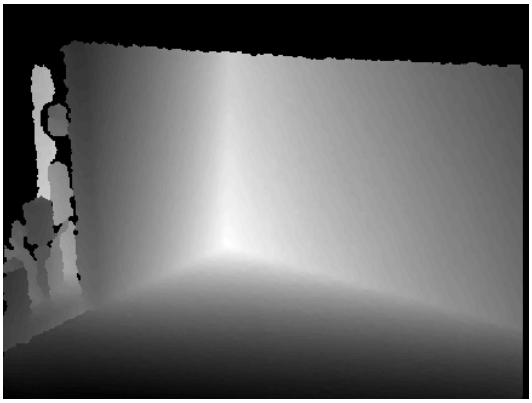


Figure 4.6: A customized RGB-D camera rig consisting of a high-resolution color camera, and a Kinect-for-Windows depth sensor. This rig was used in the experimental work described in this chapter.

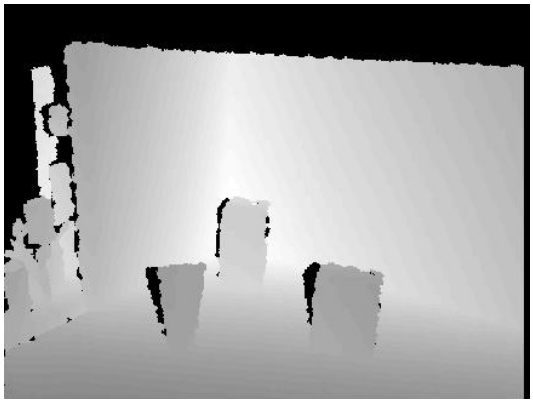


(a) Color image 1

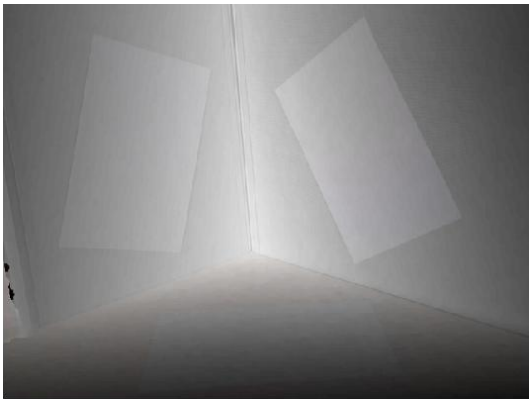
(b) Color image 2



(c) Depth image 1



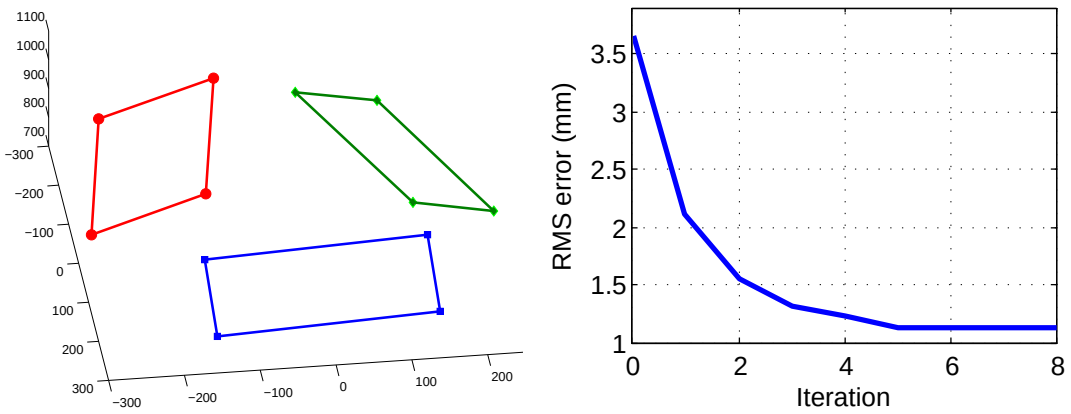
(d) Depth image 2



(e) Warping result 1

(f) Warping result 2

Figure 4.7: A real-world scene and its corresponding 3D reconstruction. (a) Three sheets of A4 paper are used to provide scene constraint. (b) Single view 3D reconstruction of the extracted points.



(a) 3D reconstruction

(b) RMS error *w.r.t.* iteration

Figure 4.8: Single-view 3D reconstruction result and convergence curve (optimized with LM) for the real-world scene of three sheets of A4 paper.

height and width of an A4 paper are 0.297m and 0.21m respectively.

To obtain a good initialization, we first apply the initial relative pose estimation approach described in Section 4.3. The single view 3D reconstruction result of the corner points is illustrated in Figure 4.8(a). We then register the reconstructed points with the dense point clouds from the depth camera to obtain an initial guess. Taking the quantization noise in the depth measurements obtained from the Kinect depth sensor into consideration, we utilized further constraints of the scene that the points were on planes to accurately estimate the depths for extracted points in the geometric error minimization procedure. Specifically, for each point on the color image, a local plane was fitted with some nearest neighbors of vertexes of its corresponding 3D triangle. The whole estimation procedure including the corner points extraction and running of the proposed method finished in minutes.

For comparison, the method of [Herrera C et al., 2012] was also applied to calibrate the same RGB-D camera rig. We used 40 color and depth image pairs of a planar calibration pattern with 10×8 checkerboard grids. The corner points on the color images and plane regions on the depth images were manually selected to calibrate the RGB-D camera rig. To avoid any bias, the intrinsic parameters from [Herrera C et al., 2012] were used in our method.

Quantitative Evaluation. In this experiment, the optimization using LM algorithm converged in 8 iterations, taking about 20 seconds. The convergence curve is shown in Figure 4.8(b). The RMS error was 3.7mm with the initial relative pose, and was reduced to 1.2mm after our optimization.

The estimated relative pose parameters from our method as well as that from [Herrera C et al., 2012] are presented in Table 4.3. As can be seen, the estimated parameters from the two methods are very similar. However, our method achieves a final RMS geometric error 1.2mm which is less than half that of the method in

Table 4.3: Results of our method and [Herrera C et al., 2012] in the real scene. The rotation is expressed with angle-axis representation.

	Angle (°)	Axis	Translation (m)	RMS error (m)
[Herrera C et al., 2012]	17.225	0.102 -0.986 0.131	0.280 0.046 0.083	0.0027
Our method	17.619	0.104 -0.983 0.153	0.273 0.043 0.091	0.0012

[Herrera C et al., 2012] (2.7mm).

Qualitative Evaluation. For a qualitative visual evaluation of warping, we register the depth image onto the color image with the obtained transformation parameters. Figure 4.7(e) shows the registration result of our method. To evaluate the registration quality more clearly, we added serval objects into the scene and obtained the color and depth images shown in Figure 4.7(b) and Figure 4.7(d). The registration result is shown in Figure 4.7(f), and it can be seen that the two images are well registered, with edges and discontinuities well aligned.

To further evaluate the method and compare with [Herrera C et al., 2012], we captured several color and depth image pairs in different scenes and used the estimated relative pose parameters to register the images. Registration results of the proposed method and [Herrera C et al., 2012] are compared in Figure 4.9, and our method achieves comparable or superior performance.

Application in Augmented Reality. Finally, we provide a demonstration of an augmented reality (AR) application for a generically configured RGB-D camera rig, where a virtual teapot is added into a complex scene. Figure 4.10 shows several frames of the AR video Using the estimated relative pose, the 3D information from the depth camera and color information from the RGB camera can be fused accurately and occlusions in the augmented reality image can be effectively handled. The registration result of the scene images is shown Figure 4.9 (top right).

4.5 Conclusion

We have presented a novel method to achieve relative pose estimation of a 2D color camera and a depth camera, with partially-known metric information of an observed scene and in a single-shot fashion. Overall, the required human intervention is minimal and not restrictive as users only need to manually mark some points for the method to automatically obtain the relative pose. The whole procedure can be done efficiently, and the input can be as simple as three sheets of A4 paper or by other user provided scene information. Our approach can greatly facilitate mixed and augmented reality applications, which, for example, might require the use of a specialized RGB camera in addition to a commodity depth sensor or where it is desired

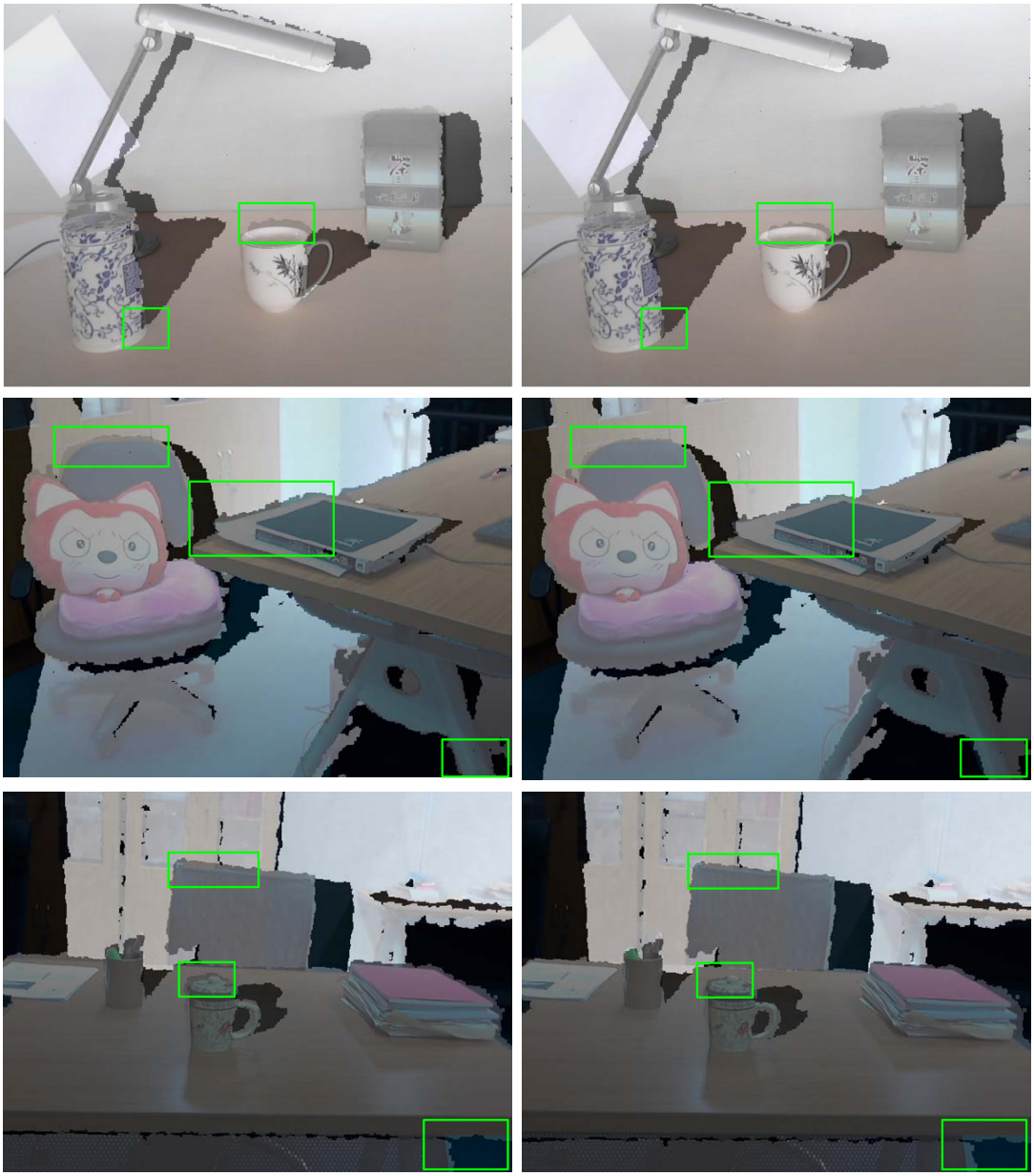


Figure 4.9: Registration result comparison. First column: [Herrera C et al., 2012]. Second column: our method. Significant differences are labeled with green boxes. (**Best viewed on screen**)

to have a large displacement between the two cameras to cover a large region. Our method can also be adapted to other types of depth imaging sensors such as 3D LIDAR, ToF camera and etc. Additionally, as a single-shot method, our general formulation enables postprocessing of arbitrary single images, to, for example, insert graphical objects, providing some scene constraints in those images are known.

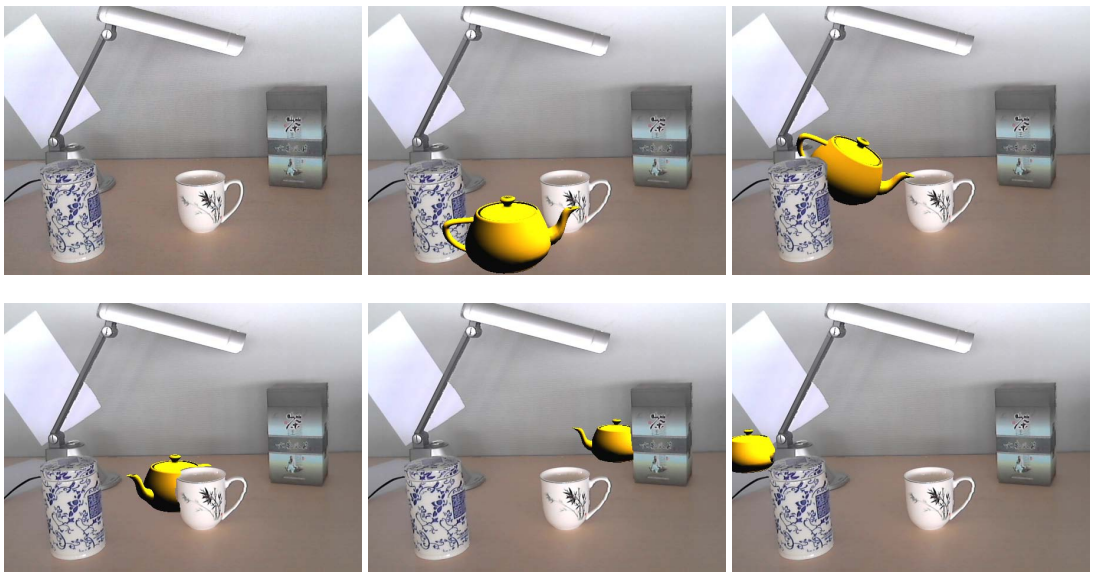


Figure 4.10: An augmented reality demonstration. The first image is the original image from the RGB camera. With the relative pose estimation result, we can fuse the depth information and color information precisely and the virtual teapot has then been added into the scene accurately. (**Best viewed on screen**)

The approach proposed in this chapter which directly minimizes the registration error in order to achieve relative pose estimation is conceptually novel. Using this approach could not only lead to a more efficient solution than traditional approaches but also achieve registration results which better conform to our visual evaluation.

In the current work, we only consider some explicit geometric constraints that can be manually extracted. In future, we plan to investigate incorporating implicit constraints to further automate the estimation process. For example, we could take into account the alignment of the discontinuities on the color and depth images.

Chapter 5

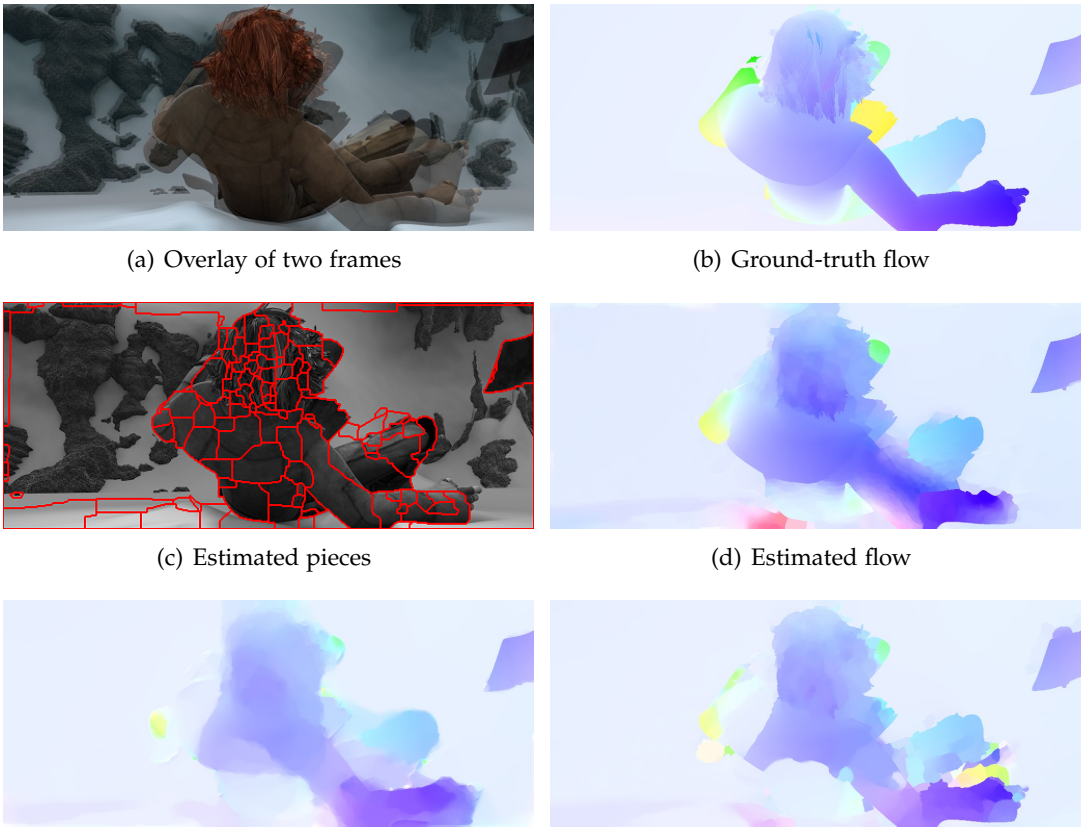
Piecewise Parametric Optical Flow Estimation

As a classic topic in computer vision, optical flow computation has attracted considerable attentions from the community. Remarkable progress has been made in the past decades, with high-performance optical flow algorithms available nowadays [Brox et al., 2004; Zach et al., 2007; Sun et al., 2014b; Xu et al., 2012; Kim et al., 2013]. Despite these successes, to obtain dense and accurate flow field remains challenging, especially for general dynamic scenes containing multiple complex, non-rigid objects,and/or large motions.

This chapter revisits the idea of piecewise parametric optical flow estimation popularized by Black *etc*. in the 1990s [Black and Jepson, 1996; Black and Anandan, 1996; Ju et al., 1996]. Unlike most modern optical flow techniques which capitalize on dense per-pixel flow vector estimation, these piecewise parametric flow methods assume a low-order parametric motion model within each segmented image piece. Using parametric models to represent a flow field, while is compact, can be rather restrictive. When the motion field is very complex, or when image segments do not conform well to motion segments, the parametrically-fitted flow field can be inaccurate or erroneous. Partly due to this reason, piecewise parametric models are seldom adopted by modern optical flow methods [Sun et al., 2014b].

In this chapter, we advocate that equipped with a carefully devised energy function and modern minimization techniques, the piecewise parametric model can be revitalized to achieve highly-accurate optical flow estimation with state-of-the-art performance.

Our motivation is described as follows. As in previous work [Black and Jepson, 1996; Ju et al., 1996], we assume that a flow field can be jointly represented by multiple parametric motion models in a piecewise fashion. To ease description, the 8-dof homography transformation model is used. To achieve accurate model fitting or approximation, we allow the size and shape of each piece to change adaptively. For example, some pieces must be large to account for large regions with homogeneous motion vectors to gain fitting robustness, while others need to be small enough to capture fine motion details within a small region containing complex motions. The approach of this chapter is to determine each piece appropriately, and at the same



(a) Overlay of two frames (b) Ground-truth flow

(c) Estimated pieces (d) Estimated flow

(e) Result of [Brox and Malik, 2011] (f) Result of [Xu et al., 2012]

Figure 5.1: The proposed method estimates optical flow using piecewise parametric (homography) models. In this example it yields accurate motion estimate on the actor's shoulder and back compared to LDOF [Brox and Malik, 2011] and MDP-OF [Xu et al., 2012].

time to fit a parametric model to each piece (see Figure 5.1 for an illustration). In light of this, the proposed method is similar to the joint motion estimation and motion segmentation scheme, as investigated in, e.g., [Cremers and Soatto, 2005; Birchfield and Tomasi, 1999; Sun et al., 2010b, 2012; Unger et al., 2012].

However, there is a subtle but critical difference. In contrast to the above methods which aimed to segment a motion field into a few independently moving regions [Cremers and Soatto, 2005; Birchfield and Tomasi, 1999; Sun et al., 2010b], our aim is to fit the entire flow field with a large number of (possibly hundreds of) piecewise parametric models. The proposed method can effectively handle complex motions which are challenging for the above methods such as [Birchfield and Tomasi, 1999; Unger et al., 2012].

5.1 Related work

There is a large volume of work on optical flow estimation. Below we mainly review the related methods for piecewise segmentation based and/or parametric flow estimation.

Computing parametric flow field on pre-segmented images is a classic idea [Black and Jepson, 1996; Ju et al., 1996; Xu et al., 2008; Lei and Yang, 2009]. Black and Jepson [1996] segment the image with color cue and fit variable-order motion model to each segment independently. Ju et al. [1996] divide the image evenly into rectangular patches, and fit affine models to them. Interactions between neighboring patches are involved and defined to be the difference of model parameters. Xu et al. [2008] fit affine models on regions segmented with color cue and initial flow; the fitting is regularized with Total Variation of the flow field. Lei and Yang [2009] represent the image with region tree built from color segmentation; constant flow vector is computed for each region.

Another category of related methods first estimates candidate parametric models then assign these models to each pixel as a segmentation process, e.g., [Wills et al., 2003; Bhat et al., 2006; Chen et al., 2013; Vogel et al., 2013]. Wills et al. [2003] use multiple homographies fitted from feature matches for segmentation, and Bhat et al. [2006] use both homographies and fundamental matrices. Recently, Chen et al. [2013] use translation and similarity transformations extracted from nearest neighbour field for segmentation. In scene flow estimation, Vogel et al. [2013] assign each pixel a segment, and each segment a 3D plane; the plane candidates are fitted based on an input scene flow estimate.

Methods have been proposed for joint motion segmentation and estimation [Mémin and Pérez, 2002; Cremers and Soatto, 2005; Roussos et al., 2012; Zitnick et al., 2005; Birchfield and Tomasi, 1999; Yamaguchi et al., 2013; Unger et al., 2012]. For example, Mémin and Pérez [2002] proposed such an approach in a variational framework, however the energy is defined on incremental motion field during the coarse-to-fine processing. Cremers and Soatto [2005] developed a variational approach to jointly estimate segmentation boundaries and affine models via continuous optimization. Roussos et al. [2012] represent and estimate dense motion field via multiple fundamental matrices plus an inverse-depth field.

Layered model estimation is another useful technique for motion segmentation and estimation [Wang and Adelson, 1994; Sun et al., 2010b, 2012]. This approach estimates a few overlapping motion layers, typically represented by parametric models, and assigns pixels to these layers. The pioneer work of Wang and Adelson [1994] uses affine layers to represent the motion field, and recent advances by Sun et al. [2010b, 2012] use affine motion to regularize the flow in each layer. The motivation of these approaches and their formulations are different from ours.

The proposed method is related to the methods based on over-parameterization [Nir et al., 2008; Hornáček et al., 2014]. Nir et al. [2008] represent optical flow with parametric (e.g., affine and rigid) model defined on every pixel. The work of Hornáček et al. [2014] defines per-pixel homography for flow estimation. In contrast to point-

wise parametric model, our method fits piecewise constant parametric models on adaptive segments.

The optimization scheme we use is similar to the multi-model fitting work [Isack and Boykov, 2012], and other relevant works in different domains, e.g., [Russell et al., 2011; Olsson and Boykov, 2012]. Compared to [Isack and Boykov, 2012] where scattered data is fitted to each model independently, we deal with dense, regular image grids where the models interact with each other to address the spatial continuity of flow field. Our idea of adaptively changing the domains of image pieces is partly related to the works of image quilting [Efros and Freeman, 2001] and photo autocollage [Rother et al., 2006].

5.2 Piecewise Parametric Flow Estimation

Given two images frames $\mathbf{I}_1$ and $\mathbf{I}_2$ as the reference frame and target frame respectively, our goal is to estimate a dense 2D displacement vector $\mathbf{u}$ at each pixel $\mathbf{x}$ on $\mathbf{I}_1$, based on the the brightness constancy assumption, i.e., $\mathbf{I}_1(\mathbf{x}) = \mathbf{I}_2(\mathbf{y})$ where $\mathbf{y} = \mathbf{x} + \mathbf{u}$. The displacement vector can be represented by a parametric transformation model T, i.e., , $\mathbf{y} = \mathrm{T} \circ \mathbf{x}$.

In this work, we choose to use the 8-dof homography as the parametric model, although other types of parametric models are also possible. One obvious benefit of choosing the homography model is, that homographies can be induced by 3D planes undergoing rigid motion. In fact, even for certain non-rigid motions or deformations, homography can be used as a good transformation model.

5.2.1 Energy function

Let $\mathcal{L} = \{1, .., K\}$ be a set of discrete labels representing the set of K homography models, i.e., $\mathcal{H} = \{\mathrm{H}_k\}, k = 1, \ldots, K$. Let Ω be the 2D image domain of $\mathbf{I}_1$, and $L : \Omega \rightarrow \mathcal{L}$ be a labeling function. Assigning label $k = L(\mathbf{x})$ to pixel $\mathbf{x}$ means that motion of $\mathbf{x}$ is induced by homography $\mathrm{H}_k \in \mathcal{H}$.

Our energy function is defined on both the unknown piecewise parametric models $\mathcal{H}$, and the unknown pixel labelling L, as

$$E(\mathcal{H}, L) = E_D(\mathcal{H}, L) + \lambda_C E_C(\mathcal{H}, L) + \lambda_P E_P(L) + \lambda_M E_M(L), \tag{5.1}$$

where E_D is a data term, E_C is a flow continuity regularization term, E_P is a Potts model term, and E_M is a *label cost* term [Li, 2007b; Delong et al., 2012] reflecting the Minimum Descriptor Length (MDL) principle. The λs are weighting parameters. Note that, one homography model can be be assigned to multiple disjoint pieces, as this is beneficial to handle occlusion.

5.2.2 Data term

The data term E_D enforces the brightness constancy constraint, subjecting to the piecewise homography models as

$$E_D(\mathcal{H}, L) = \sum_{\mathbf{x}\in\Omega} |\mathbf{I}_1(\mathbf{x}) - \mathbf{I}_2(\mathrm{H}_{L(\mathbf{x})}\mathbf{x})|, \tag{5.2}$$

where $|\cdot|$ denotes the L_1 norm. For brevity, we slightly abuse notations hereafter: H needs to be understood as an operator rather than matrix; both $\mathbf{x}$ and H$\mathbf{x}$ represent inhomogeneous image coordinates.

To improve the robustness with respect to noise and illumination changes, we use a robustified data term as in [Brox et al., 2004; Bleyer et al., 2011]. The robust version takes into account of both brightness constancy constraint and gradient constancy constraint, in addition to the use of a robust estimator ρ_D:

$$\begin{aligned} E_D(\mathcal{H}, L) = \sum_{\mathbf{x}\in\Omega} \rho_D\big((1-\alpha)|\mathbf{I}_1(\mathbf{x}) - \mathbf{I}_2(\mathrm{H}_{L(\mathbf{x})}\mathbf{x})| \\ +\alpha|\nabla\mathbf{I}_1(\mathbf{x}) - \nabla\mathbf{I}_2(\mathrm{H}_{L(\mathbf{x})}\mathbf{x})|\big), \end{aligned} \tag{5.3}$$

where we choose ρ_D to be a truncating function as $\rho_D(\cdot) = \min(\cdot, \sigma_D)$ and σ_D is a scalar parameter.

5.2.3 Flow continuity (inter-piece compatibility) term

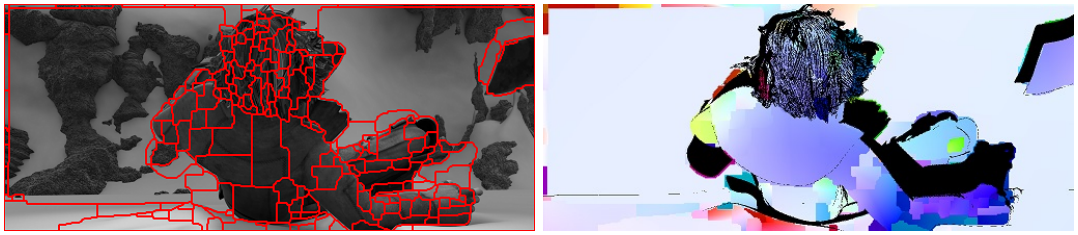
We introduce a flow continuity term E_C, which enforces the continuity constraint of the flow field, rather than the widely-used 1st-order or 2nd-order smoothness constraint (e.g., TV [Zach et al., 2007] or TGV [Braux-Zin et al., 2013] regularizer). Let $\mathcal{E}$ be the set of 4-connected pixel pairs on the image, E_C is defined to be

$$E_C(\mathcal{H}, L) = \sum_{(\mathbf{x},\mathbf{x}')\in\mathcal{E}} w(\mathbf{x}, \mathbf{x}') \cdot \rho_C\big(|\mathrm{H}_{L(\mathbf{x})}\bar{\mathbf{x}} - \mathrm{H}_{L(\mathbf{x}')}\bar{\mathbf{x}}|\big), \tag{5.4}$$

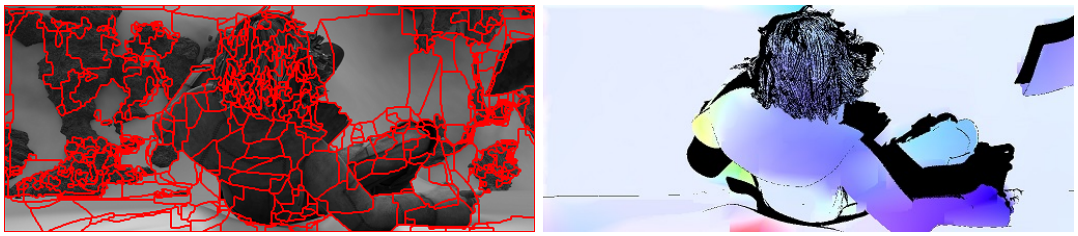
where $\bar{\mathbf{x}} = (\mathbf{x} + \mathbf{x}')/2$ is the midpoint of $(\mathbf{x}, \mathbf{x}')$, $\rho_C(\cdot) = \min(\cdot, \sigma_C)$ with σ_C a scalar parameter, and $w(\mathbf{x}, \mathbf{x}') = \exp(-\beta\|\mathbf{I}_1(\mathbf{x}) - \mathbf{I}_1(\mathbf{x}')\|)$ is a color-based weighting term. Note that if $L(\mathbf{x}) = L(\mathbf{x}')$, the cost at pixel-pair $(\mathbf{x}, \mathbf{x}')$ is nil. The properties of this term are analyzed as follows.

- E_C does not penalize the variations between neighboring pixels within a single piece (where all interior pixels have the same label), even if the variations are large. It only penalizes motion discontinuities at inter-piece boundaries (hence we also call it the *inter-piece compatibility term*).

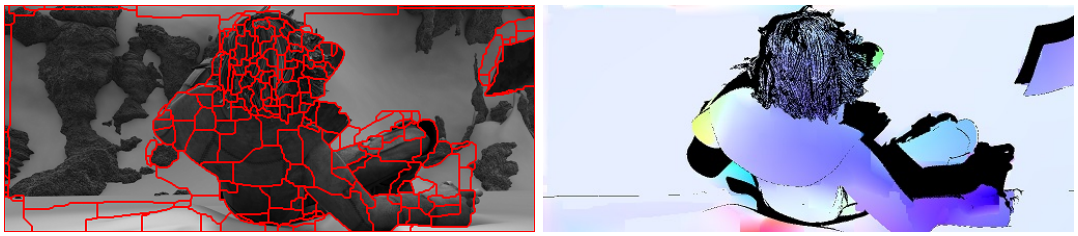
- The inter-piece motion discrepancies can be 0 or very small (i.e., the two adjacent pieces are compatible) even if their homography models differ a lot. Thus E_C allows model-switch, which is important for handling complex motion.



(a) Results without flow continuity term E_C



(b) Results without Potts model term E_P



(c) Results with both E_C and E_P

Figure 5.2: Effects of energy terms E_C and E_P. **Top row:** without E_C, the estimated flow contains many gross errors on the foreground human body with complex motion. (Occluded regions are masked black) **Middle row:** without E_P, the background regions with homogeneous motion are not well grouped, leading to 0.05∼0.1 endpoint error increase for them. **Bottom row:** with both the two terms, the method handles well both complex and homogeneous motions.

- It is easy to see that E_C is a submodular function in terms of discrete labeling variables L, which is a nice property for discrete energy minimization [Kolmogorov and Zabin, 2004].

The effect of this term is illustrated in Figure 5.2. It can be seen that without E_C the estimated flow contains many sharp discontinuous and gross errors.

5.2.4 Potts model term

In addition to the pairwise flow continuity term E_C, we use a pairwise Potts model term E_P to encourage spatially coherent labeling. This term is defined only on the discrete labeling variables as $E_P(L) = \sum_{(\mathbf{x},\mathbf{x}')\in\mathcal{E}} \delta\big(L(\mathbf{x}) \neq L(\mathbf{x}')\big)$, where $\delta(\cdot)$ is the 0-1 indicator function which takes 1 if the input argument is true, and 0 otherwise.

The terms E_C and E_P have different effects, and are complementary to each other. E_P enforces *intra-piece model constancy*; it penalizes any model change, no matter how similar the two models are. In contrast, as mentioned before, E_C enforces *inter-piece motion compatibility*; it allows compatible model switch, no matter how different the two models are (cf. Section 5.2.3).

Figure 5.2 illustrates that, without E_P the regions with homogeneous motion are not well grouped. This may lead to inferior flow estimate for these regions. Moreover, this can also be harmful to other regions: a model can be accidently assigned to many small pieces during labeling, bringing in difficulties for model estimation (cf. Section. 5.3.1).

5.2.5 MDL term

To reduce the redundancy of the fitted homographies, we employ an MDL term E_M to penalize the total number of the used homography models, i.e., $E_M(L) = \sum_{k=1}^{K} \tau(k)$, where $\tau(k) = \begin{cases} 1, & \text{if} \sum_{\mathbf{x}\in\Omega} \delta(L(\mathbf{x}) = k) > 0 \\ 0, & \text{othewise} \end{cases}$.

This term is helpful especially when a prior knowledge exists that the flow field can be well approximated by a relatively small number of parametric models. For example, in some man-made scenes where there are large planar structures, this term helps encourage fewer homographies and increase fitting quality. One may set its weight λ_M to 0 or very small if no prior is given.

5.3 Optimization

This section presents our optimization techniques. We first present the alternation based optimization assuming initial parameters given, then show our initialization method.

5.3.1 Alternation

The energy defined in (5.1) involves both discrete variables L and continuous variables $\mathcal{H}$. We approach this discrete-continues problem similarly to the multi-model fitting method of Isack and Boykov [2012]. A block coordinate descent (see Algorithm 5.1) is used that alternates between optimizing over L and $\mathcal{H}$, thus splitting the original problem into two sub-problems described as follows.

I. Labeling: Solve for L with fixed $\mathcal{H}$. With fixed homographies, the energy minimization reduces to a labeling problem with the energy

$$E(L) = \underbrace{E_D(L)}_{\substack{\text{Unary} \\ \text{potential}}} + \underbrace{\lambda_C E_C(L) + \lambda_P E_P(L)}_{\substack{\text{(Submodular) Pairwise} \\ \text{potential}}} + \underbrace{\lambda_M E_M(L)}_{\substack{\text{MDL} \\ \text{potential}}}. \tag{5.5}$$

Algorithm 5.1: Piecewise Homography Flow

Initialize $\mathcal{H}, L$.
while *not converge* **do**
  Fix $\mathcal{H}$, solve for L in (5.5) via graph-cut [Delong et al., 2012].
  Fix L, solve for $\mathcal{H}$ via Algorithm 5.2.
end

Algorithm 5.2: Piecewise Homography Fitting

Sort the input homographies $\mathrm{H}_k, k=1,...,K$ according to their labeling area in L in descending order.
for $iteration = 1, \ldots, m$ **do**
  for $k = 1 : K$ **do**
    Optimize H_k in (5.6) by simplex downhill [Nelder and Mead, 1965].
  end
end

Without the MDL term, the energy corresponds to a standard Markov Random Field (MRF) problem with unary and pairwise potentials. The α-expansion based graph-cut method [Boykov et al., 2001] can be used for fast approximate energy minimization. We use the method of [Delong et al., 2012] to handle the label costs in the MDL term.

A large set of homography models (e.g., , 1,000 in our experiments) are generated during initialization (See Section 5.3.2). For the sake of computational efficiency, if a homography is not labeled to any pixel after one round of the labeling process of L, it will be removed from the candidate model set. Another strategy to speed up computation is restricting the α-expansion within a region of a limited radius on the image plane (e.g., $<$100 pixels).

II. Fitting: Solve for $\mathcal{H}$ with fixed L. The homography parameters $\mathcal{H}$ appear in the data term E_D and flow continuity term E_C. With fixed labeling, minimizing the energy function is an unconstrained continuous optimization problem. If $\mathcal{H}$ appears only in E_D, we can optimize the parameters of each homography independently. Unfortunately, it appears also in E_C which involves pairwise iterations between adjacent pieces.

To tackle this issue we propose to use an inner block coordinate decent procedure: the homography is optimized one by one, each time with other homographies fixed. The homography models with larger labeling areas are first optimized as they are generally less affected by E_C. See Algorithm 5.2 for details. When optimizing a homography H_k, the energy reads as

$$
\begin{aligned}
E(\mathrm{H}_k) &= E_D(\mathrm{H}_k) + \lambda_C E_C(\mathrm{H}_k) \\
&= \sum_{\mathbf{x}\in\Omega_k} \rho_D\big((1-\alpha)|\mathbf{I}_1(\mathbf{x}) - \mathbf{I}_2(\mathrm{H}_k\mathbf{x})| + \alpha|\nabla\mathbf{I}_1(\mathbf{x}) - \nabla\mathbf{I}_2(\mathrm{H}_k\mathbf{x})|\big) \\
&\quad + \lambda_C \sum_{(\mathbf{x},\mathbf{x}')\in\mathcal{E}_k} w(\mathbf{x},\mathbf{x}')\cdot\rho_C\big(|\mathrm{H}_k\bar{\mathbf{x}} - \mathrm{H}_{L(\mathbf{x}')}\bar{\mathbf{x}}|\big)
\end{aligned} \tag{5.6}
$$

where $\Omega_k = \{\mathbf{x}\in\Omega \mid L(\mathbf{x}) = k\}$, $\mathcal{E}_k = \{(\mathbf{x},\mathbf{x}')\in\mathcal{E} \mid L(\mathbf{x}) = k,\ L(\mathbf{x}') \neq k\}$, and other variants and functions are as in (5.3) and (5.4). We optimize (5.6) via the derivative-free Nelder–Mead Simplex Downhill method [Nelder and Mead, 1965]. Similar to [Zhang et al., 2014], the vertexes of a simplex are initialized with the homographies of adjacent pieces. We found this strategy to be very effective in reducing the energy.

5.3.2 Initialization

The above alternation-based optimization requires an initialization to start. We propose a simple and fast algorithm to generate initial homography proposals and labeling. We will first use PatchMatch [Barnes et al., 2009] to compute an initial correspondence field. The PatchMatch algorithm is efficient and can handle large motions. It has been utilized by many optical flow and stereo matching algorithms [Xu et al., 2012; Lu et al., 2013; Chen et al., 2013; Bleyer et al., 2011].

After obtaining an initial correspondence field, we use a region-grow like algorithm to extract candidate homographies and initial labeling: we use Direct Linear Transform (DLT) [Hartley and Zisserman, 2005] to fit homographies for small local regions (e.g., 5×5 windows), and grow the regions to consistent neighboring pixels for initial labeling. See Algorithm 5.3 for details of this procedure.

5.4 Post-processing

5.4.1 Occlusion handling

We detect occlusion based on the forward-backward flow consistency check, where the forward flow is the estimated motion field from $\mathbf{I}$ to $\mathbf{I}'$ and the backward flow is the estimated motion field from $\mathbf{I}'$ to $\mathbf{I}$. A pixel $\mathbf{x}$ on $\mathbf{I}$ will be considered as occluded after its motion onto $\mathbf{I}'$ if $\|\mathbf{x} - \mathrm{H}'_{l'}\mathrm{H}_l\mathbf{x}\| > \theta$, where $\mathrm{H}'_{l'}$ is the homography of the point $\mathrm{H}_l\mathbf{x}$ on the target image, and θ is a scalar threshold. For the detected pixels, we remove the data term, and label estimated homographies to them via graph-cut.

5.4.2 Refinement

To further improve the results for complex motion, small local deformation may be necessary to compensate the discrepancy between true flow field and the piecewise approximation. Therefore we use the publicly available code of the "Classic+NL-

Algorithm 5.3: Homography proposal generation and initial labelling

```
Initialize a dense motion field by e.g., PatchMatch [Barnes et al., 2009];
Initialize a label map with all pixels unlabelled;
l ← 0;
while unlabelled pixels exist do
    Pick out an unlabelled pixel x;
    Fit a homography H_l with points in a small (e.g., 5 × 5) window W_x
     centered at x using their initial motion vectors;
    Label unlabelled pixels in W_x with l and push them into queue Q;
    while Q is not empty do
        Pop-out a pixel p from Q;
        foreach q as p's unlabelled neighbour do
            if q's motion fits H_l then
                Label q with l and push it into Q;
            end
        end
    end
    l ← l+1;
end
if l > L_max (e.g., 1000) then
    Sort the labels according to their labeling areas;
    Set all pixels of the l − L_max labels with smallest areas as unlabelled, then
     label each of them with its nearest label on the image.
end
```

fast" method [Sun et al., 2014b] for flow refinement[1]. Note that we directly refine the flow on the original image scale and no coarse-to-fine pyramid structure is used.

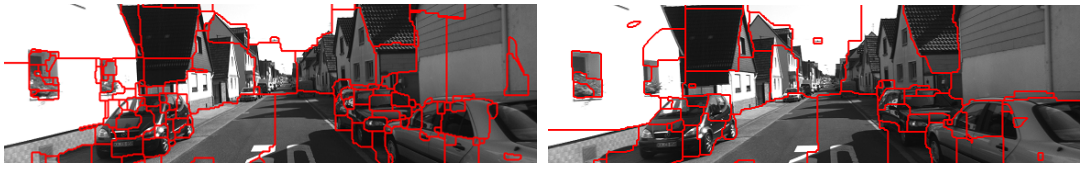
5.5 Experiments

In this section, we test the proposed method on three public benchmarks: the KITTI flow benchmark [Geiger et al., 2012a], the MPI Sintel benchmark [Butler et al., 2012], and the Middlebury flow benchmark [Baker et al., 2011b]. Our method is implemented in C++ & Matlab, and tested on a standard PC with Intel i7 3.4GHz CPU. In the following experiments, we set $\alpha = 0.9$, $\beta = 5$, $\sigma_D = 10$, iteration number of Algorithm 5.1 to be 5, maximal iteration of Algorithm 5.2 to be 15. Other parameters are trained on the benchmarks and will be explained in the corresponding sections. During initialization, we allow a maximum number of $1,000$ pieces. For KITTI and Sintel datasets, we half-size the images before running our method, and interpolate the estimated flow field back to the original size before refinement.

[1]The refinement using this method yields worse results in occluded regions; we keep the original flow for pixels that are very likely occluded (which failed the forward-backward check with a large threshold $\theta = 20$).

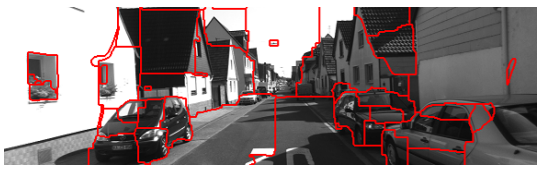
Table 5.1: End-point Error results on part of the *training* sequences in KITTI benchmark (3-pixel error threshold).

	Out-Noc	Out-All	Avg-Noc	Avg-All
Without MDL	5.94 %	11.44 %	1.58	3.69
Without refinement	5.76 %	10.84 %	1.41	3.00
Full	5.56 %	10.81 %	1.36	2.98



(a) $\lambda_M = 0$ (135 models)

(b) $\lambda_M = 300$ (56 models)



(c) $\lambda_M = 600$ (36 models)

(d) $\lambda_M = 900$ (27 models)

Figure 5.3: Effects of different MDL weights. Larger MDL weight leads to less homography models and larger pieces. We found that usually around $40 \sim 80$ homography models are adequate for flow estimation on KITTI benchmark.

5.5.1 Results on KITTI

The KITTI dataset is a challenging real-world dataset containing non-lambertian surfaces, different lighting conditions, and large displacement motions.

We first selected 20 image pairs with ground-truth flow fields from the *training* set. Based on the accuracy, we set $\lambda_C = 1$, $\lambda_P = 4$, $\lambda_M = 400$, $\sigma_C = 10$, and $\theta = 1.5$. The results on these images are shown in Table 5.1, where "Out-Noc" and "Avg-Noc" refer respectively to the outlier ratio and average end-point error in non-occluded regions and "Out-All" and "Avg-All" to all regions with ground-truth. The effect of the MDL term is obvious on this benchmark. Table 5.1 shows that the MDL term improves the results obtained without MDL term (i.e., $\lambda_M = 0$) by about $10\% \sim 20\%$. Figure 5.3 presents the estimated pieces with different MDL weights. We found that $40 \sim 80$ homography models are generally adequate for flow estimation on this dataset. Table 5.1 also shows that the refinement step improves the results by around $3\% \sim 5\%$.

We then ran our method on the *test* set where the ground-truth is hidden. Figure 5.4 shows two examples of our homography motion segmentation and flow estimation results. Note that both the large surfaces of road, green belt, building facades, cars, *etc.*, and the small objects such as road lamp and sign are well segmented. Table 5.2 compares our results with state-of-the-art two-frame optical flow methods. At

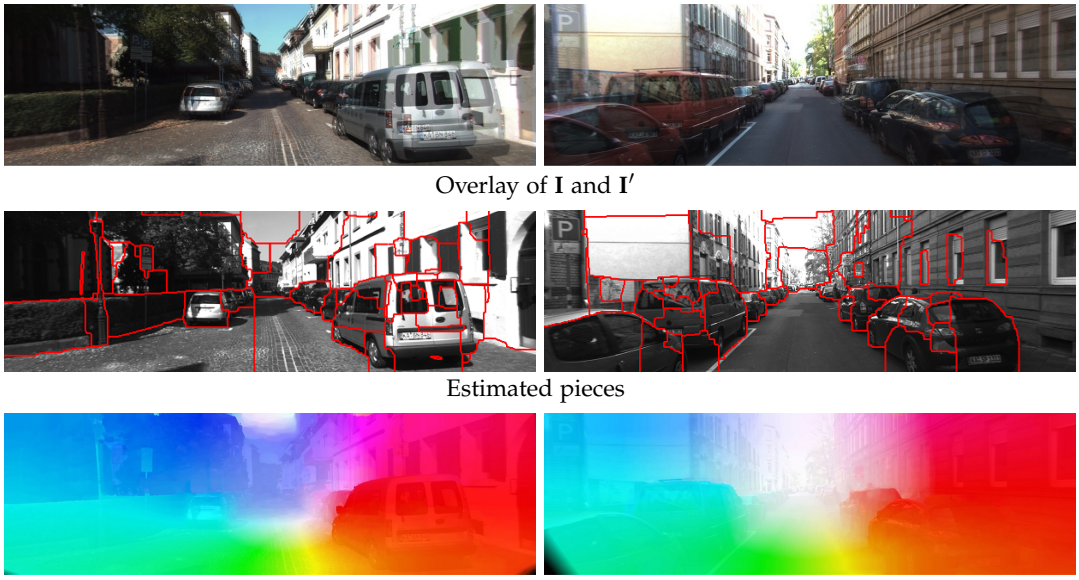


Figure 5.4: Example results of our method on KITTI benchmark. The top row overlays the input two images. The middle row shows the estimated pieces. The last row shows the estimated flow (with color coding of the benchmark). The flow images are overlaid on the first frames for better visualization. Note that in the first example, motions of small objects such the road lamp and sign are successfully estimated.

Table 5.2: Comparison with existing two-frame optical flow methods on the *test* set of the KITTI benchmark.

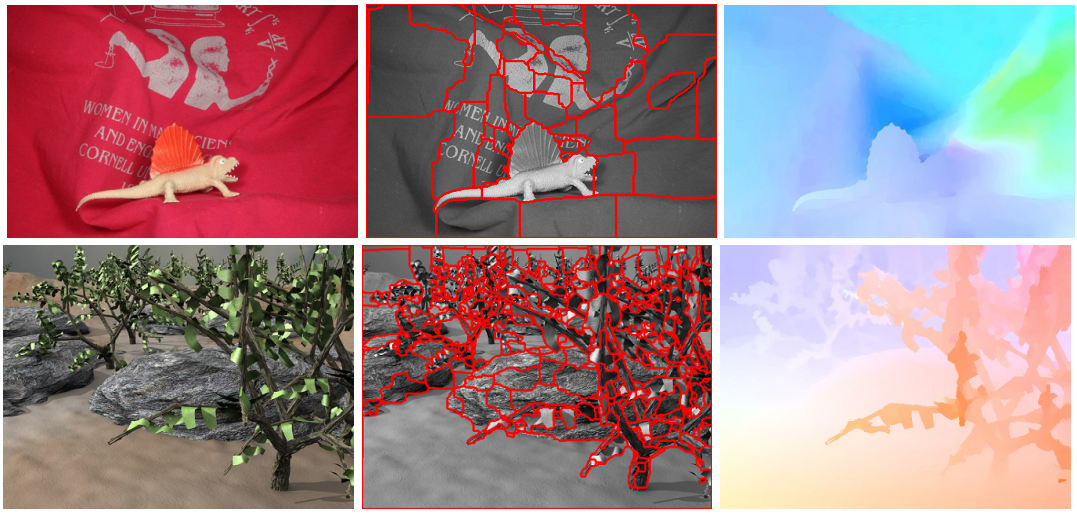
	> 2 pixels		> 3 pixels		> 4 pixels		> 5 pixels		End-Point	
	Out-Noc	Out-All	Out-Noc	Out-All	Out-Noc	Out-All	Out-Noc	Out-All	Avg-Noc	Avg-All
Our method	8.04%	**13.76%**	**5.76%**	**10.57%**	**4.64%**	**8.84%**	**3.93%**	**7.72%**	**1.3px**	2.9px
NLTGV-SC [Ranftl et al., 2014]	**7.64%**	14.55%	5.93%	11.96%	5.08%	10.48%	4.50%	9.42%	1.6px	3.8px
TGV2ADCSIFT [Braux-Zin et al., 2013]	8.04%	17.87%	6.20%	15.15%	5.24%	13.43%	4.60%	12.17%	1.5px	4.5px
BTF-ILLUM [Demetz et al., 2014]	8.84%	14.14%	6.52%	11.03%	5.38%	9.29%	4.64%	8.11%	1.5px	**2.8px**
DeepFlow [Weinzaepfel et al., 2013]	9.31%	20.44%	7.22%	17.79%	6.08%	16.02%	5.31%	14.69%	1.5px	5.8px
Classic+NL [Sun et al., 2014b]	12.94%	23.50%	10.49%	20.64%	9.21%	18.80%	8.36%	17.42%	2.8px	7.2px
EPPM [Bao et al., 2014]	17.49%	28.07%	12.75%	23.55%	10.22%	20.85%	8.58%	18.87%	2.5px	9.2px
LDOF [Brox and Malik, 2011]	24.43%	33.89%	21.93%	31.39%	20.22%	29.58%	18.83%	28.07%	5.6px	12.4px

the time of evaluation[2], our method is ranked the first among all published methods, under the by default 3-pixel threshold metric. In fact, our method shows improved performance on almost all the reported metrics used in KITTI.

[2]The KITTI benchmark receives the flow estimates for the *test* set submitted by a user, evaluates them, and publishes the errors online for method comparison. So do Middlebury and MPI Sintel.

Table 5.3: Comparison of endpoint errors on the *training* set of Middlebury benchmark.

	Dimetrodon	Grove2	Grove3	Hydrangea	RubberWhale	Urban2	Urban3	Venus
Ours	0.118	0.095	0.445	**0.146**	**0.072**	**0.196**	0.671	0.224
Ours w/o refinement	0.125	0.148	0.537	0.150	0.089	0.275	0.940	**0.190**
Classic+NL [Sun et al., 2014b]	**0.115**	**0.091**	**0.438**	0.154	0.077	0.207	**0.377**	0.229
Hornáček [Hornáček et al., 2014]	0.169	0.184	0.517	0.222	0.114	0.300	0.905	0.342



Input image **I** Estimated pieces Estimated flow

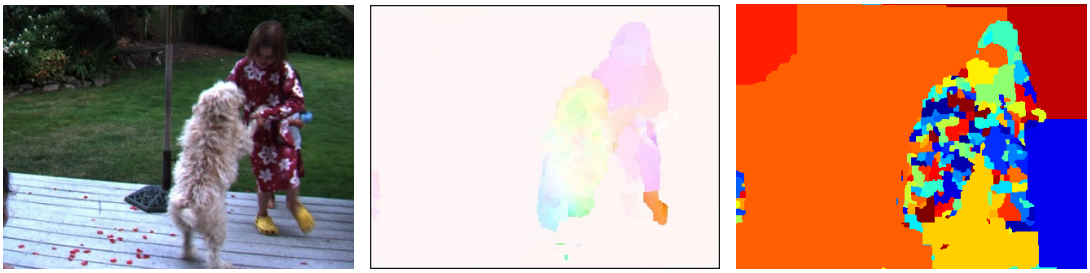
Figure 5.5: Qualitative results on "Dimetrodon" and "Grove3" sequences of the Middlebury benchmark. (The ground-truth flow fields can be found in Figure 5.8)

5.5.2 Results on Middlebury

The Middlebury optical flow benchmark only contains relatively small displacements. It has been extensively studied in recent years and sub-pixel accuracy has been achieved. However the motion is complex, e.g., there are many non-rigid deformations, making it difficult for parametric model based methods.

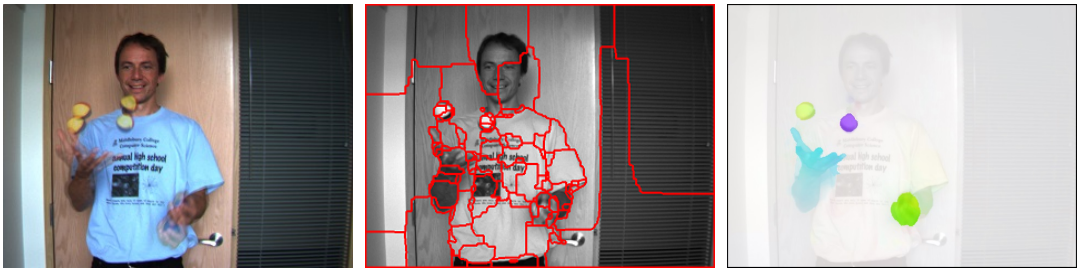
We tuned the parameters on the *training* set, ending up with $\lambda_C = 0.5$, $\lambda_P = 2$, $\lambda_M = 100$, $\sigma_C = 100$ and $\theta = 1$. The MDL weight is tuned to be much smaller than that on the KITTI benchmark, as there are many complex motions and small objects, necessitating more homography models. Table 5.3 shows our results with and without refinement, compared to the Classic+NL method [Sun et al., 2014b], and per-pixel homography estimation method [Hornáček et al., 2014]. In general, our final results are comparable to [Sun et al., 2014b] on the *training* set. Compared to [Hornáček et al., 2014], without refinement, our method outperforms [Hornáček et al., 2014] in 6 out of the 8 sequences, and outperforms it on all the sequences after refinement. Figure 5.5 shows some qualitative results of our method. Visually

(a) Input image $\mathbf{I}$ (b) Flow of [Unger et al., 2012] (c) Labeling of [Unger et al., 2012]



(d) Input image $\mathbf{I}'$ (e) Our flow (unrefined) (f) Our labeling

Figure 5.6: Comparison with the method of [Unger et al., 2012] on the Middlebury "DogDance" sequence. Our method is less suffered from the complex non-rigid motion; the flow and labelling results are clear better than [Unger et al., 2012] (images reproduced from their paper).

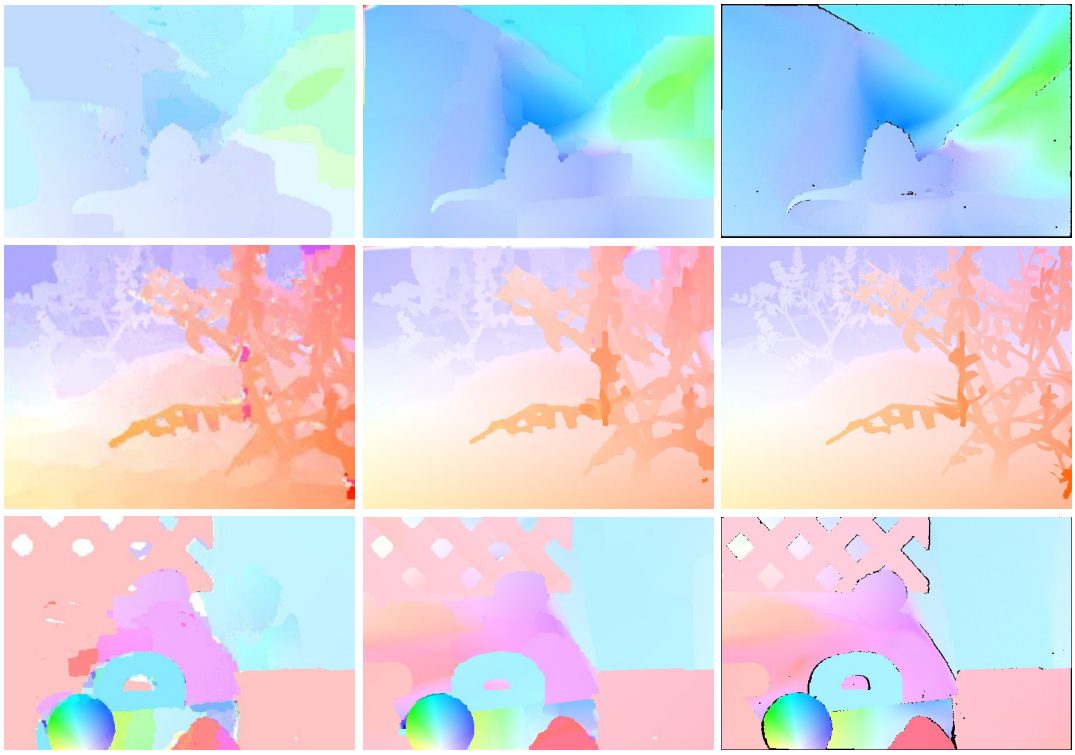


(a) Overlay of the input $\mathbf{I}$ and $\mathbf{I}'$ (b) Estimated pieces (c) Estimated flow (overlaid on $\mathbf{I}$)

Figure 5.7: Results of our method in the presence of large motions of small objects. Input images are from the "Beanbags" sequence of the Middlebury dataset.

inspected, it gives smooth and accurate flow fields. It is able to group large regions with homogeneous (homography) motion (e.g., the ground and rocks in "Grove3"), meanwhile segment out the small regions with complex motions (e.g., the leaves).

Figure 5.6 shows a challenging case (the "DogDance" sequence) with complex nonrigid motion. Our flow and segmentation are significantly better than [Unger et al., 2012], further demonstrating the ability of the proposed method in complex motion handling. Figure 5.7 shows the results of our method on the "Beanbag" sequence, which contains small objects with large motions. Figure 5.8 compares

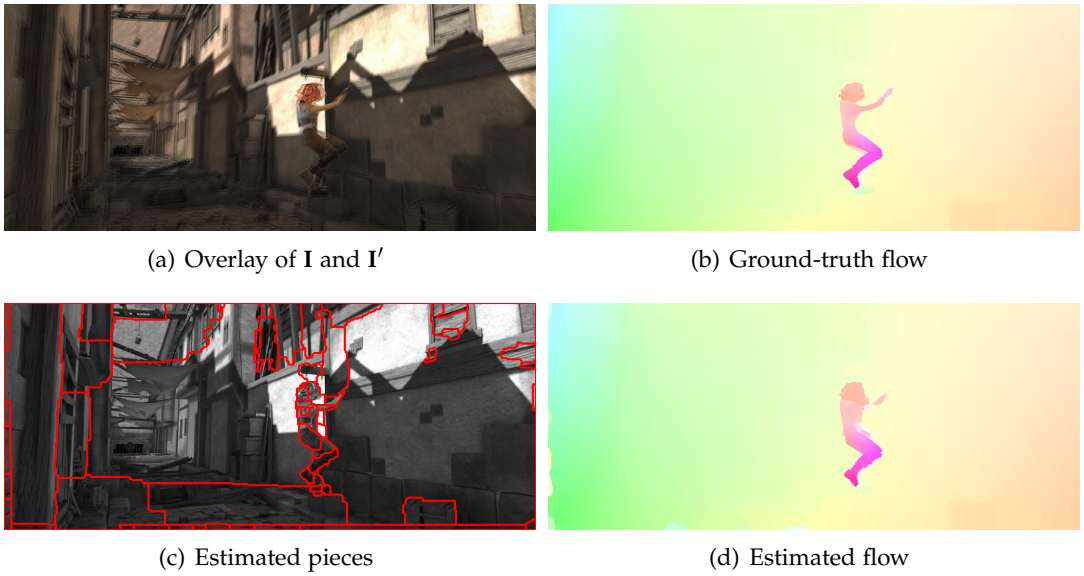


[Chen et al., 2013] (w/o refine.) Ours (w/o refinement) Ground truth

Figure 5.8: Qualitative comparison of [Chen et al., 2013] which uses global translation and similarity models (images reproduced from [Chen et al., 2013]), and our method. The flow fields shown here from both methods are without the refinement process. From top to bottom: "Dimetrodon", "Grove3" and "RubberWhale".

Table 5.4: Comparison of endpoint errors with existing methods on the *test* set of the Middlebury benchmark. The numbers in brackets show the rank of each method on each sequence. Results of [Unger et al., 2012] are reproduced from their paper.

	Army	Mequon	Schefflera	Wooden	Grove	Urban	Yosemite	Teddy
Ours	0.08 (7)	0.21 (27)	0.23(9)	0.16 (30)	0.56 (7)	0.30 (5)	0.15 (54)	0.43 (8)
Classic+NL [Sun et al., 2014b]	0.08 (7)	0.22 (33)	0.29 (25)	0.15 (19)	0.64 (18)	0.52 (48)	0.16 (65)	0.49 (19)
MDP-Flow2 [Xu et al., 2012]	0.08 (7)	0.15 (1)	0.20 (4)	0.15 (18)	0.63 (16)	0.26 (3)	0.11 (11)	0.38 (3)
NN-field [Chen et al., 2013]	0.08 (7)	0.17 (7)	0.19 (2)	0.09 (1)	0.41 (1)	0.52 (48)	0.13 (32)	0.35 (2)
Layer++ [Sun et al., 2010b]	0.08 (7)	0.19 (15)	0.20 (4)	0.13 (6)	0.48 (3)	0.47 (36)	0.15 (54)	0.46 (13)
Unger [Unger et al., 2012]	0.09	0.27	0.28	0.18	0.88	1.79	0.11	0.74



(a) Overlay of $\mathbf{I}$ and $\mathbf{I}'$ (b) Ground-truth flow

(c) Estimated pieces (d) Estimated flow

Figure 5.9: Sample results on the Sintel clean sequences.

the proposed method with method of [Chen et al., 2013] that uses translation and similarity models extracted from nearest neighbour fields. Visually inspected, our method yields smoother, and more accurate optical flow estimates.

Table 5.4 compares the performance of our method versus others on the *test* set. As can be seen, our results are comparable to state-of-the-art methods. Note that our results are superior to or on par with results of [Sun et al., 2014b] on all these sequences except for "Wooden", and outperform [Unger et al., 2012] on all the sequences except for "Yosemite".

5.5.3 Results on MPI Sintel

The Sintel benchmark contains long image sequences with large motions, severe illumination changes, and specular reflections. Moreover, it contains large numbers of non-planar surfaces and complex non-rigid deformations, making it more challenging for the proposed piecewise parametric method.

We selected 23 image pairs (1 pair per sequence) from the clean pass of *training* set to tune the parameter. The tuned parameters are $\lambda_C = 1$, $\lambda_P = 1$, $\lambda_M = 50$, $\sigma_C = 100$ and $\theta = 1.5$. The MDL is tuned to be very small due to the presence complex motions, e.g., the non-rigid motion of human and animal bodies. Figure 5.1 has shown a typical example and the result of our method, and an another example is presented in Figure 5.9.

We then ran the method on the *test* set, and Table 5.5 presents the results of our method, compared with a few state-of-the-art methods. At the time of evaluation, our method ranks 2nd, and outperforms all published methods on the clean pass. Note that it performs especially well on the occluded regions, thanks to the use of

Table 5.5: Comparison of end-point error with state-of-the-art methods on the *test* set of the Sintel benchmark. “all" / “noc" / “occ" indicate all / non-occluded / occluded regions respectively.

	Clean pass			Final pass		
	all	noc	occ	all	noc	occ
Our method	**4.388**	**1.714**	**26.202**	7.423	3.795	36.960
TF+OFM [Kennedy and Taylor, 2015]	4.917	1.874	29.735	**6.727**	3.388	**33.929**
DeepFlow [Weinzaepfel et al., 2013]	5.377	1.771	34.751	7.212	**3.336**	38.781
MDP-Flow2 [Xu et al., 2012]	5.837	1.869	38.158	8.445	4.150	43.430
EPPM [Bao et al., 2014]	6.494	2.675	37.632	8.377	4.286	41.695
Classic+NL [Sun et al., 2014b]	7.961	3.770	42.079	9.153	4.814	44.509

parametric models in the post-processing stage. The proposed method performs inferiorly on the final pass, ranking 7th among all evaluated methods. We find that the synthetic atmospheric effects on the final sequences cause difficulties for both the PathMatch-based initialization and our main algorithm. However, on the clean sequences for which the brightness constancy constraints satisfy, our method consistently produces accurate estimates.

5.5.4 Running Time

The proposed method takes a few hundreds of seconds to estimate a forward flow field of size 640×480 in our experiment settings. The optimization time spent in Algorithm 5.1 is about $200 \sim 500$ seconds depending on the number of models. The initialization stage takes about 5 seconds and the refinement stage takes about 60 seconds.

5.6 Conclusion

We have presented a simple method for optical flow estimation using piecewise parametric model. Thanks to the new energy design and the joint discrete-continuous optimization, our method produces high-quality results that are superior to or comparable with state-of-the-art methods. We believe that piecewise parametric flow estimation deserves a position in highly accurate optical flow estimation.

In future, we would like to investigate high-order parametric models (e.g., cubic or bi-cubic model) since the proposed method is general enough, and try different initialization methods. Explicitly incorporating occlusion reasoning would be another interesting future work.

Chapter 6

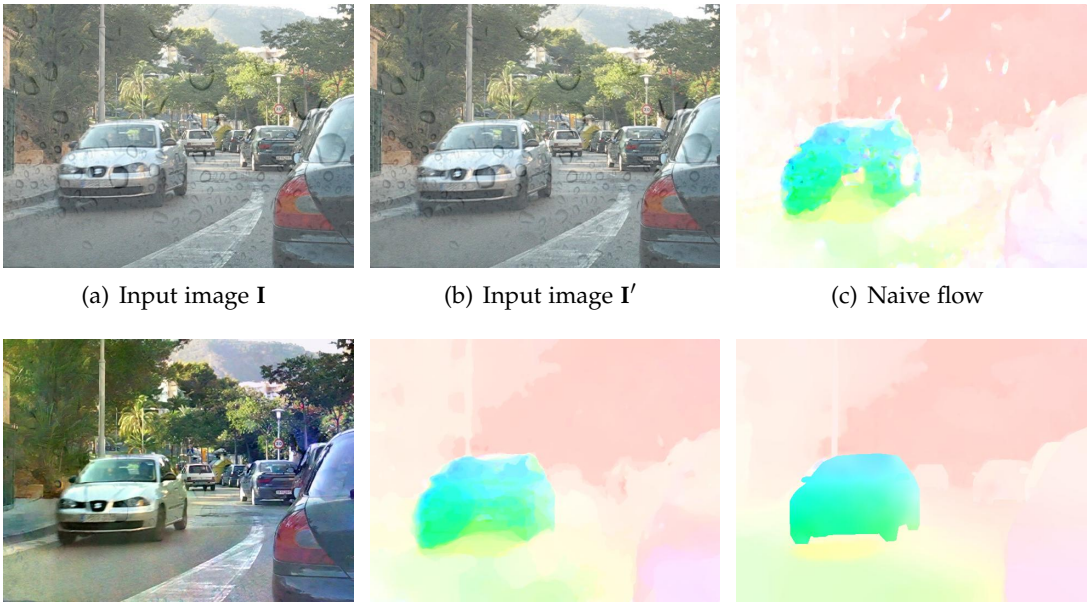
Layerwise Optical Flow Estimation under Transparency or Reflection

Most optical flow methods assume that there is only one imaging layer on the observed image with the brightness of scene objects, and use the brightness constancy constraint (BCC) to estimate the optical flow for scene objects. This single imaging layer assumption, however, can be often violated in real-world situations, especially in cases involving transparency or reflection. Transparencies and reflections are frequently met in the imaging process, e.g., when one is looking at a street scene from inside a car through a stained windscreen, or seeing through a thin layer of rain, looking into a window with semi-reflections on the window surface *etc*. The BCC will generally not hold for the resultant double-layer images, even in ideal noise-free cases. If the optical flow is estimated naively on the input images, the results will be erroneous. See Figure 6.1 for an example.

In the aforementioned situations, the observed image $\mathbf{I}$ can be modeled as a superposition of two constituting layers, denoted as $\mathbf{I} = \mathbf{L}_1 \oplus \mathbf{L}_2$, where $\oplus$ denotes some suitable layer combination operator. Without loss of generality, we call $\mathbf{L}_1$ the background scene layer, which corresponds to the image of the desired scene that we intend to capture, and $\mathbf{L}_2$ the foreground distracting layer, which corresponds to the semi-transparent media (e.g., a glass window with dirt or reflections on it) or the semi-reflected image.

The main goal of this chapter is to robustly estimate the optical flow field of the scene objects (i.e., the background layer), which is of concern for vision systems. We consider two general cases: the foreground distracting layer is *stationary*, or *dynamically changing*.

Let $\mathbf{I}$ and $\mathbf{I}'$ be two time-consecutive frames of a scene which contain the aforementioned two layers. In the presence of a dynamic foreground layer, there are two legitimate optical flow fields – one for the foreground layer and another for the background layer. Denote the two flow fields generated by the movements of the two layers as $\mathbf{U}$ and $\mathbf{V}$, respectively. The relationships among the observed images, the image layers, and the optical flow fields can be given as



(a) Input image $\mathbf{I}$ (b) Input image $\mathbf{I}'$ (c) Naive flow

(d) Estimated background layer (e) Estimated flow (f) GT flow

Figure 6.1: Illustration of the optical flow estimation problem under transparency[1]. In this example, the observed images in (a)(b) are corrupted by rain drops. Poor results will be obtained if directly estimating flow on the input images, as shown in (c). The proposed method estimates optical flow based on image layer severation. It can produce a robust optical flow estimate as in (e), meanwhile recover a clean background image layer as in (d).

$$\begin{array}{ccccc} \mathbf{I} & = & \mathbf{L}_1 & \oplus & \mathbf{L}_2 \\ & & \downarrow \mathbf{U} & & \downarrow \mathbf{V} \\ \mathbf{I}' & = & \mathbf{L}'_1 & \oplus & \mathbf{L}'_2 \end{array}$$

When $\mathbf{V} \equiv \mathbf{0}$ and $\mathbf{L}_2 \equiv \mathbf{L}'_2$, i.e., the foreground layer is static, our task is to estimate a single flow field $\mathbf{U}$ for background layer, and also estimate the layers $\mathbf{L}_1$, $\mathbf{L}'_1$, $\mathbf{L}_2$. Otherwise, when a dynamic foreground layer exists, we will estimate two flow fields $\mathbf{U}, \mathbf{V}$ as well as the layers $\mathbf{L}_1$, $\mathbf{L}'_1$, $\mathbf{L}_2$, $\mathbf{L}'_2$. As we explicitly perform image layer separation (i.e., estimating $\mathbf{L}_1$, $\mathbf{L}'_1$, $\mathbf{L}_2$, $\mathbf{L}'_2$), an appealing byproduct of our method is the restoration of the clear scene images.

For either of the two cases with a static or dynamic foreground layer, this is a highly ill-posed problem, especially considering optical flow estimation and image layer separation problems *per se* are known to be ill-posed. From only two input images, our task is to recover one or two optical fields, as well as the two unknown layers.

Little work has been reported in the literature concerning this double-layer im-

[1]The scene image is from [Liu et al., 2008].

age optical flow estimation problem, with only a few exceptions in the early days of computer vision research, e.g., [Shizawa and Mase, 1990, 1991; Langley et al., 1992b; Darrell and Simonecelli, 1993]. These works however often used over-simplified assumption and restrictive motion field models, such as assuming a constant flow field over time or space (e.g., globally translating). Bergen et al. [1992b] proposed a "three-frame algorithm" to recover two constituting flow fields, assuming the flow field is constant over at least three frames.

In contrast, we remove these restrictive assumptions, and propose a two-frame algorithm for robustly recovering the flow field(s). The proposed method works for generic motions and is thus applicable to a much wider range of practical situations for robust optical flow estimation.

6.1 Related Work

This chapter is concerned with optical flow estimation in double-layer images where both layers can possibly be moving. Despite that the phenomena of such multiple imaging layers and motions are frequently encountered in reality, few papers in the literature have been devoted to this topic. This is in a sharp contrast to the existence of a vast amount of papers on the classic optical flow problems (an analysis of recent practices of optical flow can be found in [Sun et al., 2014a]).

One of the first work for multiple optical flow computation is possibly due to Shizawa and Mase [1990, 1991]. By assuming the two underlying flow fields to be constant (e.g., pure translating), they derived a generalized brightness constancy constraint for the multi-motion case. However, this constant motion assumption is restrictive, not applicable for general flow fields with complex motions. Nevertheless, their method, being one of the first, has inspired a number of variants and extensions [Pingault and Pellerin, 2002; Auvray et al., 2009; Ramirez-Manzanares et al., 2006; Toro et al., 2000]. Some variants operate in the Fourier domain, e.g., [Langley et al., 1992a,b; Darrell and Simonecelli, 1993].

The flow estimation problem for two-layer images in this chapter should not be confused with those works concerning "motion-layer segmentation", albeit the two do share some similarity and the boundary between them can sometimes be fuzzy. For example, Wang and Adelson [1994] proposed to segment the image layers based on a pre-computed optical flow field. Irani et al. [1994] used temporal integration to track occluding or transparent moving objects with parametric motion. Black and others [Black and Anandan, 1996; Ju et al., 1996; Sun et al., 2010b; Wulff and Black, 2014] proposed a number of algorithms for multiple parametric motion estimation and segmentation. Yang and Li [2015] fit a flow filed with piecewise parametric models. Weiss [1997] presented a nonparametric motion estimation and segmentation method to handle generic smooth motions, thus this method is more related to ours. However, the method of Weiss and most other aforementioned methods primarily focused on image and motion segmentation, while we decompose the whole image into two composite brightness layers, and compute one generic flow field on

each layer.

The proposed method involves solving two tasks simultaneously: optical flow field estimation, and reflection/transparent layer separation. For the second task, many research works have been published previously. For example, Levin et al. [2002]; Levin and Weiss [2007] proposed methods for separating an image into two transparent layers using local statistics priors of natural images. Single image solutions are also investigated by Li and Brown [2014] and Yeung et al. [2008]. To utilize multiple frames, layer separation methods have been proposed based on aligning the frames with one layer [Wexler et al., 2002; Li and Brown, 2013; Guo et al., 2014] or multiple layers [Szeliski et al., 2000; Gai et al., 2012]. Sarel and Irani [2004] presented an information-theory based approach for separating transparent layers by minimizing the correlation between the layers. Chen et al. [2009] gave a gradient domain approach for moving layer separation which is also based on information theory. Schechner et al. [2000] developed a method for layer separation using image focus as a cue. By using independent component analysis, Farid and Adelson [1999] proposed a layer separation method which works on multiple observations under different mixing weights. Simon and Park [2015] proposed an average-image prior for reflection removal for in-vehicle black box videos. Techniques for image layer separation were also developed in the field of intrinsic image/video extraction [Tappen et al., 2005; Weiss, 2001; Ye et al., 2014].

In the context of stereo matching with transparency, Szeliski and Golland [1998] simultaneously recovered disparities, true colors, and opacity of visible surface elements. Tsin et al. [2006] estimated both depth and colors of the component layers. Li et al. [2015b] proposed a simultaneous video defogging and stereo matching algorithm.

The recent work of Xue et al. [2015] has a very similar formulation compared to ours. However, the goal and motivation of obstruction-free photography from a video sequence in [Xue et al., 2015] are different from ours. The underlying assumptions on the flow fields, the employed flow solvers, and the initialization techniques are dissimilar.

6.2 Problem Setup

For ease of presentation, in formulating the problem (Section 6.2 and Section 6.3) and presenting the optimization (Section 6.4), we will focus on the dynamic foreground case (i.e., double-layer flow estimation). The static foreground case (i.e., single-layer flow estimation) is simpler and can be derived accordingly. Note that, the static foreground case, though relatively simpler, is also of interest and very challenging.

6.2.1 Linear Additive Imaging Model

In the previous discussion, we simply used $\mathbf{I} = \mathbf{L}_1 \oplus \mathbf{L}_2$ to denote the layer superposition operation, but did not give its exact form. To make the idea more concrete, we opt for the linear additive model $+$ as a concrete example for $\oplus$, i.e., $\mathbf{I} = \mathbf{L}_1 + \mathbf{L}_2$.

The linear additive model itself, while simple, has been used successfully in the past in solving many vision problems involving transparency and reflection (e.g., in shadow removal [Yeung et al., 2008], image matting [Szeliski and Golland, 1998] and reflection separation [Li and Brown, 2014]). Moreover, by applying logarithm operation, a multiplicative superposition model can also be converted to an additive one.

Taking two frames of observations, $\mathbf{I}$ and $\mathbf{I}'$, at two consecutive time steps t and $t+1$, we have

$$\mathbf{I}(\mathbf{X}) = \mathbf{L}_1(\mathbf{X}) + \mathbf{L}_2(\mathbf{X}), \tag{6.1}$$
$$\mathbf{I}'(\mathbf{X}) = \mathbf{L}'_1(\mathbf{X}) + \mathbf{L}'_2(\mathbf{X}), \tag{6.2}$$

where $\mathbf{X}$ is a matrix indexing all pixel coordinates.

6.2.2 Double Layer Brightness Constancy

In the presence of transparencies or reflections, it is important to note that the conventional BCC condition cannot be applied directly to the observed images. Below, we will derive a generalized BCC condition which is applicable to the double-layer case.

The basic assumption that we will base our method on is: any component layer of the observed image must satisfy the brightness constancy condition individually. This is a realistic and mild assumption which is applicable to a wide range of transparency and reflection phenomena encountered in natural images. Cases that violate this basic assumption are deemed beyond the scope of this current work.

Suppose, during two small time steps, layer $\mathbf{L}_1$ changed to $\mathbf{L}'_1$ according to a motion field of $\mathbf{U}$, and layer $\mathbf{L}_2$ changed to $\mathbf{L}'_2$ according to a different motion field $\mathbf{V}$. Based on the assumption that the brightness of the objects in each individual layer is constant, we have

$$\mathbf{L}_1(\mathbf{X}) = \mathbf{L}'_1(\mathbf{X} + \mathbf{U}), \tag{6.3}$$
$$\mathbf{L}_2(\mathbf{X}) = \mathbf{L}'_2(\mathbf{X} + \mathbf{V}). \tag{6.4}$$

Together with the imaging model in (6.1) and (6.2), we call the above constraints the *generalized double-layer BCC condition* for an input double-layer image pair $(\mathbf{I}, \mathbf{I}')$.

6.2.3 The Double Layer Optical Flow Problem

Given the above linear additive imaging model as well as the generalized BCC conditions, we aim to recover both $\mathbf{L}_1$, $\mathbf{L}'_1$, $\mathbf{L}_2$, $\mathbf{L}'_2$ and $\mathbf{U}$, $\mathbf{V}$.

To make this severely ill-posed problem trackable, we adopt the energy minimization framework, and base it on the generalized BCC conditions as well as priors for optical flows and image layers. The energy function reads as

$$E = E_B + \lambda_L E_L + \lambda_F E_F, \tag{6.5}$$

where E_B corresponds to the double-layer BCC condition, E_L and E_F are the regularization terms (or prior terms) for the latent image layers, and the unknown optical flow fields, respectively. The λs are trade-off parameters.

In energy (6.5), we use $E_B = E_B(\mathbf{L}_1, \mathbf{L}_1', \mathbf{L}_2, \mathbf{L}_2', \mathbf{U}, \mathbf{V})$ to represent the BCC condition in the following way[2]:

$$E_B = \|\mathbf{L}_1(\mathbf{X}) - \mathbf{L}_1'(\mathbf{X}+\mathbf{U})\| + \|\mathbf{L}_2(\mathbf{X}) - \mathbf{L}_2'(\mathbf{X}+\mathbf{V})\|. \tag{6.6}$$

We use $\|\cdot\|$ to denote the ℓ_1-norm in this chapter unless otherwise specified. We choose to use ℓ_1-norm as the cost function mainly for its robustness [Brox et al., 2004; Zach et al., 2007] and its convenience in optimization. The two regularization terms E_L and E_F will be detailed in the following section.

6.3 Regularization

Using prior information as regularization is a common practice for solving ill-posed problems. In this work, the task is to separate the input frames into latent layers, and to recover the associated flow fields.

Priors are generally task-dependent. By enforcing different priors to latent layers and to optical flow fields, the algorithm can be adapted to solving different tasks. For example, if one knows the two latent layers are images of natural scenes, then the layers can be assumed to have sparse gradients (i.e., satisfying the well-known natural image priors). Moreover, for general optical flow fields, one can assume they are piecewise constant or piecewise smooth.

6.3.1 Natural Image Prior: Sparse Gradient

The research in natural image statistics shows that images of typical real-world scenes obey sparse spatial gradient distributions [Tappen et al., 2005; Levin and Weiss, 2007]. The distribution of a natural image $\mathbf{L}$ can often be modeled as a generalized Laplace distribution (*a.k.a.*, generalized Gaussian distribution), i.e.,

$$P(\mathbf{L}) \sim \prod_{\mathbf{x}\in\mathbf{X}} \exp(-|\partial_x \mathbf{L}(\mathbf{x})|^p - |\partial_y \mathbf{L}(\mathbf{x})|^p), \tag{6.7}$$

where the power p is a parameter usually within $[0.0, 1.0]$. A convenient choice is $p = 1$, with which the energy is reduced to the ℓ_1-norm of image spatial gradients. For ease exposition, we will let $p = 1$ in this work, though bear in mind that using other values of p is possible and may be advantageous in particular applications.

Taking the negative logarithm, the prior in (6.7) can be represented in the energy minimization form, i.e.,

$$\|\nabla \mathbf{L}(\mathbf{X})\| \to \min, \tag{6.8}$$

[2]For brevity, hereafter we use a short-hand notation: $\|f(\mathbf{X})\|$ and $\|f(\mathbf{X}) - g(\mathbf{X})\|$ should be understood as $\sum_{\mathbf{x}\in\mathbf{X}} \|f(\mathbf{x})\|$ and $\sum_{\mathbf{x}\in\mathbf{X}} \|f(\mathbf{x}) - g(\mathbf{x})\|$ respectively, where $f(\mathbf{x})$ and $g(\mathbf{x})$ are the layer brightness or gradient values at the pixel coordinate $\mathbf{x}$.

where $\nabla = (\partial_x, \partial_y)^\top$. Therefore, the latent layer regularization term $E_L(\mathbf{L}_1, \mathbf{L}'_1, \mathbf{L}_2, \mathbf{L}'_2)$ can be written as

$$E_L = \|\nabla \mathbf{L}_1(\mathbf{X})\| + \|\nabla \mathbf{L}'_1(\mathbf{X})\| + \|\nabla \mathbf{L}_2(\mathbf{X})\| + \|\nabla \mathbf{L}'_2(\mathbf{X})\|. \tag{6.9}$$

6.3.2 Optical Flow Priors: Spatial Smoothness

Early methods for solving multi-layer optical flow problem often made restrictive assumptions about the unknown flow fields. For example, [Bergen et al., 1992b] proposed a three-frame algorithm for recovering two component motion fields by assuming that the motion fields are constant over time, and [Shizawa and Mase, 1990] was built upon a local constant motion assumption to derive its basic equation. In this work, these restrictions are removed and the proposed method can handle more general and more complex motion fields.

We use a general assumption on flow field, namely, the optical flows are generally piecewise constant or piecewise smooth. To capture this prior, we adopt the total variation (TV) model [Zach et al., 2007] or total generalized variation (TGV) model [Bredies et al., 2010]. Specifically, a flow field $\mathbf{U}$ will be regularized by the following energy:

$$\|\mathbf{U}\|_{\mathrm{TGV}^k} \to \min, \tag{6.10}$$

where $\|\mathbf{U}\|_{\mathrm{TGV}^k} \doteq TGV^k(\mathbf{U}_x) + TGV^k(\mathbf{U}_y)$, and $TGV^k(\,\cdot\,)$ denotes the k-th order TGV measure for horizontal and vertical flow components $\mathbf{U}_x$ and $\mathbf{U}_y$.

In general, the k-th order TGV favors solutions that are piecewise composed of $(k-1)$-th order polynomials: with $k = 1$, TGV^1 reduces to the TV model which favors piecewise constant fields; with $k = 2$, TGV^2 favors piecewise affine fields. We will only consider TV and TGV^2 in this work, and the resultant prior regularization term $E_F = E_F(\mathbf{U}, \mathbf{V})$ for the flow fields can be written as

$$E_F(\mathbf{U}, \mathbf{V}) = \|\mathbf{U}\|_{\mathrm{TGV}^k} + \|\mathbf{V}\|_{\mathrm{TGV}^k}. \tag{6.11}$$

where $k = 1$ (i.e., TV) or 2.

6.4 Energy Minimization

6.4.1 The Overall Objective Function

By stacking all the constraints over both latent layers and flow fields, we reach an energy minimization problem as

$$\begin{aligned}\min E(\mathbf{L}_1, \mathbf{L}'_1, \mathbf{L}_2, \mathbf{L}'_2, \mathbf{U}, \mathbf{V}) &= E_B + \lambda_L E_L + \lambda_F E_F \\ &= \left(\|\mathbf{L}_1(\mathbf{X}) - \mathbf{L}'_1(\mathbf{X}+\mathbf{U})\| + \|\mathbf{L}_2(\mathbf{X}) - \mathbf{L}'_2(\mathbf{X}+\mathbf{V})\|\right) \\ &\quad + \lambda_L \left(\|\nabla \mathbf{L}_1\| + \|\nabla \mathbf{L}'_1\| + \|\nabla \mathbf{L}_2\| + \|\nabla \mathbf{L}'_2\|\right) \\ &\quad + \lambda_F \left(\|\mathbf{U}\|_{\mathrm{TGV}^k} + \|\mathbf{V}\|_{\mathrm{TGV}^k}\right), \end{aligned} \tag{6.12}$$

subject to

$$\mathbf{I} = \mathbf{L}_1 + \mathbf{L}_2, \quad \mathbf{I}' = \mathbf{L}'_1 + \mathbf{L}'_2, \tag{6.13}$$

$$\mathbf{0} \leq \mathbf{L}_2 \leq \min(\mathbf{I}, c), \quad \mathbf{0} \leq \mathbf{L}'_2 \leq \min(\mathbf{I}', c). \tag{6.14}$$

where the $\mathbf{X}$'s in the gradient terms of (6.9) are omitted for brevity.

Note that, to distinguish background and foreground layers, we introduce in (6.14) the element-wise bound constraints on the layers. We assume the foreground layer containing transparency or reflection has weaker signal, and use a small constant scalar c (e.g., $c = 0.25$ for brightness values in the range of [0,1]) as its brightness upper bound. This can be understood as an additional bound prior for layer separation. Also note that, putting aside (6.14), there is a global shift ambiguity for the layer values: adding an arbitrary scalar $s \in \mathbb{R}$ to $\mathbf{L}_1, \mathbf{L}'_1$ then $-s$ to $\mathbf{L}_2, \mathbf{L}'_2$ dose not change the energy in (6.12), nor dose it affect (6.13). This is because all the terms in (6.12) depend on value difference rather than absolute value. Nevertheless, both the lower and upper bounds in (6.14) help constrain the absolute values.

6.4.2 Alternated Minimization

To solve the above energy minimization problem, we first substitute the additive model constraints in (6.13) as hard constraints to eliminate $\mathbf{L}_1$ and $\mathbf{L}'_1$ in (6.12). Consequently, the energy function is now defined only on latent layers $\mathbf{L}_2, \mathbf{L}'_2$ and optical flows $\mathbf{U}, \mathbf{V}$.

Then, examining the energy form in (6.12), we notice that: *i*) the prior terms for optical flow field, i.e., E_F, is independent of the prior term for latent layers E_L; and *ii*) the BCC energy term E_B is the only term that links the flow estimation with latent layer separation. Based on these observations, we solve the minimization problem via block coordinate descent in an alternating fashion.

Specifically, starting from a proper initialization, our algorithm alternately solves the following two sub-problems:

- **(Layer Separation):** Given current flow field estimates $\{\mathbf{U}, \mathbf{V}\}$, solve for image layers $\{\mathbf{L}_2, \mathbf{L}'_2\}$ via the following minimization:

$$\min_{\mathbf{L}_2, \mathbf{L}'_2} \left(E_B(\mathbf{L}_2, \mathbf{L}'_2) + \lambda_L E_L(\mathbf{L}_2, \mathbf{L}'_2)\right). \tag{6.15}$$

- **(Flow Computation):** Given current image layers $\{\mathbf{L}_2, \mathbf{L}'_2\}$, estimate $\{\mathbf{U}, \mathbf{V}\}$ by solving the following two-layer optical flow problem:

$$\min_{\mathbf{U},\mathbf{V}} \left(E_B(\mathbf{U},\mathbf{V}) + \lambda_F E_F(\mathbf{U},\mathbf{V}) \right). \tag{6.16}$$

More details are given below.

6.4.2.1 Update the image layers

Given current optical flow estimates $\mathbf{U}$ and $\mathbf{V}$, the latent image layers $\mathbf{L}_2, \mathbf{L}_2^{'}$ can be updated by solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{L}_2,\mathbf{L}_2'} & \|(\mathbf{I}-\mathbf{L}_2)(\mathbf{X})-(\mathbf{I}'-\mathbf{L}_2')(\mathbf{X}+\mathbf{U})\|+\|\mathbf{L}_2(\mathbf{X})-\mathbf{L}_2'(\mathbf{X}+\mathbf{V})\| \\ & +\lambda_L\left(\|\nabla(\mathbf{I}-\mathbf{L}_2)\|+\|\nabla(\mathbf{I}'-\mathbf{L}_2')\|+\|\nabla\mathbf{L}_2\|+\|\nabla\mathbf{L}_2'\|\right) \\ & \text{subject to } \ \mathbf{0}\le\mathbf{L}_2\le\min(\mathbf{I},c),\ \mathbf{0}\le\mathbf{L}_2'\le\min(\mathbf{I}',c), \end{aligned} \tag{6.17}$$

This is a convex optimization problem defined on $\mathbf{L}_2$ and $\mathbf{L}_2'$, and the cost function can be arranged into

$$\begin{aligned} \min_{\mathbf{l}} & \|\mathbf{A}\cdot\mathbf{l}-\mathbf{b}\|, \\ & \text{subject to } \ lb_i \le l_i \le ub_i, \forall i \end{aligned} \tag{6.18}$$

where $\mathbf{A}$ and $\mathbf{b}$ encode all the ℓ_1 constraints on latent layers, which are extremely sparse (only a few elements in each row are non-zero). $\mathbf{l}$ is a column vector containing elements in $\mathbf{L}_2$ and $\mathbf{L}_2'$. lb_i and ub_i are constant bounds from (6.14). The constraints are linear function of the latent layers $\mathbf{L}_2$ and $\mathbf{L}_2'$, thus this problem can be solved as a linear programming using off-the-self solvers.

Nevertheless, to utilize the sparse structure in the problem and speed up the implementation, we solve the problem by using a tailored version of Iteratively Reweighted Least Squares (IRLS) [Chartrand and Yin, 2008]. With IRLS, one can also adapt the formulation to different priors readily, e.g., replacing ℓ_1-norm with ℓ_p-norm ($0<p<1$).

There are some issues need to be considered in using IRLS to solve our problem. As shown in (6.17) and (6.18), the solution vector is confined by both lower and upper bounds. To deal with these bounds, one may add a projection operator inside the iteration loop of IRLS to guarantee the solution bounded (see Line 5 in Algorithm 6.1). However, it can be seen from (6.17) that, when the bounds are ignored, there is a *offset scale ambiguity*: adding any constant scalar to $\mathbf{L}_2, \mathbf{L}_2'$ does not affect the objective function. To resolve this ambiguity, we shift the solution vector such that it minimizes the objective function after projection. The shifting scalar can be efficiently computed via a 1D search, as shown in Line 4 of Algorithm 6.1.

We found our modified IRLS algorithm outlined in Algorithm 6.1 to be both effective and efficient in solving the large-scale sparse linear problem.

Use of Color Images. The above formulations can be easily extended to color RGB images. With color images, the double-layer BCC term E_B and layer regularization

Algorithm 6.1: Iteratively reweighted least squares (IRLS) with shift-projection operation in updating latent layers

Input: $\mathbf{l}^{(0)}$
Output: $\mathbf{l}^{(n)}$
for $t = 1, \ldots, n$ **do**
  $\mathbf{w}^{(t)} = [\ldots, w_i^{(t)}, \ldots]$, where $w_i^{(t)} = |\mathbf{A}_i \cdot \mathbf{l}^{(t-1)} - b_i|^{-1}$ ▷ *Reweighting*
  $\mathbf{l}^{(t)} = (\mathbf{A}^{\mathbf{T}}\mathbf{W}^{(t)}\mathbf{A})^{-1}\mathbf{A}^{\mathbf{T}}\mathbf{W}^{(t)}\mathbf{b}$, where $\mathbf{W}^{(t)} = \mathrm{diag}(\mathbf{w}^{(t)})$ ▷ *Least square solver*
  $\mathbf{l}^{(t)} = \mathbf{l}^{(t)} + \mathbf{1} * \underset{s}{\mathrm{argmin}} \sum_i |\min(\max(l_i^{(t)} + s, ub_i), lb_i) - b_i|$ ▷ *Shift*
  $l_i^{(t)} = \min(\max(l_i^{(t)}, ub_i), lb_i), \forall i$ ▷ *Projection*
end

term E_L will be evaluated at R-G-B channels separately. The flow fields $\mathbf{U}$ and $\mathbf{V}$ are shared by all three channels.

6.4.2.2 Update the flow fields U and V

Given current layer estimates $\mathbf{L}_2, \mathbf{L}_2'$, and $\mathbf{L}_1 = \mathbf{I} - \mathbf{L}_2$, $\mathbf{L}_1' = \mathbf{I}' - \mathbf{L}_2'$, the next step is to update the associated two flow fields $\mathbf{U}$ and $\mathbf{V}$. This is done by solving the following optimization problem:

$$\begin{aligned}\min_{\mathbf{U},\mathbf{V}} &\|\mathbf{L}_1(\mathbf{X}) - \mathbf{L}_1'(\mathbf{X}+\mathbf{U})\| + \|\mathbf{L}_2(\mathbf{X}) - \mathbf{L}_2'(\mathbf{X}+\mathbf{V})\| \\ &+ \lambda_F \left(\|\mathbf{U}\|_{\mathrm{TGV}^k} + \|\mathbf{V}\|_{\mathrm{TGV}^k}\right). \end{aligned} \tag{6.19}$$

The computations for these two flow fields are in fact separable. This can be easily seen from the above optimization, as the cost function can be expressed as the sum of two terms, each of which can be solved in isolation, i.e., given $\{\mathbf{L}_1, \mathbf{L}_1', \mathbf{L}_2, \mathbf{L}_2'\}$, solve

$$\min_{\mathbf{U}} \|\mathbf{L}_1(\mathbf{X}) - \mathbf{L}_1'(\mathbf{X}+\mathbf{U})\| + \lambda_F \|\mathbf{U}\|_{\mathrm{TGV}^k}, \tag{6.20}$$

$$\min_{\mathbf{V}} \|\mathbf{L}_2(\mathbf{X}) - \mathbf{L}_2'(\mathbf{X}+\mathbf{V})\| + \lambda_F \|\mathbf{V}\|_{\mathrm{TGV}^k}. \tag{6.21}$$

To solve the above optical flow problems, we use quadratic relaxation and introduce an auxiliary flow field to decouple the BCC term and regularization term, similar to [Zach et al., 2007; Steinbrucker et al., 2009]. Taking the minimization of $\mathbf{U}$ in (6.20) for example, we introduce an auxiliary flow field $\mathbf{\Lambda}$, and relax (6.20) as

$$\min_{\mathbf{U},\mathbf{\Lambda}} \|\mathbf{L}_1(\mathbf{X}) - \mathbf{L}_1'(\mathbf{X}+\mathbf{\Lambda})\| + \sum_{i=1,2} \frac{1}{2\theta}(\mathbf{\Lambda}_i - \mathbf{U}_i)^2 + \lambda_F \sum_{i=1,2} \mathrm{TGV}^k(\mathbf{\Lambda}_i), \tag{6.22}$$

where θ is a small constant (0.2 in our implementation) such that $\mathbf{\Lambda}$ and $\mathbf{U}$ are close, $\mathbf{U}_1, \mathbf{\Lambda}_1$ are horizontal flows and $\mathbf{U}_2, \mathbf{\Lambda}_2$ are vertical flows. (6.22) is minimized via

Algorithm 6.2: The primal-dual algorithm to solve $\operatorname{argmin}_{\mathbf{U}} \frac{1}{2\theta}\|\mathbf{U}-\mathbf{\Lambda}\|_2^2 + \mathrm{TV}(\mathbf{U})$ in updating flow fields. Here $\mathbf{U}$ and $\mathbf{\Lambda}$ are the horizontal or vertical components of the flow filed. $\mathbf{P}$ is the dual variable of $\mathbf{U}$.

Input: $\mathbf{\Lambda}$
Output: $\mathbf{U}$

Set $\mathbf{U}^{(0)} = \bar{\mathbf{U}}^{(0)} = \mathbf{\Lambda},\ \mathbf{P}^{(0)} = \mathbf{0},\ \sigma = \tau = \frac{1}{\sqrt{8}}$
for $t = 1, \ldots, n$ **do**
  $\mathbf{P}^{(t)} = \mathcal{P}_{\|\cdot\|_2 \leqslant 1}(\mathbf{P}^{(t-1)} + \sigma\nabla\bar{\mathbf{U}}^{(t-1)})$, where $\left(\mathcal{P}_{\|\cdot\|_2\leqslant 1}(\hat{\mathbf{P}})\right)_{i,j} = \frac{\hat{\mathbf{P}}_{i,j}}{\max(1, \|\hat{\mathbf{P}}_{i,j}\|_2)}$
  $\mathbf{U}^{(t)} = \dfrac{\theta\mathbf{U}^{(t-1)} + \theta\tau\mathrm{div}(\mathbf{P}^{(t)}) + \tau\mathbf{\Lambda}}{\theta+\tau}$
  $\bar{\mathbf{U}}^{(t)} = 2\mathbf{U}^{(t)} - \mathbf{U}^{(t-1)}$
end

Algorithm 6.3: The primal-dual algorithm to solve $\operatorname{argmin}_{\mathbf{U}} \frac{1}{2\theta}\|\mathbf{U}-\mathbf{\Lambda}\|_2^2 + \mathrm{TGV}^2(\mathbf{U})$ in updating flow fields. $\mathbf{U}$ and $\mathbf{\Lambda}$ are the horizontal or vertical components of the flow filed. $\mathbf{P}$ and $\mathbf{Q}$ are the dual variables of $\mathbf{U}$ and $\mathbf{W}$, respectively.

Input: $\mathbf{\Lambda}$
Output: $\mathbf{U}$

Set $\mathbf{U}^{(0)} = \bar{\mathbf{U}}^{(0)} = \mathbf{\Lambda},\ \mathbf{W}^{(0)} = \bar{\mathbf{W}}^{(0)} = \mathbf{0},\ \mathbf{P}^{(0)} = 0,\ \mathbf{Q}^{(0)} = \mathbf{0},\ \sigma = \tau = \sqrt{\frac{2}{17+\sqrt{33}}}$
for $t = 1, \ldots, n$ **do**
  $\mathbf{P}^{(t)} = \mathcal{P}_{\alpha_1}(\mathbf{P}^{(t-1)} + \sigma(\nabla\bar{\mathbf{U}}^{(t-1)} - \bar{\mathbf{W}}^{(t-1)}))$, where $\left(\mathcal{P}_{\alpha_1}(\hat{\mathbf{P}})\right)_{i,j} = \frac{\hat{\mathbf{P}}_{i,j}}{\max(1, \|\hat{\mathbf{P}}_{i,j}\|_2/\alpha_1)}$
  $\mathbf{Q}^{(t)} = \mathcal{P}_{\alpha_0}(\mathbf{Q}^{(t-1)} + \sigma\nabla\bar{\mathbf{W}}^{(t-1)})$, where $\left(\mathcal{P}_{\alpha_0}(\hat{\mathbf{Q}})\right)_{i,j} = \hat{\mathbf{Q}}_{i,j} / \max(1, \|\hat{\mathbf{Q}}_{i,j}\|_2/\alpha_0)$
  $\mathbf{U}^{(t)} = \dfrac{\theta\mathbf{U}^{(t-1)} + \theta\tau\mathrm{div}(\mathbf{P}^{(t)}) + \tau\mathbf{\Lambda}}{\theta+\tau}$
  $\mathbf{W}^{(t)} = \mathbf{W}^{(t-1)} + \tau(\mathbf{P}^{(t)} + \mathrm{div}(\mathbf{Q}^{(t)}))$
  $\bar{\mathbf{U}}^{(t)} = 2\mathbf{U}^{(t)} - \mathbf{U}^{(t-1)}$
  $\bar{\mathbf{W}}^{(t)} = 2\mathbf{W}^{(t)} - \mathbf{W}^{(t-1)}$
end

alternately optimizing $\mathbf{U}$ and $\mathbf{\Lambda}$. When solving for $\mathbf{U}$, we use the first order Taylor approximation to linearize $\mathbf{L}_1'(\mathbf{X}+\mathbf{U})$ in (6.22) to get

$$\min_{\mathbf{\Lambda}} \|\mathbf{L}_1'(\mathbf{X}) + \langle\nabla\mathbf{L}_1'(\mathbf{X}), \mathbf{\Lambda}(\mathbf{X})\rangle - \mathbf{L}_1(\mathbf{X})\| + \sum_{i=1,2}\frac{1}{2\theta}\|\mathbf{\Lambda}_i - \mathbf{U}_i\|_2^2, \tag{6.23}$$

for which a closed-form solution for $\mathbf{U}$ can be obtained. Coarse-to-fine pyramid is used to ensure accurate linearization as in [Zach et al., 2007]. When solving for Λ_i, we opt for the recent first-order primal-dual technique in [Chambolle and Pock, 2011] to solve the problems of TV-ℓ_2 (i.e., $k=1$) and TGV^2-ℓ_2 (i.e., $k=2$). See Algorithm 6.2 and Algorithm 6.3 for the algorithms we applied for TV-ℓ_2 and TGV^2-ℓ_2, respectively.

6.5 Experiments

In this section, we validate the proposed model and framework, and evaluate the performance of our method. We report the experimental results on both synthetic data and real images (e.g., Middlebury [Baker et al., 2011a] and Sintel [Butler et al., 2012] flow datasets, and the reflection dataset in [Li and Brown, 2013]).

Evaluation metrics. To evaluate the performance of optical flow estimation, the average endpoint error (EPE) in pixel distance is used. When no ground truth flow is available, the image warping error is used similar to [Steinbrucker et al., 2009]. We will also qualitatively evaluate the obtained optical flow fields as well as the image separation results.

Initialization. Being an alternated method, the proposed algorithm requires an initialization to start the alternation. One can start from either an initial optical flow estimation or from an initial layer separation. The latter one is used in our experiments, and the initialization details will be given later in the experiments.

Parameters. In the following experiments, the weights of the priors, i.e., λ_L, λ_F, are roughly tuned according to the results. Both TV and TGV^2 flow regularizers worked well, consistently improving the accuracy upon initialization. In the following, we present the results using TV (i.e., $k = 1$).

6.5.1 Static Foreground Cases

We start from the simpler case where only the background layer $\mathbf{L}_1$ is dynamically changing by an unknown motion field $\mathbf{U}$, while the foreground layer is static (i.e., $\mathbf{L}_2 \equiv \mathbf{L}_2'$ and $\mathbf{V} \equiv \mathbf{0}$). The task is to estimate flow field $\mathbf{U}$ and component layers $\mathbf{L}_1, \mathbf{L}_1', \mathbf{L}_2$. Again, we would like to emphasize that, even though we call it the "simpler case", to jointly estimate an accurate flow field and recover latent layers remains a challenging task. To the best of our knowledge, there was no previous method that recovers both a complex dense flow field under transparency/reflection and separates the two constituting layers.

In the following tests, a rather conservative strategy is used to initialize the proposed method: we initiate the static foreground image $\mathbf{L}_2$ to be all zeros. Consequently, in the beginning of the optimization, we compute an initial optical flow field naively based on the two input images.

Seeing through rain is a practical situation where measures should be taken to avoid the rain ruining vision systems. In the first test, we first synthesized a scene by superimposing a static rain image over the pair of Dimetrodon in the Middlebury dataset. Gray images were used. As illustrated in Figure 6.2, within about 25 iterations, the optical flow estimation error has been decreased from about 1.0 pixels to about 0.3 pixels. This demonstrates the advantage of our formulation for robust optical flow estimation. The qualitative results are demonstrated in Figure 6.3.

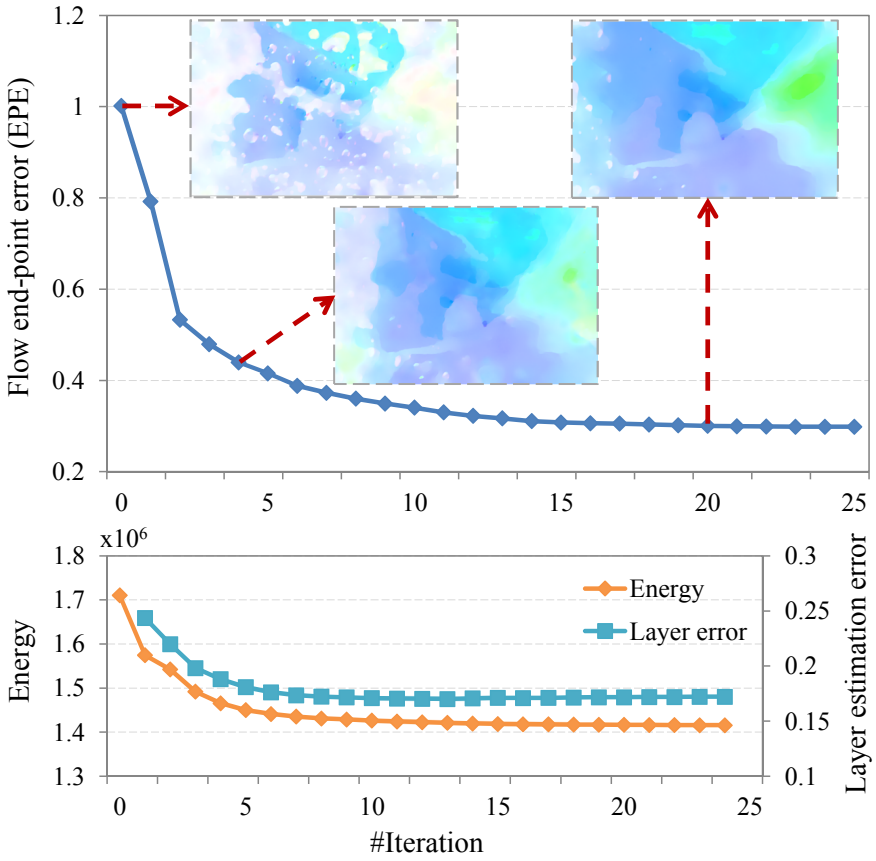


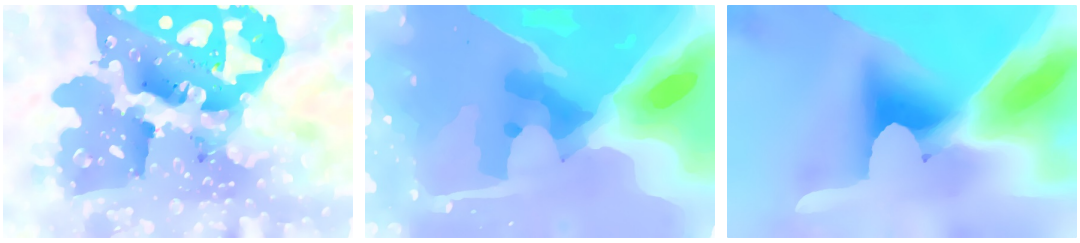
Figure 6.2: Convergence of the proposed method. Top: optical flow estimation error (EPE) *w.r.t.* iterations. Bottom: energy and layer estimation errors *w.r.t.* iterations. The layer error is evaluated as $1 - NCC(\text{GT } \mathbf{L}_2, \text{estimated } \mathbf{L}_2)$.

Table 6.1: Mean flow EPE for three Sintel image sequences superimposed with the static rain image. Oracle flows are computed with clean background images.

Sequence	Naive flow	Our flow	Oracle
"alley1"	0.49	**0.35**	0.22
"sleeping1"	0.80	**0.33**	0.12
"sleeping2"	0.26	**0.21**	0.07

Additionally, we overlay the rain image with three color image sequences from the Sintel dataset. We evenly sampled 10 images from the "alley 1", "sleeping 1" , and "sleeping 2" sequences respectively, and Table 6.1 shows that the proposed method has clearly reduced the mean EPE of initial flows. Two typical results are shown in Figure 6.4.

(a) Input $\mathbf{I}$ (b) Input $\mathbf{I}'$ (c) GT $\mathbf{U}$



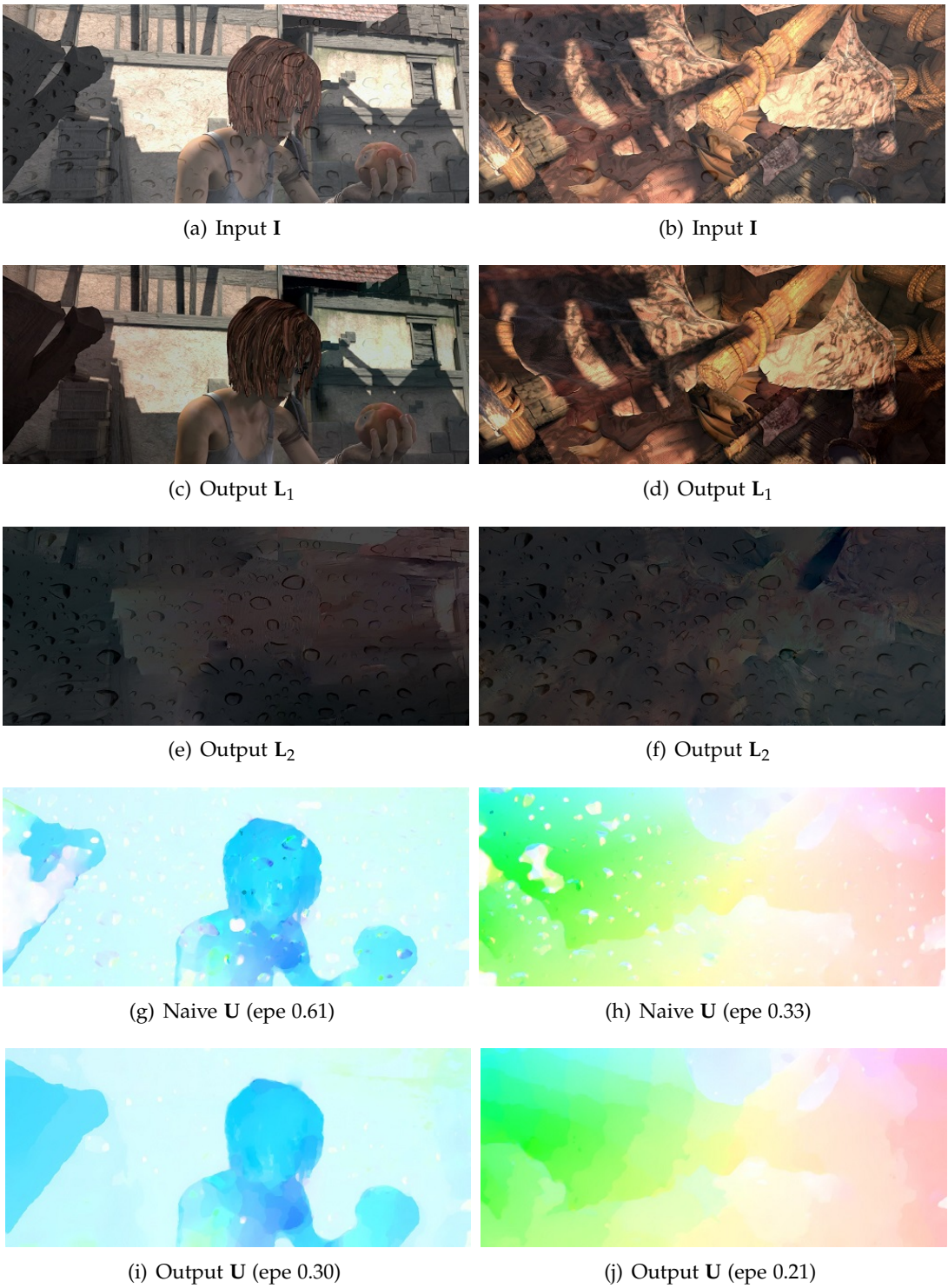
(d) Naive $\mathbf{U}$ (epe 1.01) (e) Output $\mathbf{U}$ (epe 0.29) (f) Oracle $\mathbf{U}$ (epe 0.16)

(g) Output $\mathbf{L}_1$ (h) Output $\mathbf{L}'_1$ (i) Output $\mathbf{L}_2$

(j) GT $\mathbf{L}_1$ (k) GT $\mathbf{L}'_1$ (l) GT $\mathbf{L}_2$

Figure 6.3: Performance evaluation of the proposed method on a single flow case, where a rain image is superimposed on the Dimetrodon image pair. The estimated flow (e) is significantly better than the initialization (f), a naive optical flow estimate without layer separation. The error evolution curve is shown in Figure 6.2. Oracle flow (l) is computed with clean background images (i.e., with ground-truth layer separations). (***Best viewed on screen***)

To further test the performance of our method, we synthesized another pair by superimposing the Lena image with the Grove image in the Middlebury dataset. The results are demonstrated in Figure 6.5. Again, we obtained a much better optical



(a) Input **I** (b) Input **I**

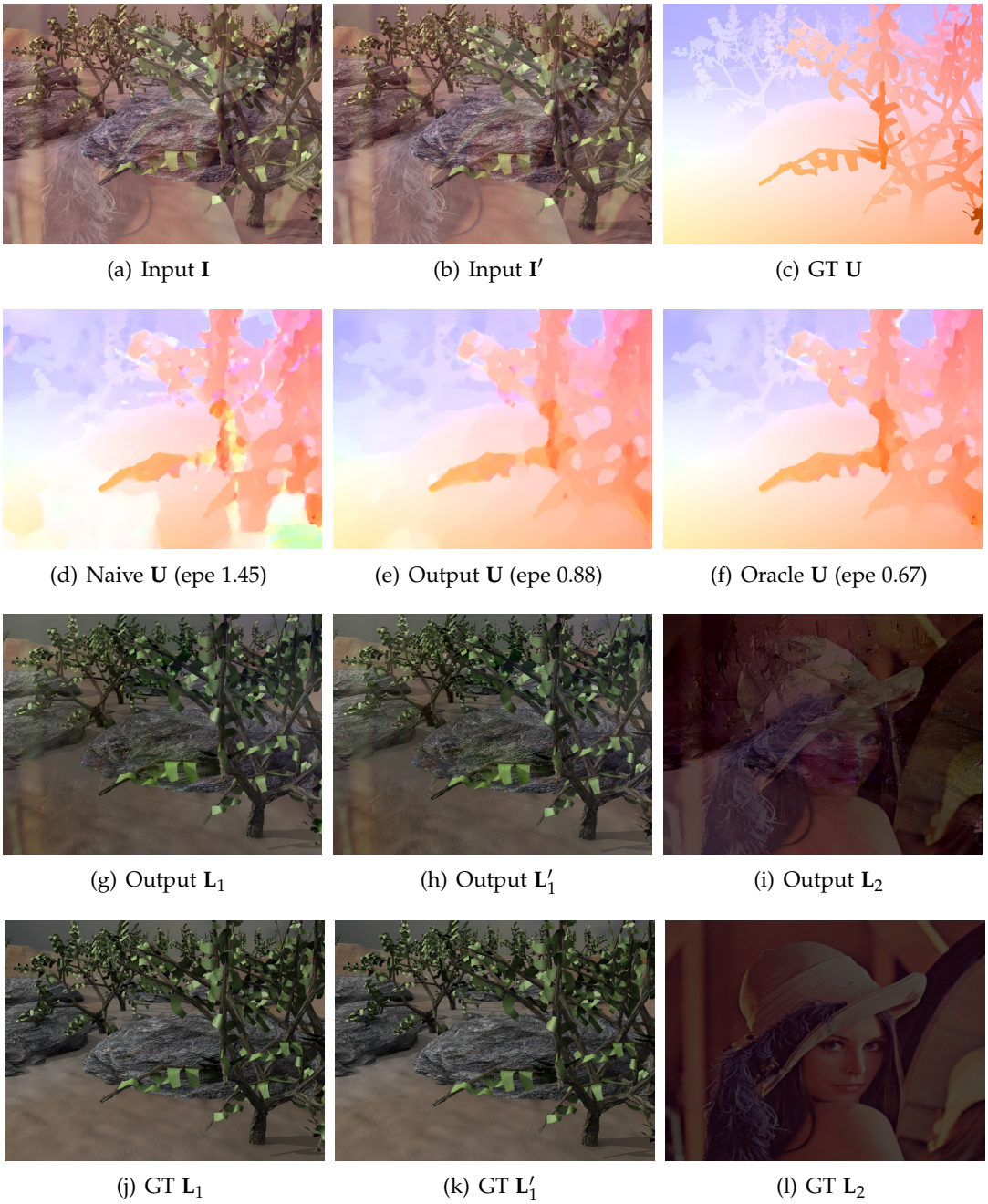
(c) Output $\mathbf{L}_1$ (d) Output $\mathbf{L}_1$

(e) Output $\mathbf{L}_2$ (f) Output $\mathbf{L}_2$

(g) Naive **U** (epe 0.61) (h) Naive **U** (epe 0.33)

(i) Output **U** (epe 0.30) (j) Output **U** (epe 0.21)

Figure 6.4: Typical results of our method on single-flow cases, where the rain drop image is superimposed on images from the Sintel dataset. For clarity, we only show here the first frame **I** and its layer separation result. (***Best viewed on screen***)



(a) Input $\mathbf{I}$ (b) Input $\mathbf{I}'$ (c) GT $\mathbf{U}$

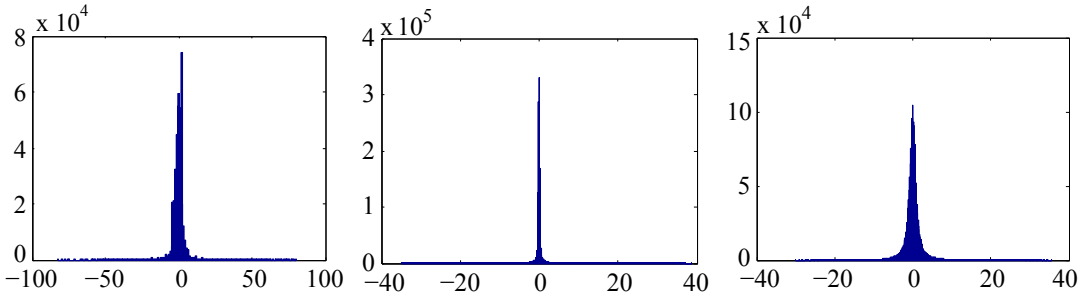
(d) Naive $\mathbf{U}$ (epe 1.45) (e) Output $\mathbf{U}$ (epe 0.88) (f) Oracle $\mathbf{U}$ (epe 0.67)

(g) Output $\mathbf{L}_1$ (h) Output $\mathbf{L}'_1$ (i) Output $\mathbf{L}_2$

(j) GT $\mathbf{L}_1$ (k) GT $\mathbf{L}'_1$ (l) GT $\mathbf{L}_2$

Figure 6.5: Performance evaluation of the proposed method on a single flow case, where the Lena image is superimposed on the Grove image pair. The estimated flow (e) is significantly better than the initialization (f), a naive optical flow estimate without layer separation. Oracle flow (l) is computed with ground-truth $\mathbf{L}_2$. (***Best viewed on screen***)

flow compared to the initial naive optical flow estimate. As for the layer separation results, the portrait of Lena can be hardly seen in the restored grove images.



(a) Dimetrodon (b) Rain drop (c) Lena

Figure 6.6: Gradient statistics of three used images.

In Figure 6.6 we show the image gradient statistics of the three foreground images used in the above experiments. The experimental results have shown that the proposed method works well on these images with the sparse gradient prior. Whenever available, other strong statistical priors can be incorporated into the optimization framework to further improve the performance.

The example in Figure 6.1 simulated a driving scenario, where a different raindrop image is superposed onto an image of the scene outside of the car window. It can be seen that the presence of raindrops can lead to poor flow estimation results. However, the proposed method produced a robust optical flow estimate and recovered a clean scene image.

6.5.2 Dynamic Foreground Cases

In this section, we test the proposed method in the dynamic foreground cases, where the task is that given two frames of input images $\mathbf{I}$ and $\mathbf{I}'$, recover four component layers $\mathbf{L}_1, \mathbf{L}'_1, \mathbf{L}_2, \mathbf{L}'_2$, and two dense motion fields $\mathbf{U}$, $\mathbf{V}$. In the problem of reflection removal, both the background scene and the reflection can be dynamic, which can give rise to such a situation.

We use two pairs of dynamic reflection scenes from [Li and Brown, 2013] to test the proposed method on the double-layer optical flow problem. In previous single-flow experiments, we initialize the method with foreground layers being all zero. However, this simple strategy did not work for the double-flow case. No reasonably good flow field could be obtained with this strategy for the background or reflection layer, especially for the reflection layer as its signal is weak. Indeed, the fact that the background layer is much more prominent has been taken advantage of by some layer separation methods Li and Brown [2013]; Guo et al. [2014] which align the input images with respect to the background layer. To obtain proper initialization, we first ran method of [Li and Brown, 2013] for initial layer separations[3], then computed initial optical flows on them.

[3]The method of [Li and Brown, 2013] takes multiple images as input, with one of them being the reference on which the reflection is to be removed. We apply this method on two images, and run it twice with each image as reference.

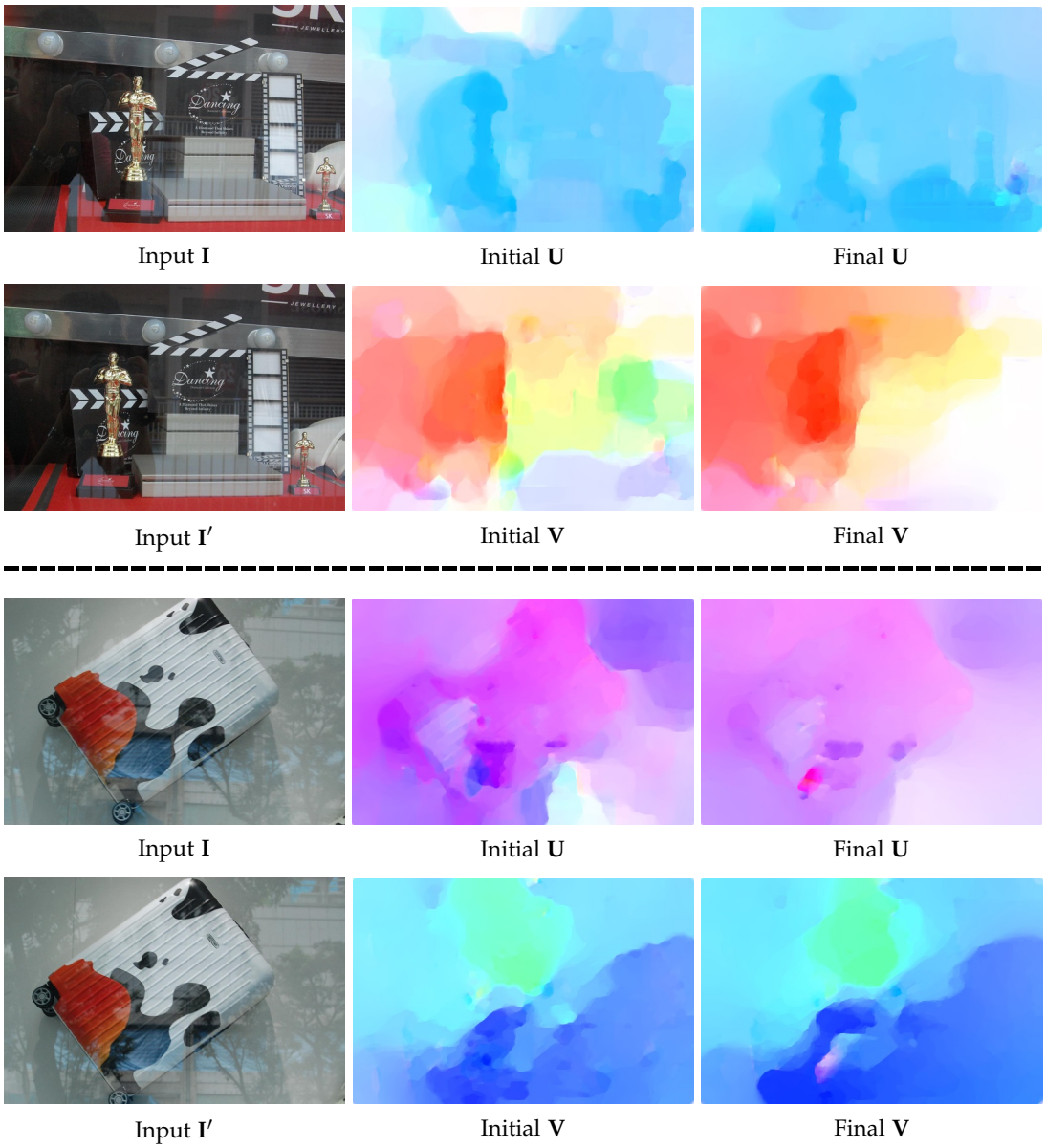


Figure 6.7: Double-layer optical flow estimation results on real reflection images. Visually inspected, the final optical flow fields are smoother and more consistent (see e.g., the results on the back wall in the first example, and results on the floor in the second example). The corresponding warping errors are presented in Table 6.2. (***Best viewed on screen***)

Table 6.2: Mean image warping errors (in gray levels) from the double-flow estimation results.

Image pair	Initial results	Our final results
#1	6.27	**2.55**
#2	3.86	**1.49**

The initial and final results are presented in Figure 6.7. Visually inspected, the final optical flow fields are smoother and more consistent (see e.g. the results on the back wall in the first example, and results on the floor in the second example). As no ground truth optical flow is available, we use image warping error to quantitatively evaluate the estimated flows. The warping error for a pixel $\mathbf{x}$ in $\mathbf{L}_1$ or $\mathbf{L}_2$ is $\|\mathbf{L}_1(\mathbf{x}+\mathbf{U}(\mathbf{x}))-\mathbf{L}_1'(\mathbf{x})\|_2$ or $\|\mathbf{L}_2(\mathbf{x}+\mathbf{V}(\mathbf{x}))-\mathbf{L}_2'(\mathbf{x})\|_2$, respectively. We compute the mean warping errors for all pixels on $\mathbf{L}_1$ and $\mathbf{L}_2$. As shown in Table 6.2, our method has significantly reduced the warping error upon the initializations. Figure 6.8 and 6.9 show the improvements of the reflection removal upon the initial estimates.

Discussion. The dynamic foreground case with double-layer flow estimation is generally much harder than the single-flow case. This is not only because the former has more unknown variables to be solved for, but also due to the difficulties in obtaining a good initialization. Nevertheless, our experiments show that the proposed method consistently improved the reasonable initializations given to it, for both the single-flow and double-flow cases.

Limitation. The proposed method is better suited for scenarios where the correlation between latent layers and their flow fields are relatively small. It will fail if both the two layers are textureless (as infinite numbers of possible motions exist satisfying the BCC constraints), or they undergo the same motion (thus the original BCC holds and only a single motion field can be extracted).

6.6 Conclusion

This chapter has defined the problem of robust optical flow estimation in the presence of possibly moving transparent or reflective layers. To our knowledge, the problem goes beyond the scope of conventional optical flow methods and was not properly investigated before.

We have presented a generalized double-layer brightness constancy condition as well as an optimization framework to solve this problem. The double-layer brightness constancy condition couples the flow fields and the brightness layers. Encouraging experimental results of optical flow estimation and layer separation on challenging data have been obtained, even though we are using simple priors for them.

The current framework is based on a generative model, which is applied uniformly to both the foreground and background layers. In future, we plan to leverage discriminative models to exploit the differences between the two layers for better layer separation. We also would like to explore some other optical flow priors. One possible strategy is to apply piecewise parametric motion model proposed in Chapter 5, which provides stronger constraints than general smoothness regularizers such as a TV, and is demonstrated to have advanced performances. Some other issues such as occlusion handling could also be considered.

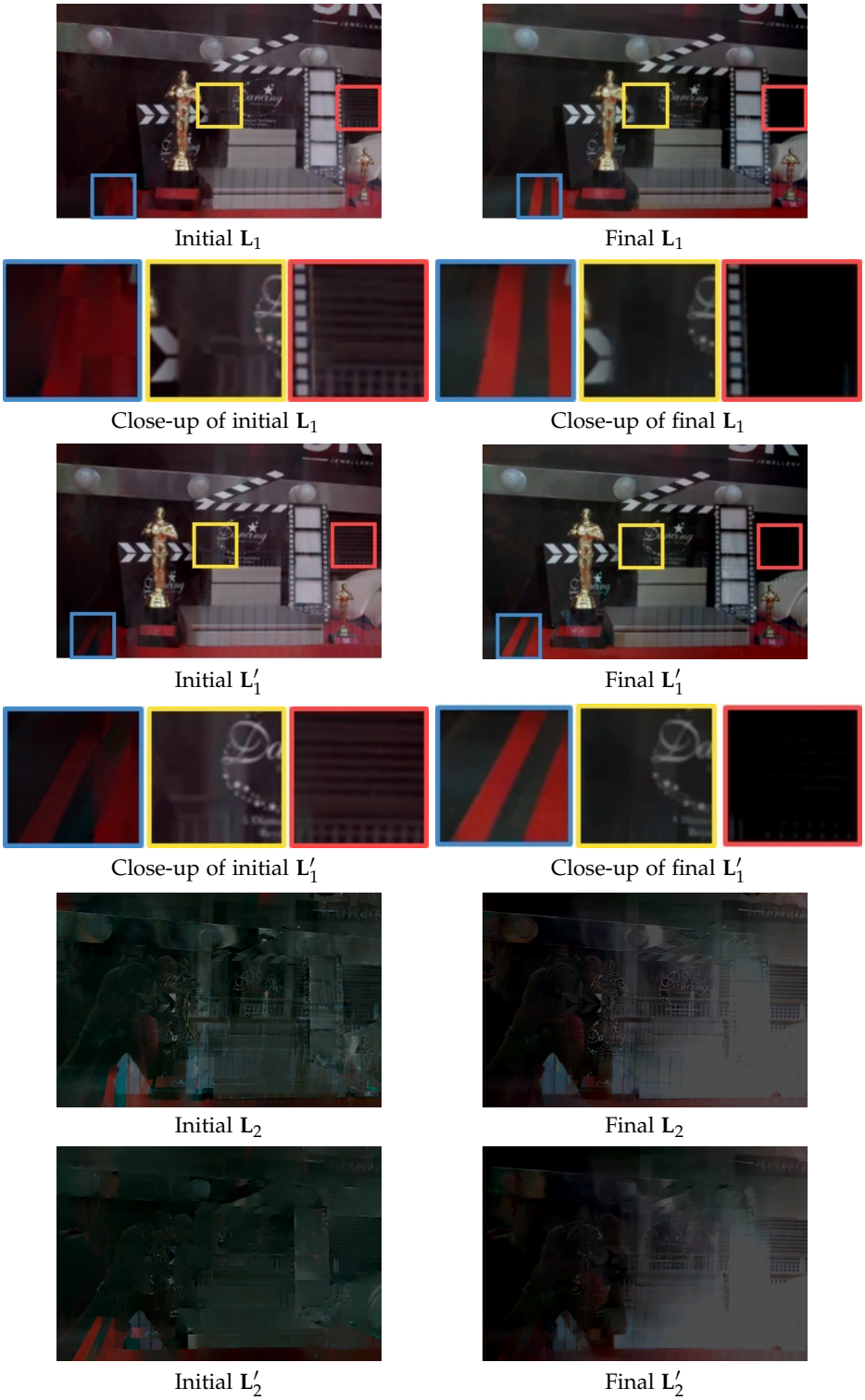


Figure 6.8: Layer separation results on real reflection images (the 1st pair). The initial layer separations are estimated by running method of [Li and Brown, 2013] on the two input images. The corresponding warping errors are presented in Table 6.2. The close-up images show the improvements of the reflection removal results upon the initial estimates. (***Best viewed on screen***)

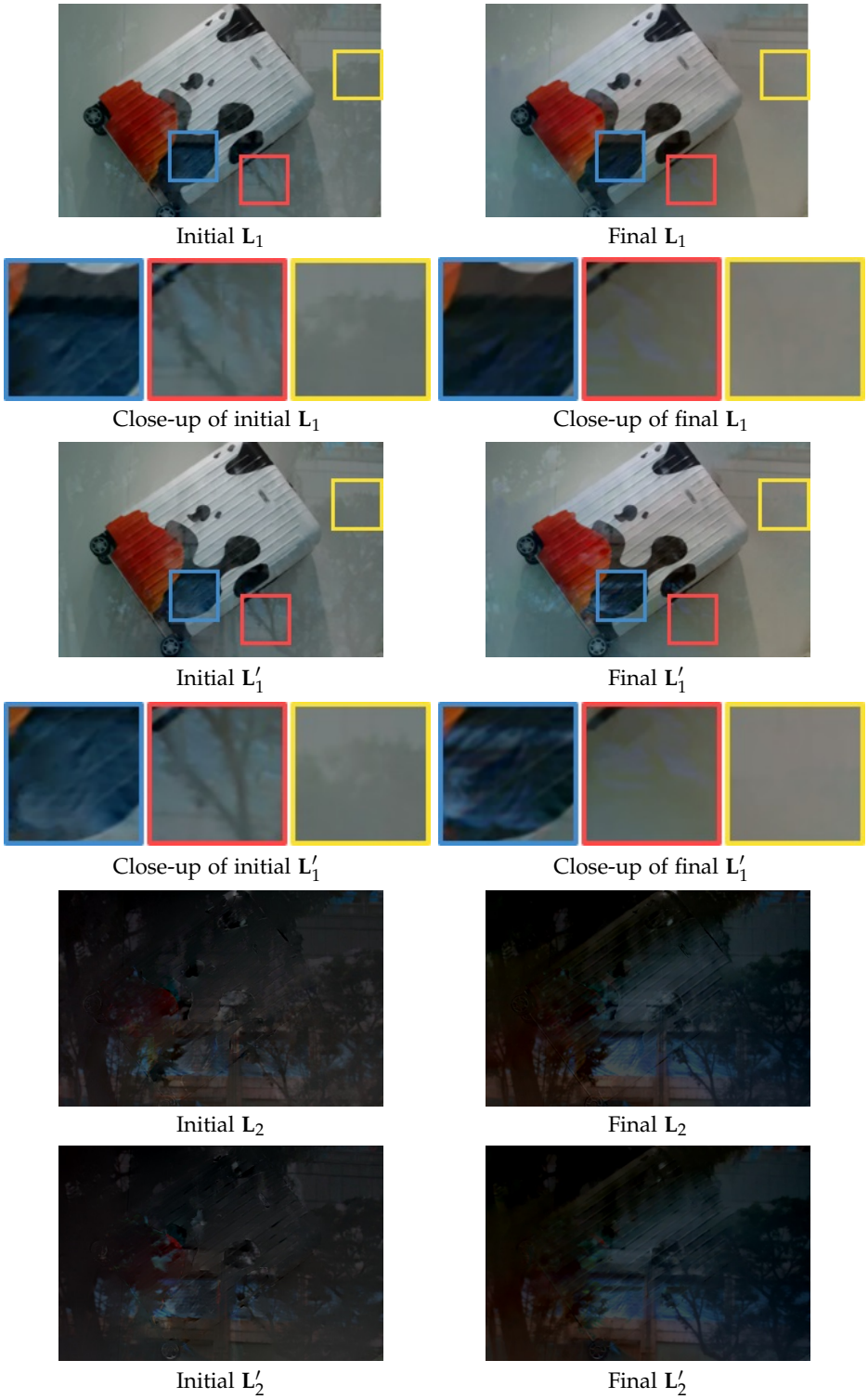


Figure 6.9: Layer separation results on real reflection images (the 2nd pair). The initial layer separations are estimated by running method of [Li and Brown, 2013] on the two input images. The corresponding warping errors are presented in Table 6.2. The close-up images show the improvements of the reflection removal results upon the initial estimates. (***Best viewed on screen***)

Chapter 7

Summary and Future Work

Motion estimation is one of the fundamental problems in computer vision which has broad application. The studies of camera and image motion have started since the emergence of the computer vision field. However, motion estimation remains an active topic nowadays with many challenging problems yet to be solved, as we have shown in the previous chapters.

7.1 Summary and Contributions

This dissertation has been devoted to analyzing the current challenges and push the limits of the state-of-the-art in various aspects, such as optimality, robustness, accuracy, and flexibility. A summary of the contributions is given below.

Optimality for 3D point cloud registration and 3D camera motion estimation (Chapter 2) and 2D color camera relative motion estimation (Chapter 3). We have proposed the first globally optimal algorithm for the ICP-style 3D point cloud registration problem and applied it to the motion estimation of 3D imaging devices. The idea is to analyze the structure of the $\mathrm{SE}(3)$ geometry and derive the error bounds for Branch-and-Bound (BnB) optimization. Similarly, we also have proposed a globally optimal inlier-set maximization algorithm for color camera relative motion estimation. We achieve this by analyzing the structure of the 5-D essential manifold, and presenting a new parameterization which enables efficient BnB search. The two BnB-based methods are actually highly related to each other with the similar insights in bound derivation.

Robustness for 2D color camera relative motion estimation (Chapter 3) and image motion estimation in the presence of transparency or reflection (Chapter 6). To deal with outliers/feature mismatches in 2D camera motion estimation, we have formulated an inlier-set maximization problem as in the popular RANSAC algorithm, but solved it optimally via BnB. Experiments have shown that our method always finds more inliers than RANSAC, and can work under high outlier ratio especially for the wide-FOV cases. To achieve robust image motion estimation under transparency or reflection, we have proposed an algorithm which performs both optical flow estimation and image layer separation. It exploits a generalized double-layer

brightness consistency constraint connecting these two tasks and utilizes the priors for both of them. In this way, not only the robustness is achieved as shown in the experiments, but also clean background images are restored which are appealing for other vision tasks.

Accuracy for classical image motion estimation (Chapter 5). We have proposed a highly-accurate optical flow estimation algorithm based on a piecewise parametric motion model. A key innovation is that we fit a flow field piecewise to a variety of parametric models where the domain of each piece (i.e., shape, position and size) and adaptively determine model parameters, while at the same time maintaining a global inter-piece flow continuity constraint. The proposed algorithm has archived top-tier performances on three public optical flow benchmarks (KITTI, MPI Sintel, and Middlebury).

Flexibility for 2D color camera and 3D camera relative motion estimation (Chapter 4). Existing works for the 2D color camera and 3D camera relative motion estimation often involve cumbersome human intervention and lack flexibility (e.g., for on-site estimation). In this dissertation, we have developed a single-shot method and provided a corresponds-free solution in order to minimize human intervention. We make use of known geometric constraints from the scene, and formulate relative pose estimation as a 2D-3D registration problem minimizing the geometric errors from scene constraints. The experiments have shown that the method is both flexible and accurate.

7.2 Future Work

Certainly, it is possible to further refine and improve the proposed methods, such as improving their efficiency via different strategies as well as exploiting other constraints and priors. Discussions about the possible amelioration for them have been provided in the conclusion part of each chapter.

There are also some other promising future works along different directions. For example, two research topics of our interest regarding to optical flow estimation will be briefly described as follows.

Semantic optical flow – *bridging low-level vision and high-level vision and improving both*. Up to now, optical flow estimation is rarely coupled with semantic information, and often used as a black-box for high-level vision problems such as object/action recognition. Recently, ideas have emerged for boosting the performance of optical flow estimation with the aid of image semantics [Sevilla-Lara et al., 2016; Bai et al., 2016]. For example, Sevilla-Lara et al. [2016] proposed to use different motion models for different semantic objects in the images. To this end, they first segment the image into different objects based on recent advances in image semantic segmentation. They then fit motion models to each connected semantic region and jointly refine the segmentation. Bai et al. [2016] propose to detect and segment out moving objects such as cars in the driving scene. In this way, in the remaining static background re-

gion, the image motion is purely induced by the camera ego-motion and optical flow estimation is reduced to a 1-D search. The motions of cars can be independently estimated. In fact, it seems quite natural to incorporate semantic segmentation in our piecewise parametric optical flow estimation framework (Chapter 5). We could endow the motion pieces with semantic labels, enforces appropriate constraints for pieces of one semantic object, and jointly solve for motion estimation and semantic segmentation. In this way, low-level motion estimation and high-level can be connected. This is not only theoretically meaningful, but also of practical value (e.g., for autonomous driving).

Scene flow estimation – *dense 3D motion analysis of the scene*. Optical flow estimation only provides 2D motion on the image plane. With a stereo camera rig or a range/depth sensor, dense 3D motions and structures of the scene can be estimated. Recently, there are rising interests in scene flow estimation, either using a stereo camera [Wedel et al., 2011; Vogel et al., 2014, 2015; Menze and Geiger, 2015], or a RGB-D camera rig (which is also called RGB-D flow estimation) [Herbst et al., 2013; Quiroga et al., 2014; Sun et al., 2015a]. With a stereo camera rig, the input includes two stereo image pairs (i.e., four images in total), and the output can be two flow fields and two depth (disparity) maps. In this way, both the 3D structure and the 3D motion are recovered. RGB-D flow estimation is to compute dense correspondences between two RGB-D image pairs. In each pair, the color and depth images are registered. So the problem is more similar to the conventional optical flow estimation, except in this case we have the extra depth information and the flow vectors are essentially in 3D. Scene flow estimation provides dense 3D motion analysis, which is more appealing for scene understanding. It is also a promising technique for autonomous driving and augmented reality.

APPENDIX: Neural Aggregation Network for Video Face Recognition*

Face video bear more information of the subjects than images. Video face recognition has caught more and more attention from the community in recent years [Wolf et al., 2011; Li et al., 2013; Wolf and Levy, 2013; Cui et al., 2013; Li et al., 2014; Liu et al., 2014; Parkhi et al., 2014; Hu et al., 2014a; Taigman et al., 2014; Schroff et al., 2015].

A naive approach to build a feature representation of the video face would be using a set of frame-level face features [Taigman et al., 2014; Schroff et al., 2015]. However, such a set-based representation would incur $O(n)$ space complexity per video face example for storage, and $O(n^2)$ complexity for face comparison (verification) in the recognition phase. Therefore, it is more desirable to come with a compact, fixed-size visual representation for video faces, irrespective of the varied length of the videos. A straightforward solution might be conducting pooling to aggregate the frame-level features, such as the commonly adopted *average* and *max* pooling.

However, we argue that a good pooling strategy should adaptively weigh the frame-level features across all frames. To this end, we look for an adaptive weighting scheme to linearly combine all frame features from a video together to form a compact and discriminative face representation. We designed a neural network, named the *neural aggregation network (NAN)* to adaptively calculate the weights at runtime.

The proposed NAN is designed to inherit the main advantages of pooling techniques, including the ability to handle arbitrary input size and producing ordering invariant representation. Its key component is inspired by the Neural Turing Machine [Graves et al., 2014] and the Orderless Set Network [Vinyals et al., 2016], both of which applied an attention mechanism to organize the input through a memory.

A.1 Neural Aggregation Network

As shown in Figure A.1, the NAN is composed of two modules that could be trained end-to-end or one by one separately: one *feature embedding module* with a deep CNN model and an *aggregation module* that adaptively fuses the frame features.

*This work was done when I was interning at Microsoft Research. It is included and briefly introduced in this appendix chapter only for the sake of completeness of all works done during my PhD study.

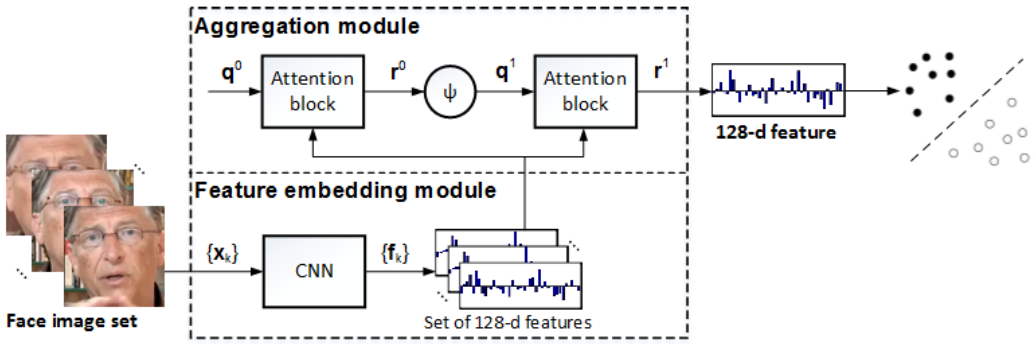


Figure A.1: The face recognition framework of our method.

A.1.1 Feature embedding module

Let $X = \{\mathbf{x}_k\}$, $k = 1, ..., t$ be a face video of t frames. The feature embedding module is a deep Convolution Neural Network (CNN) which embeds each $\mathbf{x}_k$ to a feature representation $\mathbf{f}_k$. In this work, we employ GoogLeNet [Szegedy et al., 2015] with Batch Normalization (BN) [Ioffe and Szegedy, 2015] as our CNN. It outputs 128-d frame features $\{\mathbf{f}_k\}$, which are then fed into the aggregation module. In the remaining text, we simply refer to the GoogLeNet-BN network as CNN.

A.1.2 Aggregation module

Given the frame features $\{\mathbf{f}_k\}$, the goal of the aggregation module is to generate a set of linear weights $\{a_k\}_{k=1}^t$, so that the aggregated feature representation becomes

$$\mathbf{r} = \sum_k a_k \mathbf{f}_k. \tag{A.1}$$

If $a_k \equiv \frac{1}{t}$, (A.1) will degrade to naive averaging which is usually non-optimal. We instead try to let the data itself help generate better weights. To this end, we employ a content based attention mechanism [Graves et al., 2014] in our new network structure.

The crux of our aggregation module is two attention blocks, as shown in Figure A.1. Each attention block takes $\{\mathbf{f}_k\}$ as input, filters them with a kernel $\mathbf{q}$ via dot product, yielding a set of corresponding significances $\{e_k\}$ which are then passed to a softmax operator to generate normalized weights $\{a_k\}$ with $\sum_k a_k = 1$:

$$e_k = \mathbf{q}^T \mathbf{f}_k \tag{A.2}$$

$$a_k = \frac{\exp(e_k)}{\sum_j \exp(e_j)}. \tag{A.3}$$

The attention block is modulated solely by a filter kernel $\mathbf{q}$. One key advantage of the attention block is that its output is *invariant* to the input order of $\mathbf{f}_k$: it can be seen from (A.2), (A.3) and (A.1) that permuting $\mathbf{f}_k$ and $\mathbf{f}_{k'}$ has no effects on $\mathbf{r}$. Another appealing property is that the number of frames (i.e., t) does not affect the size of

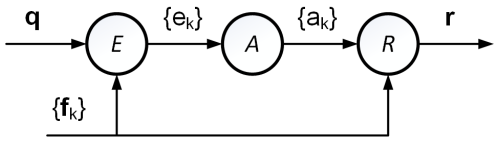


Figure A.2: Illustration of an attention block.

output $\mathbf{r}$ which is of the same dimension with a single $\mathbf{f}_k$.

We employed two attention blocks with filters $\mathbf{q}^0$ and $\mathbf{q}^1$ respectively. $\mathbf{q}^0$ serves as a universal prior measuring the quality of face features. In contrast, $\mathbf{q}^1$ gives rise to an aggregation that is content aware and discriminative for recognition. It is dynamically computed from the first's output $\mathbf{r}^0$, through a transfer layer:

$$\mathbf{q}^1 = \tanh(\mathbf{W}\mathbf{r}^0 + \mathbf{b}) \tag{A.4}$$

where $\mathbf{W}$ and $\mathbf{b}$ are the weight matrix and bias vector respectively, and $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ imposes the hyperbolic tangent nonlinearity. Therefore, the parameters of the aggregation module consists of $\mathbf{q}^0$ and $(\mathbf{W}, \mathbf{b})$, all of which can be trained by supervised learning via standard gradient descent.

A.1.3 Network training

The NAN can be trained for both face verification and identification tasks. For *identification*, we add on top a fully-connected layer followed by softmax, and minimize the classification loss. For *verification*, we add on top a normalization layer to generate unit feature vectors, then build a *siamese* structure [Chopra et al., 2005] with two NANs and minimize the contrastive loss [Hadsell et al., 2006].

In this work, we train the two modules of NAN one by one: we first train the CNN on single images with the identification task, then train the aggregation module for identification and verification with the features extracted by CNN .

A.2 Experiments

This section evaluates the performance of the NAN for video face recognition tasks. We will report results on three video face recognition datasets: the YouTube Face dataset [Wolf et al., 2011], the IARPA Janus Benchmark A (IJB-A) [Klare et al., 2015], and the Celebrity-1000 dataset [Liu et al., 2014].

Training details. To train the CNN, we use around 3M face images of 50K identities crawled from the internet to perform identification. The input image size is 224x224, and the CNN is fixed after training. The aggregation module is trained on the video face datasets we use with standard backpropagation and gradient descent.

Baselines. The performance of our method is evaluated against a few baselines:

CNN+Mean L_2 measures the similarity of two video faces via averaging the feature distances of all frame pairs. It necessitates storing all image features of a video or subject, and has $O(n^2)$ complexity for similarity computation.

Table A.1: Verification accuracy comparison of state-of-the-art methods, our baselines and NAN network on the YouTube Face dataset.

Method	Accuracy (%)	AUC
LM3L [Hu et al., 2014b]	81.3 ± 1.2	89.3
DDML (combined) [Hu et al., 2014a]	82.3 ± 1.5	90.1
EigenPEP [Li et al., 2014]	84.8 ± 1.4	92.6
DeepFace-single [Taigman et al., 2014]	91.4 ± 1.1	96.3
DeepID2+ [Sun et al., 2015b]	93.2 ± 0.2	–
FaceNet [Schroff et al., 2015]	95.12 ± 0.39	–
CNN+Min. L_2	94.46 ± 0.10	98.3
CNN+Mean L_2	95.30 ± 0.08	98.6
CNN+MaxPool	86.60 ± 0.13	94.1
CNN+AvePool	95.10 ± 0.08	98.5
NAN	**95.52 ± 0.06**	**98.7**

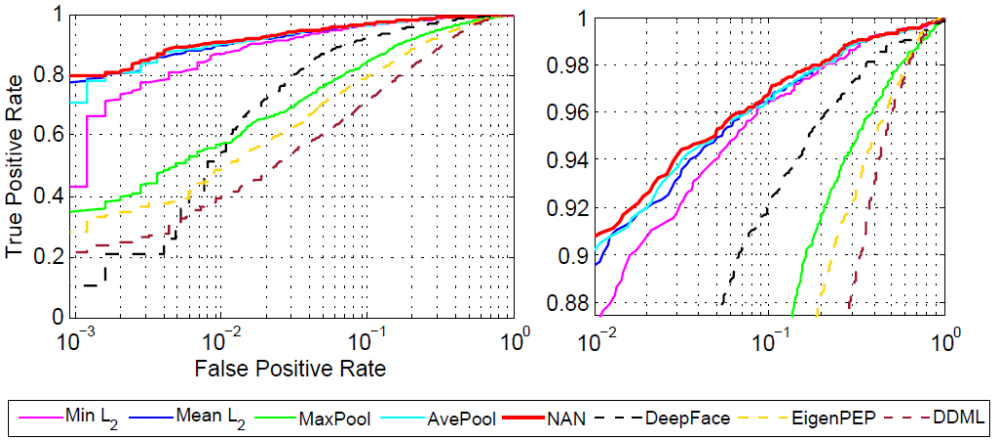


Figure A.3: Average ROC curves of different methods on the YouTube Face dataset.

CNN+Min L_2 is similar to the above but uses the smallest pairwise distance.
CNN+AvePool is average-pooling along each feature dimension for aggregation.
CNN+MaxPool is max-pooling along each feature dimension for aggregation.

A.2.1 Results on YouTube Face dataset

The YouTube Face dataset is designed for unconstrained *face verification* in videos. It contains 3,425 videos of 1,595 different people, with an average of 2.15 videos per subject. The video lengths vary from 48 to 6,070 frames. Ten folds of 500 video pairs are available for cross-validation.

The results of our NAN, its baselines and other methods are presented in Table A.1, with their ROC curves in Figure A.3. The baselines, except CNN+MaxPool, achieve similar accuracies to the state-of-the-art method FaceNet [Schroff et al., 2015], which has an accuracy of 95.12%±0.39. Note that FaceNet is also based on a GoogLeNet style network, and the average similarity of all pairs of 100 frames in each video (*i.e.*,

Table A.2: Verification accuracy comparison of different methods on the IJB-A dataset.

Method	TAR @ FAR=0.001	TAR @ FAR=0.01	TAR @ FAR=0.1
GOTS	0.198 ± 0.008	0.406 ± 0.014	0.627 ± 0.012
OpenBR [Klontz et al., 2013]	0.104 ± 0.014	0.236 ± 0.009	0.433 ± 0.006
LSFS [Wang et al., 2015]	0.514 ± 0.060	0.733 ± 0.034	0.895 ± 0.013
DCNN_{manual}+metric [Chen et al., 2015]	–	0.787 ± 0.043	0.947 ± 0.011
CNN+Mean L_2	0.453 ± 0.015	0.828 ± 0.023	0.957 ± 0.007
CNN+AvePool	0.559 ± 0.014	0.856 ± 0.017	0.953 ± 0.007
NAN	**0.785 ± 0.028**	**0.897 ± 0.010**	**0.959 ± 0.005**

10K pairs) was used. Our NAN outperforms all the baselines and other methods. It achieves **95.52%±0.06** accuracy, reducing the error of FaceNet by 8.2%.

The face variations in the videos of this dataset are relatively small. We then test on more challenging datasets which better shows the advantage of the NAN.

A.2.2 Results on IJB-A dataset

The IJB-A dataset contains 5,397 images and 2,042 videos for 500 subjects, with 11.4 images and 4.2 videos per subject on average. This challenging dataset features full pose variation and wide variations in imaging conditions. Each instance is called a 'template', which comprises a mixture of still images and sampled video frames. Each template contains 1 to 190 images, with 10 images per template on average.

We tested the NAN on the 'compare' (1:1 matching) protocol for *face verification* on IJB-A with 10 training and testing splits. The results are presented in Table A.2. The NAN outperforms its baselines, especially on the low FAR cases. The TARs of NAN at FARs of 0.001, 0.01 and 0.1 are 0.785, 0.897 and 0.959 respectively. The errors are reduced by about 56%, 51% and 22% respectively compared to [Chen et al., 2015], which used averaged image features. The proposed NAN has learned a discriminative aggregation mechanism, which results in better verification accuracy compared to the baseline aggregations and set-distance measurements.

A.2.3 Results on Celebrity-1000 dataset

The Celebrity-1000 dataset is designed to study the unconstrained video-based *face identification* problem. This dataset contains 159,726 video sequences of 1,000 human subjects, with 2.4M frames in total (15 frames per sequence on average). Two types of protocols – open-set and close-set – exist on this dataset.

Close-set tests. For the close-set tests, the subject with the maximum score from the FC layer is the identification result for the NAN, and a linear classifier is trained for 'CNN+AvePool'. We call this approach 'VideoAggr'. We can also build a single representation for each subject by aggregating all its images in all the gallery video sequences. In this way, the linear classifier can be bypassed and identification can be achieved simply by comparing the L_2 feature distances. This approach is called

Table A.3: Identification performance comparison on the Celebrity-1000 dataset with the *close-set* protocol. The Rank-1 accuracies (%) are presented.

Method	100 subjects	200 subjects	500 subjects	1000 subjects
MTJSR [Liu et al., 2014]	50.60	40.80	35.46	30.04
Eigen-PEP [Li et al., 2014]	50.60	45.02	39.97	31.94
CNN+AvePool - VideoAggr	86.06	82.38	80.48	74.26
CNN+AvePool - SubjectAggr	84.46	78.93	77.68	73.41
NAN - VideoAggr	88.04	82.95	**82.27**	76.24
NAN - SubjectAggr	**90.44**	**83.33**	**82.27**	**77.17**

Table A.4: Identification performance comparison on the Celebrity-1000 dataset with the *open-set* protocol. The Rank-1 accuracies (%) are presented.

Method	100 subjects	200 subjects	400 subjects	800 subjects
MTJSR [Liu et al., 2014]	46.12	39.84	37.51	33.50
Eigen-PEP [Li et al., 2014]	51.55	46.15	42.33	35.90
CNN+Mean L_2	84.88	79.88	76.76	70.67
CNN+AvePool - SubjectAggr	84.11	79.09	78.40	75.12
NAN - SubjectAggr	**88.76**	**85.21**	**82.74**	**79.87**

'SubjectAggr'. As shown in Table A.3, all the baseline methods and our NAN outperformed previous methods by large margins, and the NAN outperformed the baseline methods on all tasks. It is interesting to see that, 'SubjectAggr' leads to a clear performance drop by CNN+AvePool. This indicates that the hand-crafted aggregation gets even worse when applied on the subject level. However, our NAN can benefit from 'SubjectAggr', yielding a result better than or on par with the 'VideoAggr' approach.

Open-set tests. For the open-set protocol, after training we took the 'SubjectAggr' approach described before to build a highly-compact face representation for each gallery subject. Identification was performed similarly by comparing the L_2 feature distances. The results in Table A.4 shows that our NAN significantly reduced the error of the baseline CNN+AvePoolfor all settings. This again suggests that in the presence of large face variances, the widely used strategies such as average pooling aggregation and the pairwise distance computation are far from optimal. In such cases, the learned NAN model is powerful and can yield much superior results.

A.3 Conclusion

We have presented a Neural Aggregation Network, which is based on CNN and an attention mechanism, for video face representation and recognition. It fuses all input frames with a set of content adaptive weights, resulting in a compact (128-d) representation that is invariant to the frame order. The structure of the aggregation module is simple with small computation and memory footprints, but can generate a comprehensive face representation after trained through supervised learning.

Bibliography

Agarwal, S.; Snavely, N.; Simon, I.; Seitz, S. M.; and Szeliski, R., 2009. Building Rome in a day. In *International Conference on Computer Vision*, 72–79. (cited on page 9)

Aiger, D.; Mitra, N. J.; and Cohen-Or, D., 2008. 4-points congruent sets for robust pairwise surface registration. *ACM Transactions on Graphics (TOG)*, 27, 3 (2008). (cited on pages 5 and 22)

Alismail, H.; Baker, L.; and Browning, B., 2012. Automatic calibration of a range sensor and camera system. In *International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPTV)*, 286–292. (cited on pages 11 and 67)

Arun, K.; Huang, T.; and Blostein, S., 1987. Least-squares fitting of two 3-D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, , 5 (1987), 698–700. (cited on page 23)

Auvray, V.; Bouthemy, P.; and Liénard, J., 2009. Joint motion estimation and layer segmentation in transparent image sequences: application to noise reduction in X-ray image sequences. *EURASIP Journal on Advances in Signal Processing*, (2009), 19:1–19:21. (cited on page 105)

Bai, M.; Luo, W.; Kundu, K.; and Urtasun, R., 2016. Deep semantic matching for optical flow. *arXiv preprint arXiv:1604.01827*, (2016). (cited on page 126)

Baker, S.; Scharstein, D.; Lewis, J.; Roth, S.; Black, M.; and Szeliski, R., 2011a. A database and evaluation methodology for optical flow. *International Journal of Computer Vision (IJCV)*, 92, 1 (2011), 1–31. (cited on page 114)

Baker, S.; Scharstein, D.; Lewis, J.; Roth, S.; Black, M. J.; and Szeliski, R., 2011b. A database and evaluation methodology for optical flow. *International Journal of Computer Vision (IJCV)*, 92, 1 (2011), 1–31. (cited on pages 12, 14, 15, and 94)

Bao, L.; Yang, Q.; and Jin, H., 2014. Fast edge-preserving patchmatch for large displacement optical flow. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3534–3541. (cited on pages 14, 96, and 101)

Barnes, C.; Shechtman, E.; Finkelstein, A.; and Goldman, D., 2009. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (TOG)*, 28, 3 (2009), 24. (cited on pages 14, 93, and 94)

Barnes, C.; Shechtman, E.; Goldman, D. B.; and Finkelstein, A., 2010. The generalized patchmatch correspondence algorithm. In *European Conference on Computer Vision (ECCV)*, 29–43. (cited on page 14)

Bay, H.; Tuytelaars, T.; and Van Gool, L., 2006. Surf: Speeded up robust features. In *European Conference on Computer Vision (ECCV)*, 404–417. (cited on page 8)

Bazin, J.-C.; Seo, Y.; and Pollefeys, M., 2012. Globally optimal consensus set maximization through rotation search. In *Asian Conference on Computer Vision (ACCV)*, 539–551. (cited on pages 5, 21, 22, and 48)

Belongie, S.; Malik, J.; and Puzicha, J., 2002. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24, 4 (2002), 509–522. (cited on pages 5, 19, and 21)

Bergen, J. R.; Anandan, P.; Hanna, K. J.; and Hingorani, R., 1992a. Hierarchical model-based motion estimation. In *European Conference on Computer Vision (ECCV)*, 237–252. (cited on page 14)

Bergen, J. R.; Burt, P. J.; Hingorani, R.; and Peleg, S., 1992b. A three-frame algorithm for estimating two-component image motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, , 9 (1992), 886–896. (cited on pages 105 and 109)

Besl, P. and McKay, N., 1992. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 14, 2 (1992), 239–256. (cited on pages 3, 19, 23, 32, and 76)

Besse, F.; Rother, C.; Fitzgibbon, A.; and Kautz, J., 2014. PMBP: Patchmatch belief propagation for correspondence field estimation. *International Journal of Computer Vision (IJCV)*, 110, 1 (2014), 2–13. (cited on page 14)

Bhat, P.; Zheng, K. C.; Snavely, N.; Agarwala, A.; Agrawala, M.; Cohen, M. F.; and Curless, B., 2006. Piecewise image registration in the presence of multiple large motions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2491–2497. (cited on page 87)

Biber, P. and Strasser, W., 2003. The normal distributions transform: A new approach to laser scan matching. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, 2743–2748. (cited on page 21)

Birchfield, S. and Tomasi, C., 1999. Multiway cut for stereo and motion with slanted surfaces. In *International Conference on Computer Vision (ICCV)*, vol. 1, 489–495. (cited on pages 86 and 87)

Black, M. J. and Anandan, P., 1996. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding (CVIU)*, 63, 1 (1996), 75–104. (cited on pages 15, 85, and 105)

Black, M. J. and Jepson, A. D., 1996. Estimating optical flow in segmented images using variable-order parametric models with local deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 18, 10 (1996), 972–986. (cited on pages 85 and 87)

Blais, G. and Levine, M. D., 1995. Registering multiview range data to create 3D computer objects. *IEEE Transactions on on Pattern Analysis and Machine Intelligence (TPAMI)*, 17, 8 (1995), 820–824. (cited on pages 4, 19, and 21)

Bleyer, M.; Rhemann, C.; and Rother, C., 2011. Patchmatch stereo-stereo matching with slanted support windows. In *British Machine Vision Conference (BMVC)*, vol. 11, 1–11. (cited on pages 14, 89, and 93)

Bouaziz, S.; Tagliasacchi, A.; and Pauly, M., 2013. Sparse iterative closest point. In *Eurographics Symposium on on Geometry Processing*, vol. 32. (cited on pages 4, 33, and 44)

Boyd, S. P. and Vandenberghe, L., 2004. *Convex Optimization*. Cambridge University Press. (cited on page 23)

Boykov, Y.; Veksler, O.; and Zabih, R., 2001. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 23, 11 (2001), 1222–1239. (cited on page 92)

Braux-Zin, J.; Dupont, R.; and Bartoli, A., 2013. A general dense image matching framework combining direct and feature-based costs. In *International Conference on Computer Vision (ICCV)*, 185–192. (cited on pages 15, 89, and 96)

Bredies, K.; Kunisch, K.; and Pock, T., 2010. Total generalized variation. *SIAM Journal on Imaging Sciences*, 3, 3 (2010), 492–526. (cited on pages 15 and 109)

Breuel, T. M., 2003. Implementation techniques for geometric branch-and-bound matching methods. *Computer Vision and Image Understanding (CVIU)*, 90, 3 (2003), 258–294. (cited on pages 5, 22, and 24)

Brox, T.; Bruhn, A.; Papenberg, N.; and Weickert, J., 2004. High accuracy optical flow estimation based on a theory for warping. In *European Conference on Computer Vision (ECCV)*, 25–36. (cited on pages 12, 14, 15, 85, 89, and 108)

Brox, T. and Malik, J., 2011. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33, 3 (2011), 500–513. (cited on pages 86 and 96)

Bruhn, A.; Weickert, J.; and Schnörr, C., 2005. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International Journal of Computer Vision (IJCV)*, 61, 3 (2005), 211–231. (cited on page 14)

Bustos, A. P.; Chin, T.-J.; and Suter, D., 2014. Fast rotation search with stereographic projections for 3D registration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 5 and 22)

Butler, D. J.; Wulff, J.; Stanley, G. B.; and Black, M. J., 2012. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision (ECCV)*, 611–625. (cited on pages 12, 14, 94, and 114)

Campbell, D. and Petersson, L., 2015. An adaptive data representation for robust point-set registration and merging. In *International Conference on Computer Vision (ICCV)*, 4292âĂŞ–4300. (cited on pages 4 and 21)

Chambolle, A., 2004. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision (JMIV)*, 20, 1-2 (2004), 89–97. (cited on page 15)

Chambolle, A. and Pock, T., 2011. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision (JMIV)*, 40, 1 (2011), 120–145. (cited on pages 15 and 113)

Champleboux, G.; Lavallee, S.; Szeliski, R.; and Brunie, L., 1992. From accurate range imaging sensor calibration to accurate model-based 3D object localization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 83–89. (cited on pages 4, 24, and 33)

Chartrand, R. and Yin, W., 2008. Iteratively reweighted algorithms for compressive sensing. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 3869–3872. (cited on page 111)

Chen, J.-C.; Ranjan, R.; Kumar, A.; Chen, C.-H.; Patel, V.; and Chellappa, R., 2015. An end-to-end system for unconstrained face verification with deep convolutional neural networks. In *IEEE International Conference on Computer Vision Workshops (ICCVW)*, 118–126. (cited on page 133)

Chen, Y.; Chang, T.; Zhou, C.; and Fang, T., 2009. Gradient domain layer separation under independent motion. In *International Conference on Computer Vision (ICCV)*, 694–701. (cited on page 106)

Chen, Y. and Medioni, G., 1991. Object modeling by registration of multiple range images. In *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 3, 2724–2729. (cited on pages 3 and 19)

Chen, Z.; Jin, H.; Lin, Z.; Cohen, S.; and Wu, Y., 2013. Large displacement optical flow from nearest neighbor fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2443–2450. (cited on pages xiv, 87, 93, 99, and 100)

Chetverikov, D.; Stepanov, D.; and Krsek, P., 2005. Robust euclidean alignment of 3D point sets: the trimmed iterative closest point algorithm. *Image and Vision Computing (IVC)*, 23, 3 (2005), 299–309. (cited on pages 4, 24, 32, 33, and 46)

Chopra, S.; Hadsell, R.; and LeCun, Y., 2005. Learning a similarity metric discriminatively, with application to face verification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 539–546. (cited on page 131)

Chum, O. and Matas, J., 2005. Matching with prosac-progressive sample consensus. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 220–226. (cited on page 8)

Chum, O. and Matas, J., 2008. Optimal randomized RANSAC. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30, 8 (2008), 1472–1482. (cited on page 8)

Chum, O.; Matas, J.; and Kittler, J., 2003. Locally optimized RANSAC. In *Annual Symposium of the German Association for Pattern Recognition (DAGM)*, 236–243. (cited on pages 8 and 61)

Crandall, D. J.; Owens, A.; Snavely, N.; and Huttenlocher, D. P., 2013. SfM with MRFs: Discrete-continuous optimization for large-scale structure from motion. *IEEE transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35, 12 (2013), 2841–2853. (cited on page 9)

Cremers, D. and Soatto, S., 2005. Motion competition: A variational approach to piecewise parametric motion segmentation. *International Journal of Computer Vision (IJCV)*, 62, 3 (2005), 249–265. (cited on pages 86 and 87)

Cui, Z.; Li, W.; Xu, D.; Shan, S.; and Chen, X., 2013. Fusing robust face region descriptors via multiple metric learning for face recognition in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3554–3561. (cited on page 129)

Dai, Y.; Li, H.; and He, M., 2010. Element-wise factorization for n-view projective reconstruction. In *European Conference on Computer Vision (ECCV)*, 396–409. (cited on page 10)

Dai, Y.; Trumpf, J.; Li, H.; Barnes, N.; and Hartley, R., 2009. Rotation averaging with application to camera-rig calibration. In *Asian Conference on Computer Vision (ACCV)*, 335–346. (cited on page 67)

Daniilidisl, K. and Spetsakisz, M. E., 1997. Understanding noise sensitivity in structure from motion. *Visual navigation: from biological systems to unmanned ground vehicles*, (1997). (cited on page 63)

Darrell, T. and Simonecelli, E., 1993. 'nulling' filters and the separation of transparent motions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 738–739. (cited on page 105)

Dellaert, F.; Seitz, S. M.; Thorpe, C. E.; and Thrun, S., 2000. Structure from motion without correspondence. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 557–564. (cited on page 49)

Delong, A.; Osokin, A.; Isack, H. N.; and Boykov, Y., 2012. Fast approximate energy minimization with label costs. *International Journal on Computer Vision (IJCV)*, 96, 1 (2012), 1–27. (cited on pages 88 and 92)

Demetz, O.; Stoll, M.; Volz, S.; Weickert, J.; and Bruhn, A., 2014. Learning brightness transfer functions for the joint recovery of illumination changes and optical flow. In *European Conference on Computer Vision (ECCV)*, 455–471. (cited on page 96)

Dreisigmeyer, D. W., 2006. Direct search algorithms over riemannian manifolds. (cited on page 72)

Efros, A. A. and Freeman, W. T., 2001. Image quilting for texture synthesis and transfer. In *ACM SIGGRAPH*. (cited on page 88)

Enqvist, O.; Jiang, F.; and Kahl, F., 2011. A brute-force algorithm for reconstructing a scene from two projections. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2961–2968. (cited on pages 8, 48, 54, 55, and 63)

Enqvist, O.; Josephson, K.; and Kahl, F., 2009. Optimal correspondences from pairwise constraints. In *International Conference on Computer Vision (ICCV)*, 1295–1302. (cited on pages 5 and 22)

Enqvist, O. and Kahl, F., 2008. Robust optimal pose estimation. In *European Conference on Computer Vision (ECCV)*, 141–153. (cited on page 22)

Enqvist, O. and Kahl, F., 2009. Two view geometry estimation with outliers. In *British Machine Vision Conference (BMVC)*, vol. 2, 3. (cited on pages 8, 9, 48, and 63)

Farid, H. and Adelson, E. H., 1999. Separating reflections and lighting using independent components analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 262–267. (cited on page 106)

Faugeras, O. D.; Luong, Q.-T.; and Maybank, S. J., 1992. Camera self-calibration: Theory and experiments. In *European Conference on Computer Vision (ECCV)*, 321–334. (cited on page 6)

Faugeras, O. D. and Maybank, S., 1990. Motion from point matches: multiplicity of solutions. *International Journal of Computer Vision (IJCV)*, 4, 3 (1990), 225–246. (cited on page 8)

Fischler, M. A. and Bolles, R. C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24, 6 (1981), 726–740. (cited on pages 5, 8, 22, and 47)

Fitzgibbon, A., 2003. Robust registration of 2D and 3D point sets. *Image and Vision Computing (IVC)*, 21, 13 (2003), 1145–1153. (cited on pages 4, 20, 21, 24, 33, and 34)

Gai, K.; Shi, Z.; and Zhang, C., 2012. Blind separation of superimposed moving images using image statistics. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34, 1 (2012), 19–32. (cited on page 106)

Geiger, A.; Lenz, P.; and Urtasun, R., 2012a. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3354–3361. (cited on pages 14 and 94)

Geiger, A.; Moosmann, F.; Car, O.; and Schuster, B., 2012b. Automatic camera and range sensor calibration using a single shot. In *IEEE International Conference on Robotics and Automation (ICRA)*, 3936–3943. (cited on pages 12 and 67)

Geiger, A.; Moosmann, F.; Car, O.; and Schuster, B., 2012c. Automatic camera and range sensor calibration using a single shot. In *IEEE International Conference on Robotics and Automation (ICRA)*, 3936–3943. (cited on page 19)

Gelfand, N.; Mitra, N. J.; Guibas, L. J.; and Pottmann, H., 2005. Robust global registration. In *Eurographics Symposium on Geometry Processing*, vol. 2, 5–14. (cited on pages 5, 21, and 22)

Gomez, J.-F. V.; Simon, G.; and Berger, M.-O., 2005. Calibration errors in augmented reality: A practical study. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 154–163. (cited on page 66)

Goshen, L. and Shimshoni, I., 2008. Balanced exploration and exploitation model search for efficient epipolar geometry estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30, 7 (2008), 1230–1242. (cited on page 47)

Govindu, V. M., 2001. Combining two-view constraints for motion estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, II–218. (cited on page 9)

Granger, S. and Pennec, X., 2002. Multi-scale EM-ICP: A fast and robust approach for surface registration. In *European Conference on Computer Vision (ECCV)*, 418–432. (cited on page 21)

Graves, A.; Wayne, G.; and Danihelka, I., 2014. Neural turing machines. *CoRR*, abs/1410.5401 (2014). http://arxiv.org/abs/1410.5401. (cited on pages 129 and 130)

Guo, X.; Cao, X.; and Ma, Y., 2014. Robust separation of reflection from multiple images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2195–2202. (cited on pages 106 and 119)

Hadsell, R.; Chopra, S.; and LeCun, Y., 2006. Dimensionality reduction by learning an invariant mapping. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 1735–1742. (cited on page 131)

Harris, C. and Stephens, M., 1988. A combined corner and edge detector. In *Alvey Vision Conference*, vol. 15, 50. (cited on page 8)

Hartley, R. and Li, H., 2012. An efficient hidden variable approach to minimal-case camera motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34, 12 (2012), 2303–2314. (cited on pages 8 and 61)

Hartley, R.; Trumpf, J.; Dai, Y.; and Li, H., 2013. Rotation averaging. *International Journal of Computer Vision*, 103, 3 (2013), 267–305. (cited on page 9)

Hartley, R. and Zisserman, A., 2004a. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edn. (cited on page 25)

Hartley, R. and Zisserman, A., 2004b. *Multiple View Geometry in Computer Vision (2nd Edition)*. Cambridge University Press. (cited on page 50)

Hartley, R. and Zisserman, A., 2005. *Multiple view geometry in computer vision (2nd edition)*. Cambridge university press. (cited on pages 7, 9, and 93)

Hartley, R. I., 1997. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 19, 6 (1997), 580–593. (cited on pages 8 and 47)

Hartley, R. I. and Kahl, F., 2007. Global optimization through searching rotation space and optimal estimation of the essential matrix. In *International Conference on Computer Vision (ICCV)*, 1–8. (cited on pages 20, 22, 48, 52, 53, 54, and 63)

Hartley, R. I. and Kahl, F., 2009. Global optimization through rotation space search. *International Journal of Computer Vision (IJCV)*, 82, 1 (2009), 64–79. (cited on pages 26 and 34)

Hartley, R. I. and Sturm, P., 1997. Triangulation. *Computer Vision and Image Understanding (CVIU)*, 68, 2 (1997), 146–157. (cited on page 54)

Hauck, G., 1883. Neue constructionen der perspective und photogrammetrie.(theorie der trilinearen verwandtschaft ebener systeme, i. artikel.). *Journal für die reine und angewandte Mathematik*, 95 (1883), 1–35. (cited on page 6)

Heller, J.; Havlena, M.; and Pajdla, T., 2012. A branch-and-bound algorithm for globally optimal hand-eye calibration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1608–1615. (cited on pages 48 and 63)

Helmke, U.; Hüper, K.; Lee, P. Y.; and Moore, J., 2007. Essential matrix estimation using gauss-newton iterations on a manifold. *International Journal of Computer Vision (IJCV)*, 74, 2 (2007), 117–136. (cited on page 51)

Herbst, E.; Ren, X.; and Fox, D., 2013. RGB-D Flow: Dense 3-D motion estimation using color and depth. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2276–2282. (cited on page 127)

Herrera C, D.; Kannala, J.; and Heikkil, J., 2012. Joint depth and color camera calibration with distortion correction. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34, 10 (2012), 2058–2064. (cited on pages 11, 66, 81, 82, and 83)

Heyden, A. and Sparr, G., 1999. Reconstruction from calibrated cameras – a new proof of the kruppa-demazure theorem. *Journal of Mathematical Imaging and Vision (JMIV)*, 10, 2 (1999), 123–142. (cited on page 8)

Horaud, R. and Dornaika, F., 1995. Hand-eye calibration. *International Journal of Robotics Research (IJRR)*, 14, 3 (1995), 195–210. (cited on page 67)

Horn, B. K., 1987. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4, 4 (1987), 629–642. (cited on page 23)

Horn, B. K. and Schunck, B. G., 1981. Determining optical flow. In *Artificial Intelligence*, vol. 17, 185âĂŞ203. (cited on pages 1, 12, 14, and 15)

Horn̊áček, M.; Besse, F.; Kautz, J.; Fitzgibbon, A.; and Rother, C., 2014. Highly overparameterized optical flow using patchmatch belief propagation. In *European Conference on Computer Vision (ECCV)*, 220–234. (cited on pages 87 and 97)

Hosni, A.; Rhemann, C.; Bleyer, M.; Rother, C.; and Gelautz, M., 2013. Fast cost-volume filtering for visual correspondence and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35, 2 (2013), 504–511. (cited on page 14)

Hu, J.; Lu, J.; and Tan, Y.-P., 2014a. Discriminative deep metric learning for face verification in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1875–1882. (cited on pages 129 and 132)

Hu, J.; Lu, J.; Yuan, J.; and Tan, Y.-P., 2014b. Large margin multi-metric learning for face and kinship verification in the wild. In *Asian Conference on Computer Vision (ACCV)*, 252–267. (cited on page 132)

Huber, D. F. and Hebert, M., 2003. Fully automatic registration of multiple 3d data sets. *Image and Vision Computing (IVC)*, 21, 7 (2003), 637–650. (cited on page 19)

Ioffe, S. and Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, (2015). (cited on page 130)

Irani, M.; Rousso, B.; and Peleg, S., 1994. Computing occluding and transparent motions. *International Journal of Computer Vision (IJCV)*, 12, 1 (1994), 5–16. (cited on page 105)

Irani, S. and Raghavan, P., 1999. Combinatorial and experimental results for randomized point matching algorithms. *Computational Geometry*, 12, 1 (1999), 17–31. (cited on pages 5 and 22)

Isack, H. and Boykov, Y., 2012. Energy-based geometric multi-model fitting. *International Journal of Computer Vision (IJCV)*, 97, 2 (2012), 123–147. (cited on pages 88 and 91)

Jian, B. and Vemuri, B., 2005. A robust algorithm for point set registration using mixture of gaussians. In *International Conference on Computer Vision (ICCV)*, 1246–1251. (cited on pages 4, 21, and 24)

Jiang, N.; Cui, Z.; and Tan, P., 2013. A global linear method for camera pose registration. In *International Conference on Computer Vision (ICCV)*, 481–488. (cited on page 9)

Johnson, A. E. and Hebert, M., 1999. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 21, 5 (1999), 433–449. (cited on pages 4, 5, 19, and 21)

Johnson, A. E. and Sing, B. K., 1999. Registration and integration of textured 3D data. *Image and Vision Computing (IVC)*, 17, 2 (1999), 135–147. (cited on pages 4 and 21)

Ju, S. X.; Black, M. J.; and Jepson, A. D., 1996. Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 307–314. (cited on pages 85, 87, and 105)

Kahl, F. and Hartley, R., 2008. Multiple-view geometry under the lâĹđ norm. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 30, 9 (2008), 1603–1617. (cited on pages 54 and 61)

Ke, Q. and Kanade, T., 2007. Quasiconvex optimization for robust geometric reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29, 10 (2007), 1834–1847. (cited on pages 9, 49, and 54)

Kennedy, R. and Taylor, C. J., 2015. Optical flow with geometric occlusion estimation and fusion of multiple frames. In *International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, 364–377. (cited on page 101)

Kim, J.-H. et al., 2008. *Camera Motion Estimation for Multi-Camera Systems*. PhD thesis, The Australian National University. (cited on page 7)

Kim, T. H.; Lee, H. S.; and Lee, K. M., 2013. Optical flow via locally adaptive fusion of complementary data costs. In *International Conference on Computer Vision (ICCV)*, 3344–3351. IEEE. (cited on page 85)

Klare, B. F.; Klein, B.; Taborsky, E.; Blanton, A.; Cheney, J.; Allen, K.; Grother, P.; Mah, A.; Burge, M.; and Jain, A. K., 2015. Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1931–1939. (cited on page 131)

Klein, G. and Murray, D., 2007. Parallel tracking and mapping for small ar workspaces. In *International Symposium on Mixed and Augmented Reality (ISMAR)*, 225–234. (cited on page 9)

Klontz, J. C.; Klare, B. F.; Klum, S.; Jain, A. K.; and Burge, M. J., 2013. Open source biometric recognition. In *IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, 1–8. (cited on page 133)

Kolmogorov, V. and Zabin, R., 2004. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26, 2 (2004), 147–159. (cited on page 90)

Kruppa, E., 1913. Zur Ermittlung eines Objektes aus zwei Perspektiven mit innerer Orientierung. *Sitzungsberichte der Mathematisch Naturwissenschaftlichen Kaiserlichen Akademie der Wissenschaften*, 122 (1913), 1939–1948. (cited on page 8)

Lai, K.; Bo, L.; Ren, X.; and Fox, D., 2011. A large-scale hierarchical multi-view RGB-D object dataset. In *IEEE International Conference on Robotics and Automation (ICRA)*, 1817–1824. (cited on page 43)

Langley, K.; Fleet, D.; and Atherton, T., 1992a. Multiple motions from instantaneous frequency. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 846–849. (cited on page 105)

Langley, K.; Fleet, D. J.; and Atherton, T. J., 1992b. On transparent motion computation. In *British Machine Vision Conference (BMVC)*, 247–256. (cited on page 105)

Lawler, E. L. and Wood, D. E., 1966. Branch-and-bound methods: A survey. *Operations Research*, 14, 4 (1966), 699–719. (cited on page 24)

Lee, J., 2010. *Introduction to topological manifolds*. Springer. (cited on page 50)

Lei, C. and Yang, Y.-H., 2009. Optical flow estimation on coarse-to-fine region-trees using discrete optimization. In *International Conference on Computer Vision (ICCV)*, 1562–1569. (cited on page 87)

Lepetit, V.; Moreno-Noguer, F.; and Fua, P., 2009. EPnP: An accurate O(n) solution to the PnP problem. *International Journal on Computer Vision (IJCV)*, 81, 2 (2009), 155–166. (cited on page 75)

Levin, A. and Weiss, Y., 2007. User assisted separation of reflections from a single image using a sparsity prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29, 9 (2007), 1647–1654. (cited on pages 106 and 108)

Levin, A.; Zomet, A.; and Weiss, Y., 2002. Learning to perceive transparency from the statistics of natural scenes. In *Advances in Neural Information Processing Systems (NIPS)*, 1247–1254. (cited on page 106)

Li, H., 2007a. A practical algorithm for l_∞ triangulation with outliers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–8. (cited on page 49)

Li, H., 2007b. Two-view motion segmentation from linear programming relaxation. In *CVPR 2007*. (cited on page 88)

Li, H., 2009. Consensus set maximization with guaranteed global optimality for robust geometry estimation. In *International Conference on Computer Vision (ICCV)*, 1074–1080. (cited on pages 9 and 48)

Li, H., 2010. Multi-view structure computation without explicitly estimating motion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2777–2784. (cited on page 74)

Li, H. and Hartley, R., 2006. Five-point motion estimation made easy. In *International Conference on Pattern Recognition (ICPR)*, vol. 1, 630–633. (cited on pages 8 and 47)

Li, H. and Hartley, R., 2007. The 3D-3D registration problem revisited. In *International Conference on Computer Vision (ICCV)*, 1–8. (cited on pages 5, 20, 22, and 24)

Li, H.; Hua, G.; Lin, Z.; Brandt, J.; and Yang, J., 2013. Probabilistic elastic matching for pose variant face verification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3499–3506. (cited on page 129)

Li, H.; Hua, G.; Shen, X.; Lin, Z.; and Brandt, J., 2014. Eigen-pep for video face recognition. In *Asian Conference on Computer Vision (ACCV)*, 17–33. (cited on pages 129, 132, and 134)

Li, Y. and Brown, M., 2013. Exploiting reflection change for automatic reflection removal. In *International Conference on Computer Vision (ICCV)*, 2432–2439. (cited on pages 106, 114, 119, 122, and 123)

Li, Y. and Brown, M. S., 2014. Single image layer separation using relative smoothness. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2752–2759. (cited on pages 106 and 107)

Li, Y.; Min, D.; Brown, M. S.; Do, M. N.; and Lu, J., 2015a. SPM-BP: Sped-up patchmatch belief propagation for continuous mrfs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4006–4014. (cited on page 14)

Li, Z.; Tan, P.; Tan, R. T.; Zou, D.; Zhou, S. Z.; and Cheong, L.-F., 2015b. Simultaneous video defogging and stereo reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4988–4997. (cited on page 106)

Lindeberg, T., 1998. Feature detection with automatic scale selection. *International Journal of Computer Vision (IJCV)*, 30, 2 (1998), 79–116. (cited on page 8)

Liu, C.; Freeman, W. T.; Adelson, E. H.; and Weiss, Y., 2008. Human-assisted motion annotation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–8. (cited on pages 12 and 104)

Liu, L.; Zhang, L.; Liu, H.; and Yan, S., 2014. Toward large-population face identification in unconstrained videos. *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 24, 11 (2014), 1874–1884. (cited on pages 129, 131, and 134)

Longuet-Higgins, H. C., 1981. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293 (1981), 133–135. (cited on pages 1, 6, 8, and 47)

Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 60, 2 (2004), 91–110. (cited on page 8)

Lu, J.; Yang, H.; Min, D.; and Do, M., 2013. Patch match filter: Efficient edge-aware filtering meets randomized search for fast correspondence field estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1854–1861. (cited on pages 14 and 93)

Lucas, B. D. and Kanade, T., 1981. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 81, 674–679. (cited on pages 1, 12, and 14)

Magnusson, M.; Nuchter, A.; Lorken, C.; Lilienthal, A. J.; and Hertzberg, J., 2009. Evaluation of 3D registration reliability and speed-a comparison of ICP and NDT. In *IEEE International Conference on Robotics and Automation (ICRA)*, 3907–3912. (cited on page 21)

Mahamud, S. and Hebert, M., 2000. Iterative projective reconstruction from multiple views. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 430–437. (cited on page 10)

Mahamud, S.; Hebert, M.; Omori, Y.; and Ponce, J., 2001. Provably-convergent iterative methods for projective structure from motion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, I–1018. (cited on page 10)

Makadia, A.; Geyer, C.; and Daniilidis, K., 2007. Correspondence-free structure from motion. *International Journal of Computer Vision (IJCV)*, 75, 3 (2007), 311–327. (cited on page 49)

Makadia, A.; Patterson, A.; and Daniilidis, K., 2006. Fully automatic registration of 3D point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 1297–1304. (cited on pages 4, 5, 20, and 21)

Makela, T.; Clarysse, P.; Sipila, O.; Pauna, N.; Pham, Q. C.; Katila, T.; and Magnin, I. E., 2002. A review of cardiac image registration methods. *IEEE Transactions on Medical Imaging*, 21, 9 (2002), 1011–1021. (cited on pages 3 and 19)

Martinec, D. and Pajdla, T., 2007. Robust rotation and translation estimation in multiview reconstruction. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 1–8. (cited on page 9)

Masuda, T. and Yokoya, N., 1994. A robust method for registration and segmentation of multiple range images. In *CAD-Based Vision Workshop*, 106–113. (cited on pages 4 and 33)

Mémin, E. and Pérez, P., 2002. Hierarchical estimation and segmentation of dense motion fields. *International Journal of Computer Vision (IJCV)*, 46, 2 (2002), 129–155. (cited on page 87)

Menze, M. and Geiger, A., 2015. Object scene flow for autonomous vehicles. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3061–3070. (cited on page 127)

Mian, A. S.; Bennamoun, M.; and Owens, R., 2006. Three-dimensional model-based object recognition and segmentation in cluttered scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28, 10 (2006), 1584–1601. (cited on page 44)

Mikolajczyk, K. and Schmid, C., 2004. Scale & affine invariant interest point detectors. *International Journal of Computer Vision (IJCV)*, 60, 1 (2004), 63–86. (cited on page 8)

Moré, J. J., 1978. The levenberg-marquardt algorithm: implementation and theory. *Numerical analysis*, (1978), 105–116. (cited on pages 4, 21, and 72)

Moulon, P.; Monasse, P.; and Marlet, R., 2013. Global fusion of relative motions for robust, accurate and scalable structure from motion. In *International Conference on Computer Vision (ICCV)*, 3248–3255. (cited on page 9)

Mount, D. M.; Netanyahu, N. S.; and Le Moigne, J., 1999. Efficient algorithms for robust feature matching. *Pattern Recognition (PR)*, 32, 1 (1999), 17–38. (cited on pages 5 and 22)

Musser, D. R., 1997. Introspective sorting and selection algorithms. *Software: Practice and Experience*, 27, 8 (1997), 983–993. (cited on page 33)

Myronenko, A. and Song, X., 2010. Point set registration: coherent point drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32, 12 (2010), 2262–2275. (cited on pages 4 and 21)

Nelder, J. A. and Mead, R., 1965. A simplex method for function minimization. *The Computer Journal*, 7, 4 (1965), 308–313. (cited on pages 72, 92, and 93)

Newcombe, R. A.; Davison, A. J.; Izadi, S.; Kohli, P.; Hilliges, O.; Shotton, J.; Molyneaux, D.; Hodges, S.; Kim, D.; and Fitzgibbon, A., 2011. Kinectfusion: Real-time dense surface mapping and tracking. *IEEE International Symposium on Mixed and Augmented Reality*, (2011), 127–136. (cited on pages 3 and 19)

Ni, K.; Jin, H.; and Dellaert, F., 2009. Groupsac: Efficient consensus in the presence of groupings. In *International Conference on Computer Vision (ICCV)*, 2193–2200. (cited on page 8)

Nir, T.; Bruckstein, A. M.; and Kimmel, R., 2008. Over-parameterized variational optical flow. *International Journal of Computer Vision (IJCV)*, 76, 2 (2008), 205–216. (cited on page 87)

Nistér, D., 2003. An efficient solution to the five-point relative pose problem. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 195–202. (cited on page 8)

Nistér, D., 2004. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26, 6 (2004), 756–770. (cited on pages 8, 47, and 61)

Nistér, D., 2005. Preemptive RANSAC for live structure and motion estimation. *Machine Vision and Applications (MVA)*, 16, 5 (2005), 321–329. (cited on page 8)

Nüchter, A.; Lingemann, K.; Hertzberg, J.; and Surmann, H., 2007. 6D SLAM–3D mapping outdoor environments. *Journal of Field Robotics*, 24, 8-9 (2007), 699–722. (cited on page 19)

Oliensis, J. and Hartley, R., 2007. Iterative extensions of the sturm/triggs algorithm: Convergence and nonconvergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 29, 12 (2007), 2217–2233. (cited on page 10)

Olsson, C. and Boykov, Y., 2012. Curvature-based regularization for surface approximation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1576–1583. (cited on page 88)

Olsson, C.; Eriksson, A.; and Hartley, R., 2010. Outlier removal using duality. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1450–1457. (cited on pages 9 and 49)

Olsson, C.; Kahl, F.; and Oskarsson, M., 2009. Branch-and-bound methods for euclidean registration problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 31, 5 (2009), 783–794. (cited on pages 22 and 23)

Papazov, C. and Burschka, D., 2011. Stochastic global optimization for robust point set registration. *Computer Vision and Image Understanding (CVIU)*, 115, 12 (2011), 1598–1609. (cited on pages 4 and 21)

Parkhi, O. M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A., 2014. A compact and discriminative face track descriptor. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1693–1700. (cited on page 129)

Pfeuffer, F.; Stiglmayr, M.; and Klamroth, K., 2012. Discrete and geometric branch and bound algorithms for medical image registration. *Annals of Operations Research*, 196, 1 (2012), 737–765. (cited on pages 5 and 22)

Philip, J., 1996. A non-iterative algorithm for determining all essential matrices corresponding to five point pairs. *The Photogrammetric Record*, 15, 88 (1996), 589–599. (cited on page 8)

Pilet, J.; Geiger, A.; Lagger, P.; Lepetit, V.; and Fua, P., 2006. An all-in-one solution to geometric and photometric calibration. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 69–78. (cited on page 66)

Pingault, M. and Pellerin, D., 2002. Optical flow constraint equation extended to transparency. In *European Signal Processing Conference*, 1–4. (cited on page 105)

Pollefeys, M.; Koch, R.; and Van Gool, L., 1999. Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *International Journal of Computer Vision (IJCV)*, 32, 1 (1999), 7–25. (cited on page 6)

Pollefeys, M.; Van Gool, L.; Vergauwen, M.; Verbiest, F.; Cornelis, K.; Tops, J.; and Koch, R., 2004. Visual modeling with a hand-held camera. *International Journal of Computer Vision (IJCV)*, 59, 3 (2004), 207–232. (cited on page 9)

Pomerleau, F.; Colas, F.; Siegwart, R.; and Magnenat, S., 2013. Comparing ICP variants on real-world data sets. *Autonomous Robots*, 34, 3 (2013), 133–148. (cited on page 19)

Pulli, K., 1999. Multiview registration for large data sets. In *International Conference on 3-D Digital Imaging and Modeling*, 160–168. (cited on page 4)

Quiroga, J.; Brox, T.; Devernay, F.; and Crowley, J., 2014. Dense semi-rigid scene flow estimation from rgbd images. In *European Conference on Computer Vision (ECCV)*, 567–582. (cited on page 127)

Raguram, R.; Chum, O.; Pollefeys, M.; Matas, J.; and Frahm, J., 2013. USAC: A universal framework for random sample consensus. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35, 8 (2013), 2022–2038. (cited on pages 8 and 47)

Ramirez-Manzanares, A.; Rivera, M.; Kornprobst, P.; and Lauze, F., 2006. Multi-valued motion fields estimation for transparent sequences with a variational approach. *INRIA Technical Report*, (2006). (cited on page 105)

Ranftl, R.; Bredies, K.; and Pock, T., 2014. Non-local total generalized variation for optical flow estimation. In *European Conference on Computer Vision (ECCV)*, 439–454. (cited on pages 15 and 96)

Rangarajan, A.; Chui, H.; Mjolsness, E.; Pappu, S.; Davachi, L.; Goldman-Rakic, P.; and Duncan, J., 1997. A robust point-matching algorithm for autoradiograph alignment. *Medical Image Analysis*, 1, 4 (1997), 379–398. (cited on page 21)

Recht, B.; Fazel, M.; and Parrilo, P. A., 2010. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52, 3 (2010), 471–501. (cited on page 75)

Revaud, J.; Weinzaepfel, P.; Harchaoui, Z.; and Schmid, C., 2015. EpicFlow: Edge-preserving interpolation of correspondences for optical flow. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1164–1172. (cited on page 12)

Robertson, C. and Fisher, R. B., 2002. Parallel evolutionary registration of range data. *Computer Vision and Image Understanding (CVIU)*, 87, 1 (2002), 39–50. (cited on pages 4 and 21)

Rother, C.; Bordeaux, L.; Hamadi, Y.; and Blake, A., 2006. Autocollage. In *ACM Transactions on Graphics (TOG)*, vol. 25, 847–852. (cited on page 88)

Roussos, A.; Russell, C.; Garg, R.; and Agapito, L., 2012. Dense multibody motion estimation and reconstruction from a handheld camera. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 31–40. (cited on page 87)

Rublee, E.; Rabaud, V.; Konolige, K.; and Bradski, G., 2011. ORB: An efficient alternative to SIFT or SURF. In *International Conference on Computer Vision (ICCV)*, 2564–2571. (cited on page 8)

Rudin, L. I.; Osher, S.; and Fatemi, E., 1992. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60, 1 (1992), 259–268. (cited on page 15)

Ruland, T.; Pajdla, T.; and Kruger, L., 2012. Globally optimal hand-eye calibration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1035–1042. (cited on page 22)

Rusinkiewicz, S. and Levoy, M., 2001. Efficient variants of the icp algorithm. In *International Conference on 3-D Digital Imaging and Modeling (3DIM)*, 145–152. (cited on page 24)

Russell, C.; Fayad, J.; and Agapito, L., 2011. Energy based multiple model fitting for non-rigid structure from motion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3009–3016. (cited on page 88)

Rusu, R. B.; Blodow, N.; and Beetz, M., 2009. Fast point feature histograms (FPFH) for 3D registration. In *IEEE International Conference on Robotics and Automation (ICRA)*, 3212–3217. (cited on pages 4, 5, 20, and 21)

Sandhu, R.; Dambreville, S.; and Tannenbaum, A., 2010. Point set registration via particle filtering and stochastic dynamics. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 32, 8 (2010), 1459–1473. (cited on pages 4, 20, and 21)

Sarel, B. and Irani, M., 2004. Separating transparent layers through layer information exchange. In *European Conference on Computer Vision (ECCV)*, 328–341. (cited on page 106)

Scaramuzza, D.; Harati, A.; and Siegwart, R., 2007. Extrinsic self calibration of a camera and a 3D laser range finder from natural scenes. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4164–4169. (cited on pages 11 and 67)

Scaramuzza, D.; Martinelli, A.; and Siegwart, R., 2006. A toolbox for easily calibrating omnidirectional cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5695–5701. (cited on page 63)

Schechner, Y. Y.; Kiryati, N.; and Basri, R., 2000. Separation of transparent layers using focus. *International Journal of Computer Vision (IJCV)*, 39, 1 (2000), 25–39. (cited on page 106)

Schroff, F.; Kalenichenko, D.; and Philbin, J., 2015. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 815–823. (cited on pages 129 and 132)

Seitz, S. M.; Curless, B.; Diebel, J.; Scharstein, D.; and Szeliski, R., 2006. A comparison and evaluation of multi-view stereo reconstruction algorithms. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1 (2006), 519–528. (cited on pages 3 and 19)

Senst, T.; Eiselein, V.; and Sikora, T., 2012. Robust local optical flow for feature tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 22, 9 (2012), 1377–1387. (cited on page 14)

Sevilla-Lara, L.; Sun, D.; Jampani, V.; and Black, M. J., 2016. Optical flow with semantic segmentation and localized layers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3889–3898. (cited on page 126)

Sharp, G. C.; Lee, S. W.; and Wehe, D. K., 2002. ICP registration using invariant features. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24, 1 (2002), 90–102. (cited on pages 4 and 21)

Shi, J. and Tomasi, C., 1994. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 593–600. (cited on page 8)

Shizawa, M. and Mase, K., 1990. Simultaneous multiple optical flow estimation. In *International Conference on Pattern Recognition (ICPR)*, 274–278. (cited on pages 105 and 109)

Shizawa, M. and Mase, K., 1991. Unified computational theory for motion transparency and motion boundaries based on eigenenergy analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 289–295. (cited on page 105)

Shotton, J.; Glocker, B.; Zach, C.; Izadi, S.; Criminisi, A.; and Fitzgibbon, A., 2013. Scene coordinate regression forests for camera relocalization in RGB-D images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2930–2937. (cited on pages 3, 43, and 45)

Silva, L.; Bellon, O. R. P.; and Boyer, K. L., 2005. Precision range image registration using a robust surface interpenetration measure and enhanced genetic algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 27, 5 (2005), 762–776. (cited on pages 4, 20, and 21)

Sim, K. and Hartley, R., 2006. Removing outliers using the l_∞ norm. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 485–494. (cited on pages 9 and 49)

Simon, C. and Park, I. K., 2015. Reflection removal for in-vehicle black box videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4231–4239. (cited on page 106)

Slama, C. C.; Theurer, C.; Henriksen, S. W.; et al., 1980. *Manual of Photogrammetry.* Ed. 4. American Society of Photogrammetry. (cited on page 6)

Smisek, J.; Jancosek, M.; and Pajdla, T., 2011. 3D with kinect. In *ICCV Workshop on Consumer Depth Cameras for Computer Vision*, 1154–1160. (cited on pages 11 and 67)

Snavely, N.; Seitz, S. M.; and Szeliski, R., 2006. Photo tourism: exploring photo collections in 3D. In *ACM transactions on graphics (TOG)*, vol. 25, 835–846. (cited on page 9)

Steinbrucker, F.; Pock, T.; and Cremers, D., 2009. Large displacement optical flow computation without warping. In *International Conference on Computer Vision (ICCV)*, 1609–1614. IEEE. (cited on pages 112 and 114)

Sturm, P. and Triggs, B., 1996. A factorization based algorithm for multi-image projective structure and motion. In *European Conference on Computer Vision (ECCV)*, 709–720. (cited on page 10)

Subbarao, R.; Genc, Y.; and Meer, P., 2008. Robust unambiguous parametrization of the essential manifold. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–8. (cited on page 51)

Sun, D.; Roth, S.; and Black, M., 2014a. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *International Journal of Computer Vision (IJCV)*, 106, 2 (2014), 115–137. (cited on page 105)

Sun, D.; Roth, S.; and Black, M. J., 2010a. Secrets of optical flow estimation and their principles. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2432–2439. (cited on page 15)

Sun, D.; Roth, S.; and Black, M. J., 2014b. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *International Journal of Computer Vision (IJCV)*, 106, 2 (2014), 115–137. (cited on pages 12, 16, 85, 94, 96, 97, 99, 100, and 101)

Sun, D.; Sudderth, E. B.; and Black, M. J., 2010b. Layered image motion with explicit occlusions, temporal consistency, and depth ordering. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2226–2234. (cited on pages 86, 87, 99, and 105)

Sun, D.; Sudderth, E. B.; and Black, M. J., 2012. Layered segmentation and optical flow estimation over time. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1768–1775. (cited on pages 86 and 87)

Sun, D.; Sudderth, E. B.; and Pfister, H., 2015a. Layered RGBD scene flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 548–556. (cited on page 127)

Sun, Y.; Wang, X.; and Tang, X., 2015b. Deeply learned face representations are sparse, selective, and robust. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2892–2900. (cited on page 132)

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A., 2015. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–9. (cited on page 130)

Szeliski, R.; Avidan, S.; and Anandan, P., 2000. Layer extraction from multiple images containing reflections and transparency. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 246–253. (cited on page 106)

Szeliski, R. and Golland, P., 1998. Stereo matching with transparency and matting. In *International Conference on Computer Vision (ICCV)*, 517–524. (cited on pages 106 and 107)

Taigman, Y.; Yang, M.; Ranzato, M.; and Wolf, L., 2014. Deepface: Closing the gap to human-level performance in face verification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1701–1708. (cited on pages 129 and 132)

Tappen, M. F.; Freeman, W. T.; and Adelson, E. H., 2005. Recovering intrinsic images from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 27, 9 (2005), 1459–1472. (cited on pages 106 and 108)

Thompson, E. H., 1959. A rational algebraic formulation of the problem of relative orientation. *The Photogrammetric Record*, 3, 14 (1959), 152–157. (cited on page 6)

Toh, K.; Todd, M.; and Tutuncu, R., 1999. SDPT3 — a matlab software package for semidefinite programming. *Optimization Methods and Software*, 11 (1999), 545–581. (cited on page 74)

Tomasi, C. and Kanade, T., 1992. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision (IJCV)*, 9, 2 (1992), 137–154. (cited on page 10)

Toro, J.; Owens, F. J.; and Medina, R., 2000. Multiple motion estimation and segmentation in transparency. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 6, 2087–2090. (cited on page 105)

Triggs, B., 1996. Factorization methods for projective structure and motion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 845–851. (cited on page 10)

Triggs, B., 2000. Routines for relative pose of two calibrated cameras from 5 points. Technical Report INRIA-Technical Report, Robotics Institute, Carnegie Mellon University, INRIA, France. (cited on page 8)

Triggs, B.; McLauchlan, P. F.; Hartley, R. I.; and Fitzgibbon, A. W., 1999. Bundle adjustment – a modern synthesis. In *International Workshop on Vision Algorithms*, 298–372. (cited on page 9)

Tron, R. and Daniilidis, K., 2014. On the quotient representation for the essential manifold. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1574–1581. (cited on pages 50 and 51)

Tron, R.; Vidal, R.; and Terzis, A., 2008. Distributed pose averaging in camera networks via consensus on SE(3). In *ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, 1–10. (cited on page 72)

Tsai, R. Y. and Lenz, R. K., 1989. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation*, 5, 3 (1989), 345–358. (cited on page 67)

Tsin, Y. and Kanade, T., 2004. A correlation-based approach to robust point set registration. In *European Conference on Computer Vision (ECCV)*, 558–569. (cited on pages 4, 20, and 21)

Tsin, Y.; Kang, S. B.; and Szeliski, R., 2006. Stereo matching with linear superposition of layers. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28, 2 (2006), 290–301. (cited on page 106)

Unger, M.; Werlberger, M.; Pock, T.; and Bischof, H., 2012. Joint motion estimation and segmentation of complex scenes with label costs and occlusion modeling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1878–1885. (cited on pages xiv, 86, 87, 98, 99, and 100)

Vasconcelos, F.; Barreto, J.; and Nunes, U., 2012. A minimal solution for the extrinsic calibration of a camera and a laser-rangefinder. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34, 11 (2012), 2097–2107. (cited on pages 11 and 67)

Vinyals, O.; Bengio, S.; and Kudlur, M., 2016. Order matters: sequence to sequence for sets. In *International Conference on Learning Representation (ICLR)*. (cited on page 129)

Vogel, C.; Roth, S.; and Schindler, K., 2014. View-consistent 3d scene flow estimation over multiple frames. In *European Conference on Computer Vision (ECCV)*, 263–278. (cited on page 127)

Vogel, C.; Schindler, K.; and Roth, S., 2013. Piecewise rigid scene flow. In *International Conference on Computer Vision (ICCV)*, 1377–1384. (cited on page 87)

Vogel, C.; Schindler, K.; and Roth, S., 2015. 3d scene flow estimation with a piecewise rigid scene model. *International Journal of Computer Vision (IJCV)*, 115, 1 (2015), 1–28. (cited on page 127)

Von Sanden, H., 1908. *Die Bestimmung der Kernpunkte in der Photogrammetrie*. PhD thesis, University of Göttingen. (cited on page 6)

Wachowiak, M. P.; Smolíková, R.; Zheng, Y.; Zurada, J. M.; and Elmaghraby, A. S., 2004. An approach to multimodal biomedical image registration utilizing particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 8, 3 (2004), 289–301. (cited on pages 4 and 21)

Wang, D.; Otto, C.; and Jain, A. K., 2015. Face search at scale: 80 million gallery. *arXiv preprint arXiv:1507.07242*, (2015). (cited on page 133)

Wang, J. Y. and Adelson, E. H., 1994. Representing moving images with layers. *IEEE Transactions on Image Processing (TIP)*, 3, 5 (1994), 625–638. (cited on pages 87 and 105)

Wedel, A.; Brox, T.; Vaudrey, T.; Rabe, C.; Franke, U.; and Cremers, D., 2011. Stereoscopic scene flow computation for 3d motion understanding. *International Journal of Computer Vision (IJCV)*, 95, 1 (2011), 29–51. (cited on page 127)

Wedel, A.; Pock, T.; Zach, C.; Bischof, H.; and Cremers, D., 2009. An improved algorithm for tv-l 1 optical flow. In *Statistical and Geometrical Approaches to Visual Motion Analysis*, 23–45. (cited on page 15)

Weinzaepfel, P.; Revaud, J.; Harchaoui, Z.; and Schmid, C., 2013. DeepFlow: Large displacement optical flow with deep matching. In *International Conference on Computer Vision (ICCV)*, 1385–1392. (cited on pages 96 and 101)

Weiss, Y., 1997. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 520–526. (cited on page 105)

Weiss, Y., 2001. Deriving intrinsic images from image sequences. In *International Conference on Computer Vision (ICCV)*, 68–75. (cited on page 106)

Wexler, Y.; Fitzgibbon, A.; and Zisserman, A., 2002. Bayesian estimation of layers from multiple images. In *European Conference on Computer Vision (ECCV)*, 487–501. (cited on page 106)

Wilczkowiak, M.; Sturm, P.; and Boyer, E., 2005. Using geometric constraints through parallelepipeds for calibration and 3d modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 27, 2 (2005), 194–207. (cited on page 75)

Wills, J.; Agarwal, S.; and Belongie, S., 2003. What went where. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 37–44. (cited on page 87)

Wolf, L.; Hassner, T.; and Maoz, I., 2011. Face recognition in unconstrained videos with matched background similarity. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 529–534. (cited on pages 129 and 131)

Wolf, L. and Levy, N., 2013. The SVM-Minus similarity score for video face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3523–3530. (cited on page 129)

Woodford, O. J.; Pham, M.-T.; Maki, A.; Perbet, F.; and Stenger, B., 2014. Demisting the hough transform for 3d shape recognition and registration. *International Journal of Computer Vision (IJCV)*, 106, 3 (2014), 332–341. (cited on pages 5 and 21)

Wulff, J. and Black, M. J., 2014. Modeling blurred video with layers. In *European Conference on Computer Vision (ECCV)*, 236–252. (cited on page 105)

Xu, L.; Chen, J.; and Jia, J., 2008. A segmentation based variational model for accurate optical flow estimation. In *European Conference on Computer Vision (ECCV)*, 671–684. (cited on page 87)

Xu, L.; Jia, J.; and Matsushita, Y., 2012. Motion detail preserving optical flow estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34, 9 (2012), 1744–1757. (cited on pages 12, 85, 86, 93, 99, and 101)

Xue, T.; Rubinstein, M.; Liu, C.; and Freeman, W. T., 2015. A computational approach for obstruction-free photography. *ACM Transactions on Graphics (TOG)*, 34, 4 (2015), 79. (cited on page 106)

Yamaguchi, K.; McAllester, D.; and Urtasun, R., 2013. Robust monocular epipolar flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1862–1869. (cited on page 87)

Yang, J.; Dai, Y.; Li, H.; Gardner, H.; and Jia, Y., 2013a. Single-shot extrinsic calibration of a generically configured RGB-D camera rig from scene constraints. In *International Symposium on Mixed and Augmented Reality (ISMAR)*, 181–188. (cited on page 19)

Yang, J. and Li, H., 2015. Dense, accurate optical flow estimation with piecewise parametric model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1019–1027. (cited on page 105)

Yang, J.; Li, H.; and Jia, Y., 2013b. Go-ICP: Solving 3D regstration efficiently and globally optimally. In *International Conference on Computer Vision (ICCV)*, 1457–1464. (cited on page 48)

Yang, J.; Li, H.; and Jia, Y., 2014. Optimal essential matrix estimation via inlier-set maximization. In *European Conference on Computer Vision (ECCV)*, 111–126. (cited on page 22)

Ye, G.; Garces, E.; Liu, Y.; Dai, Q.; and Gutierrez, D., 2014. Intrinsic video and applications. *ACM Transactions on Graphics (TOG)*, 33, 4 (2014), 80. (cited on page 106)

Yeung, S.-K.; Wu, T.-P.; and Tang, C.-K., 2008. Extracting smooth and transparent layers from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–7. (cited on pages 106 and 107)

Zach, C.; Pock, T.; and Bischof, H., 2007. A duality based approach for realtime tv-l 1 optical flow. In *Pattern Recognition*, 214–223. Springer. (cited on pages 12, 15, 85, 89, 108, 109, 112, and 113)

Zhang, C.; Li, Z.; Cai, R.; Chao, H.; and Rui, Y., 2014. As-rigid-as-possible stereo under second order smoothness priors. In *European Conference on Computer Vision (ECCV)*, 112–126. (cited on page 93)

Zhang, C. and Zhang, Z., 2011. Calibration between depth and color sensors for commodity depth cameras. In *IEEE International Conference on Multimedia and Expo (ICME)*, 1–6. (cited on pages 11 and 66)

ZHANG, Q. AND PLESS, R., 2004. Extrinsic calibration of a camera and laser range finder (improves camera calibration). In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, 2301–2306. (cited on pages 11, 66, and 67)

ZHANG, Z., 1994. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision (IJCV)*, 13, 2 (1994), 119–152. (cited on pages 3 and 19)

ZHANG, Z., 2000. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22, 11 (2000), 1330–1334. (cited on pages 6, 11, and 66)

ZHANG, Z.; ISONO, K.; AND AKAMATSU, S., 1998. Euclidean structure from uncalibrated images using fuzzy domain knowledge: application to facial images synthesis. In *International Conference on Computer Vision (ICCV)*, 784–789. (cited on page 75)

ZHAO, W.; NISTER, D.; AND HSU, S., 2005. Alignment of continuous video onto 3d point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 27, 8 (2005), 1305–1318. (cited on page 19)

ZHENG, Y.; SUGIMOTO, S.; AND OKUTOMI, M., 2011. A branch and contract algorithm for globally optimal fundamental matrix estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2953–2960. (cited on page 48)

ZITNICK, C. L.; JOJIC, N.; AND KANG, S. B., 2005. Consistent segmentation for optical flow estimation. In *International Conference on Computer Vision (ICCV)*, vol. 2, 1308–1315. (cited on page 87)