Efficient Extraction and Representation of Spatial Information from Video Data

Hajar Sadeghi Sokeh, Stephen Gould and Jochen Renz
Research School of Computer Science
The Australian National University
Canberra, ACT 0200
{hajar.sadeghi,stephen.gould,jochen.renz}@anu.edu.au

Abstract

Vast amounts of video data are available on the web and are being generated daily using surveillance cameras or other sources. Being able to efficiently analyse and process this data is essential for a number of different applications. We want to be able to efficiently detect activities in these videos or be able to extract and store essential information contained in these videos for future use and easy search and access. Cohn et al. (2012) proposed a comprehensive representation of spatial features that can be efficiently extracted from video and used for these purposes. In this paper, we present a modified version of this approach that is equally efficient and allows us to extract spatial information with much higher accuracy than previously possible. We present efficient algorithms both for extracting and storing spatial information from video, as well as for processing this information in order to obtain useful spatial features. We evaluate our approach and demonstrate that the extracted spatial information is considerably more accurate than that obtained from existing approaches.

1 Introduction

The majority of data available on the web is video data. New video data is continuously being generated in mind-blowing amounts, for example, from surveillance cameras, webcams, and user uploaded videos on websites such as YouTube. Automatically analysing and processing this video data is a major challenge. Several important tasks are to automatically detect and classify activities in videos, or to detect essential features and to represent them in a compact way. This allows us to achieve fast indexing and search of videos, frame by frame, as well as meaningful input for machine learning tools or other future processing of the relevant information.

Some of the most useful information that can be extracted from video are the changing spatial properties of objects in video and their changing relationships with other objects. These changing properties and relationships are often characteristic of particular activities. It is then possible to express rules in terms of these changes or to learn activities based on similar change patterns [Sridhar *et al.*, 2011a].

The actual location and size of objects in video depends on a number of factors, such as the position and angle of the camera or its internal parameters (e.g., zoom). Therefore, it is not very useful to represent spatial properties of objects and their relationship with other objects in exact numerical form, for example exact coordinates, exact pixel size of objects, exact pixel distance between objects, or exact angles. Instead, it is more useful to look at qualitative relationships between objects, such as topology, relative size, qualitative direction, or qualitative distance, and how these qualitative relationships change over time.

In the area of *qualitative spatial representation and reasoning* (QSR) [Cohn and Renz, 2008], these qualitative relationships are formalized and analyzed. Typically, each aspect of space, such as topology, distance, direction, etc., are represented using an individual *qualitative calculus*. Instead of extracting each of these aspects separately, Cohn et al. [2012] developed a comprehensive representation of spatial information, called CORE-9, that integrates all the important aspects. What makes CORE-9 an ideal representation for video processing is the fact that it relies purely on obtaining minimal bounding boxes of the objects in each video frame. This is a standard task in computer vision that can be done very efficiently and relatively accurately using object tracking methods [Leal-Taixe, 2012] or from the output of a sliding-window object detector [Viola and Jones, 2004].

Cohn et al. showed that all the important aspects of space and the corresponding spatial calculi (such as RCC-8 [Randell *et al.*, 1992] or STAR [Renz and Mitra, 2004]) can be obtained easily from the CORE-9 representation. They also showed that their representation is useful for learning activities from tagged videos and subsequently detecting these activities in untagged videos.

Now while CORE-9 is a very comprehensive representation that can be efficiently extracted from video data, the main problem is that using axis-aligned bounding boxes to represent objects is very inaccurate. For example, objects that are relatively close to each other typically have bounding boxes that overlap, while the objects themselves do not overlap. In this paper we propose a modification of CORE-9 that is equally easy to obtain, but that considerably increases the accuracy of the spatial information we extract. An increased accuracy will consequently lead to better activity detection, which our empirical evaluation confirms.

The paper is structured as follows. In the next section, we review prior work in this area. We analyse the performance of CORE-9 with respect to different aspects of space and its inability to identify certain changes over time. In Section 3 we present and formalize our new approach. In Section 4 we develop algorithms that allow us to efficiently extract the different aspects of space from our new formalism and investigate how this differs from what we get from CORE-9. In Sections 5 and 6 we evaluate the performance of our approach on real video data with respect to the accuracy of detecting spatial features and learning activities. We then discuss our results and possible future work.

2 Knowledge Representation Formalisms for Video Understanding

There is an increasing interest in involving knowledge representation methods in the area of video analysis and understanding [Sridhar *et al.*, 2011b]. Morariu and Davis [2011] used logic to encode knowledge about spatio-temporal structure to automatically analyse video and detect activities. Spatio-temporal relations were also used by Sridhar et al. [2011a] as a relational graph to describe video activities and understand the video. Sridhar et al. [2010] used unsupervised learning with spatial relations to recognize multiple actions happening together.

One of the most recent approaches is CORE-9 [Cohn *et al.*, 2012] which extracts comprehensive spatial information using minimal bounding rectangles/boxes (MBBs) parallel to the video frame for all tracked objects. For every pair of tracked objects a and b, CORE-9 takes the corresponding MBBs A and B and the coordinates of two diagonally opposite corners of each MBB. Extrapolating from these four corners $(x_1, y_1), \ldots, (x_4, y_4)$ in the x- and y-direction forms a 3×3 grid consisting of nine *cores* (see Figure 3b). Each of these cores can be contained in A, in B, in both or in none, which gives a total of 169 different assignments of the nine cores, called *states*. We can also compare the relative size of the nine cores and the relative size of intervals formed from the four points $x_1, \ldots, x_4$ and the four points $y_1, \ldots, y_4$. These relative sizes of intervals allow us to obtain qualitative information about the relative distance of MBBs, both externally and internally. We can also easily express changes over time of these elements. Cohn et al. demonstrated that a number of different aspects of space and their changes can be obtained from the initial four corner points. So the only thing that needs to be extracted from a video frame are these corner points, which can be done very efficiently.

3 CORE-9 with Different Angles

The problem with CORE-9 is that the relations between MBBs do not accurately reflect the actual relations between the objects, which is well known. This restricts the acquired information for different aspects of space like topology, size and distance. For example, the objects in Figure 3b are clearly disjoint, while their MBBs overlap.

Our aim in this paper is to increase the accuracy of the spatial information we can easily extract while keeping the representation simple and comprehensive at the same time. The proposed idea in this paper is to use minimal bounding boxes at different angles, not just ones that are parallel to the video frame. If we look at the regions from more than one direction, there is the possibility of getting more information about the regions (see Figure 4). Stereo vision uses a similar approach, where two different views of a scene give us more information than just using one view.[1]

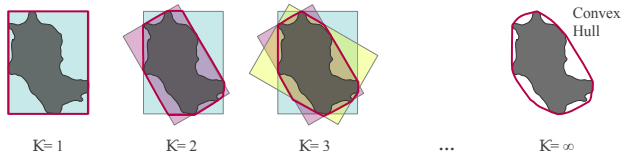


Figure 1: Using MBBs of different angles gives us a better approximation of region shapes. The convex hull is the limit.

For each new angle, we can use the same CORE-9 representation, but with respect to a particular angle α. We write CORE$_\alpha$-9 to indicate which angle we are using to represent our bounding boxes and assume that $\alpha = 0$ corresponds to the original CORE-9. We call this representation *AngledCORE-9* or *MultiAngledCORE-9* and write it as CORE$^9_{\alpha_1,\ldots,\alpha_k}$.

The question now is what is the best angle to use, or if using multiple angles how many and which angles to use? It is clear that when using an infinite number of different angles to represent the regions, the intersection of the bounding boxes will give the convex hull of the object (see Figure 1).

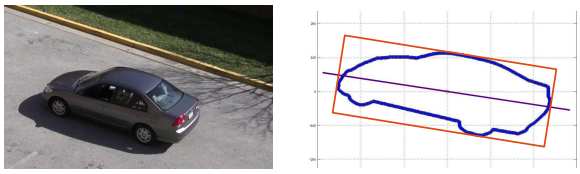
While the convex hull can obviously be obtained more easily than using an infinite number of bounding boxes, it clearly demonstrates the limit of using multiple angles. The more angles we use, the more accurate information we get, but also the higher the computational cost. This cost is even higher if we use the real shape of the object—we might get more accurate representations, but it would require us to identify and store the shape of the convex hull or the real boundary of objects. Our aim is, therefore, to use the smallest number of angles that give us a good accuracy and can still be efficiently extracted and represented. In the following we analyse different possible angles to use.

3.1 Identifying Good Angles

Each region has one or more tight bounding boxes that fits it best, i.e., that contains the least amount of "white space" not belonging to the region. Identifying one of these bounding boxes requires us to test and compare a potentially large number of different MBBs with different angles. We need to be able to determine angles quickly, so we approximate the tightest bounding box using Principle Component Analysis (PCA) [Pearson, 1901] commonly used in image processing.

PCA is one of the simplest and most robust ways of finding the direction of maximum variance of a set of points. Figure 2 shows application of PCA for finding the direction which the object is spread along. In our approach we apply PCA to the

[1] In the case of stereo vision a 3D scene is projected onto two different 2D planes. In our case a 2D object is projected onto different 1D axes.



(a) original frame (b) the direction by PCA

Figure 2: The tightest bounding box of object by PCA

set of boundary pixels of an object making it very fast. For example, for a big object like the one shown in Figure 2, it takes only 42 μs to find the eigenvector (the direction of maximum variance) of the boundary pixels. We denote the angle with the most variance for a region a as α_a. This gives us two candidates for good angles for CORE$_\alpha$-9, one from each object.

Spatial aspects such as topology or distance require us to separate objects as much as possible. The angle that separates two rectangles most seems like another useful angle. Again, we would have to test a potentially large number of angles to find the best one. Instead, we use a well-known machine learning method that is typically used to find the best hyperplane that separates two or more clusters. The so-called Maximum Margin (MM) technique, which is the principle behind support vector machine classification [Cortes and Vapnik, 1995], seems closely related to what we need. This technique tries to make the gap between two regions as big as possible and then the linear separator of these regions can be used as an angle to apply CORE-9. The boundary points of regions are enough for MM technique to find the best linear separator.

The identified angle is particularly useful when two regions are very close but still separate. In many of these cases, CORE-9 would not be able to detect that two regions are disconnected, but as Figure 3 shows, the angle found by MM is successful. We performed a number of tests that showed MM finds good angles against other directions (see Section 6).

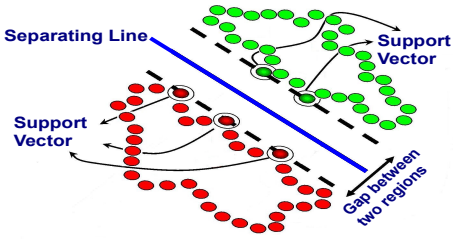
After finding good directions, we rotate the set of boundary pixels about each of these directions and try to find the spatial information between regions.

4 Individual Aspects

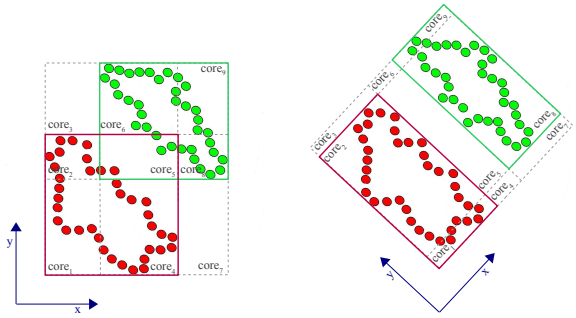
In this section, we analyse how multiple angles can be used to obtain more accurate spatial information. Specifically, we show that we can extract more accurate topology, size, distance and direction information between two objects in video than what is possible with standard CORE-9.

For each α, CORE$_\alpha$-9 allows us to infer an approximation of the actual spatial relation between two objects a and b. Cohn et al. [2012] showed how we can get topological relations, size, distance and direction relations, and changes of these relations over time from CORE-9. It is clear that their method works for any fixed angle α. If we take multiple angles, then we could possible get multiple different approximations of the real spatial relations between two objects a and b.

We say that A_α and B_α are the bounding boxes of objects



(a) The direction found by MM technique



(b) CORE-9 (c) Angled CORE-9

Figure 3: Using MM to find the best linear separator between two regions. (a) the separator found by MM technique, (b) the bounding boxes used in CORE-9, (c) the bounding boxes of objects along the direction found by the MM technique

a and b when using angle α. The spatial relations between A_α and B_α are denoted as θ_α for topology, σ_α for size, δ_α for direction, and Δ_α for distance. If we have multiple angles $\alpha_1, \ldots, \alpha_k$, then the union of the different spatial relations we might obtain for rectangles with these angles (but for the same objects a, b in the same video frame) is denoted as $\theta = \{\theta_{\alpha_1}, \ldots, \theta_{\alpha_k}\}$ and the same for σ, δ, and Δ.

4.1 Topology

The topological relations we use are mainly RCC-8 relations [Randell *et al.*, 1992]. RCC-8 consists of the eight basic relations DC (disconnected), EC (externally connected), PO (partial overlapped), EQ (equal), TPP (tangential proper part), NTPP (non-tangential proper part), and their converse relations TPPi and NTPPi. These relations are jointly exhaustive and pairwise disjoint, i.e., between any two objects exactly one of these eight relations holds. If it is not sure which one holds, we can express this as a union of possible basic relations. This gives us $2^8 = 256$ possible RCC-8 relations. Which relation holds between two rectangles A_α and B_α can be computed as given in [Cohn *et al.*, 2012]. How good an approximation this relation is to the actual RCC-8 relation between a and b largely depends on overlaps of parts of rectangles that are not part of the actual region. By finding tighter bounding boxes, we should be able to get a better approximation.

In the following, we develop an algorithm that allows us to integrate different RCC-8 relations between rectangles with different angles α. Figure 4 shows an example of how integrating the topological relations of rectangles with different

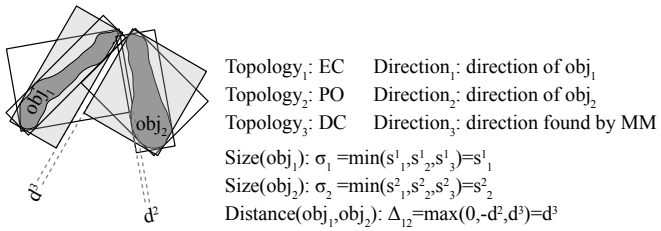


Figure 4: AngledCORE-9 and extracted topology, size and distance using three different angles

angles allows us to obtain a better approximation of the real topological relation between a and b. We analysed all possible combination of bounding boxes and their topological relations θ. The number of possible states with two regions is given by $\binom{p+r-1}{r}$, where $p = 8$ (8 basic RCC-8 relations) and r is the number of angles. For three different angles, there is a total of 120 different possible combinations. Enumerating all combinations as a table is too big to include in this paper, but the combinations can be encoded efficiently by an algorithm that allows us to compute the possible RCC-8 relations between two objects given different angles (see Algorithm 1).

One interesting point about using CORE-9 with multiple angles is that sometimes the topological relation we obtain is different from those of each individual angle. For example, in lines 17–18 of the algorithm, we start with the RCC-8 relations NTPP, NTPP and EQ, but the actual RCC-8 relation can only be PO or TPP. This is a clear indication that combining more angles gives us considerably more accurate information than using only one angle.

If two regions are in relation EC, there exists a supporting hyperplane which can be pictured as a straight line in 2-dimensional space. Having relation EC in two different angles means we have two intersecting supporting hyperplanes and the two regions can be placed only on two opposite spaces created by these two lines. In this case, these two regions can be DC or EC in reality. The only possible point where they can meet each other is the crossing point of two hyperplanes. So, if the third angle also results in EC (Algorithm 1, lines 4–5), the corresponding hyperplane can only passes through the same intersection point and because both regions should be tangent to this hyperplane when EC holds, both regions must be externally connected in reality.

As another example, suppose in two angles we have EC and EQ as topological relations (Algorithm 1, lines 4–5). EQ needs to have at least one boundary point of each object on each side of the bounding boxes. On the other hand, EC needs to have regions in two different spaces created by the corresponding hyperplane. If we have both EC and EQ, the only possible situation is when the EC hyperplane is crossing two opposite corners of the EQ regions' bounding box. Then two regions should be tangent in both corners. Thus, the real topological relation between these two regions must be EC.

Some of the relations in RCC-8 are symmetric, like PO, EQ, EC and DC. Therefore, many combinations have the same result. For example the output for NTPP, NTPP, EQ and for NTPPi, NTPPi, EQ is the same.

Algorithm 1 Integrating RCC-8 relations for MBBs with three different angles

1: **if** $DC \in \theta$ **then**
2: $\quad topology = \{DC\}$
3: **else if** $EC \in \theta$ **then**
4: $\quad$ **if** $(\theta = \{EC\}) \vee (EQ \in \theta)$ **then**
5: $\quad\quad topology = \{EC\}$
6: $\quad$ **else**
7: $\quad\quad topology = \{EC, DC\}$
8: $\quad$ **end if**
9: **else if** $\theta = \{NTPPi\}$ **then**
10: $\quad topology = \{EC, DC, PO, TPPi, NTPPi\}$
11: **else if** $\theta = \{NTPP\}$ **then**
12: $\quad topology = \{EC, DC, PO, TPP, NTPP\}$
13: **else if** $(\theta = \{TPPi\}) \vee (\theta = \{TPPi, NTPPi\})$ **then**
14: $\quad topology = \{EC, DC, PO, TPPi\}$
15: **else if** $(\theta = \{TPP\}) \vee (\theta = \{TPP, NTPP\})$ **then**
16: $\quad topology = \{EC, DC, PO, TPP\}$
17: **else if** $(EQ \in \theta) \wedge (\theta \cap \{TPP, NTPP\} \neq \emptyset)$ **then**
18: $\quad topology = \{PO, TPP\}$
19: **else if** $(EQ \in \theta) \wedge (\theta \cap \{TPPi, NTPPi\} \neq \emptyset)$ **then**
20: $\quad topology = \{PO, TPPi\}$
21: **else if** $\theta = \{EQ\}$ **then**
22: $\quad topology = \{PO, TPP, EQ\}$
23: **else if** $(\theta \setminus \{TPP, NTPP\} = \{PO\}) \vee (\theta \setminus \{TPPi, NTPPi\} = \{PO\})$ **then**
24: $\quad topology = \{EC, DC, PO\}$
25: **else**
26: $\quad topology = \{PO\}$
27: **end if**

4.2 Size and Distance

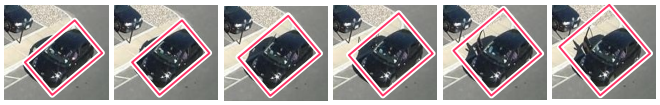
Other aspects of space that are very interesting for video analysis, are size and distance. Changing size allows us to infer if objects are moving closer or away from the camera, but also if objects are changing shape, for example, if a car door opens (see Figure 5).

Relative size and distance is important for determining the changing interaction between two entities. Similar to topology, we can also obtain a better approximation of the actual size and distance relationships of objects by considering multiple angles as compared to only one angle. Each MBB must be larger than the object, so the smallest MBB (in terms of length times width) we obtain is obviously the best approximation of size. The more angles we use, the higher the chance of finding the tightest and smallest MBB.

When comparing the relative size of two objects a and b, we can compare the smallest bounding box we get for a with the smallest bounding box we get for b independent of their respective angles.

For external distances between objects, but also for the internal distances (for example how much are two objects overlapping, or how far away from the opposite edge is an object contained in another object), the maximum distance over all angles is most important. By changing angles, it is always possible to bring to MBBs closer to each other, but there is a unique maximal distance.

(a) No changes when using horizontal rectangles



(b) Changes when using the angled rectangles

Figure 5: Difference between standard and angled MBBs

4.3 Direction

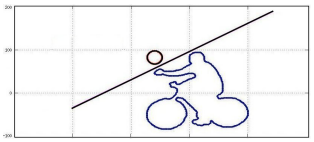
While direction is a very important spatial concept, direction is essentially a ternary relation where the direction between two objects is relative to a reference direction. This can be a global direction, relative to an observer, or relative to the inherent direction of one of the objects. For standard CORE-9, we use a global direction which is fixed by the video frame. For multi-angled CORE-9 we deliberately give up this global reference direction. Therefore, the directions we can specify now are with respect to the angle α that is used for a particular representation CORE$_\alpha$-9. Obviously this does not allow us to obtain any meaningful integration of information as it does for the other spatial relations, but we obtain something that could be equally valuable: relative direction information.

Relative direction is defined with respect to the inherent orientation of an observer. As described in Section 3.1, we compute the spread of each object and extract an angle from this that we can use for AngledCORE-9. If an object moves or rotates over time, then its spread changes accordingly. There are many objects where the spread corresponds to its inherent direction, e.g. the car in Figure 2. Therefore, the angle we obtain from the spread could be regarded as the angle that defines relative directions for this object. If we always use the spread of each object as an angle α for AngledCORE-9, than we can track how the relative direction of other objects change with respect to our reference object over time. So we can keep track of relative direction over time, despite changing objects and changing angles. This seems much more useful than tracking changes of absolute direction, as we do for standard CORE-9.

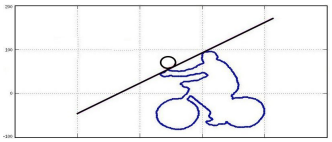
5 Action Classification

To compare the efficiency of AngledCORE-9 against CORE-9, we conduct experiments on action recognition. We extract spatial information between objects in the video. Then we use them as input features to an unsupervised clustering method.

One unsupervised probabilistic model for topic modeling is Latent Dirichlet Allocation (LDA) [Blei *et al.*, 2003] which is useful for automatically clustering collections of discrete data. This learning algorithm which was first presented for text to group words into topics, recently has been used in the computer vision field, like action recognition [Carlos *et al.*, 2008], classification [Fei-Fei and Perona, 2005] and image segmentation [Sokeh and Gould, 2012].



(a) Topology by CORE-9: TPP, by CORE$^9_{\alpha_{MM}}$: DC



(b) Topology by CORE-9: NTPP, by CORE$^9_{\alpha_{MM}}$: EC

Figure 6: Difference between CORE-9 and Angled CORE-9 in finding the topological relation between two objects.

Briefly, we treat each video as a document and the whole dataset as the corpus. Video frames containing two interacting objects get treated as a single word for that document (video). We generally use four spatial features including topology, size, distance and direction. So for frame t of video v which includes two interacting objects a and b, we will have four-dimensional words like $W_{vt} = [\theta(a,b), \sigma(a,b), \delta(a,b), \Delta(a,b)]$. The quantization is done by k-means algorithm to discretise these four-dimensional values for each word. Associated with each video, there is a latent variable z_k which represents its cluster. Our goal is then to cluster similar activities together.

6 Experimental Results

To evaluate the proposed method for extracting various spatio-temporal aspects of activities, we ran experiments on short video sequences of five different actions including *approach*, *carry*, *catch*, *collide* and *drop*. Experiments were carried out using 50 videos from the Mind's Eye video dataset.[2] We sampled every 10th frame in the videos and manually labeled the objects in the scene.

First we evaluated the accuracy of AngledCORE-9 method for computing topology. Specifically, we were interested in determining whether two objects are very close or touching. We compared AngledCORE-9 using the MM method, CORE$^9_{\alpha_{MM}}$, for calculating bounding box angle with standard CORE-9 (using axis-aligned bounding boxes). We chose 15 different frames that included two close or overlapping objects. CORE$^9_{\alpha_{MM}}$ correctly determined the correct topology 87% of the time compared with 47% for CORE-9.

This experiment demonstrates that the MM method provides a significantly better direction for determining topological relationships. The reason can be understood by examining Figure 6, which shows two different scenarios involving the same two objects. In the first scenario (Figure 6a) the two objects are close but not touching, whereas in the second scenario (Figure 6b) they are actually touching. Only CORE$^9_{\alpha_{MM}}$ is able to correctly distinguish between the two scenarios.

Next we experimented with clustering activities based on the four spatial features described above. We extracted three angled bounding boxes for AngledCORE-9 using the PCA and MM methods applied to the boundary pixels of each object. We then used an implementation of LDA with Gibbs

[2] Videos available from `http://www.visint.org`.

Sampling [Phan and Nguyen, 2007] to cluster the videos into K different topics. In our experiments we set the number of topics K to 5 (and set the LDA parameters α and β to 10 and 0.01, respectively).

We evaluate the quality of topics found by LDA for each video. Table 1 shows the results of clustering videos by topic. Here we also compute the purity of each topic, which measures the fraction of the majority occurring action within a cluster to the total size of the cluster.

Table 1: Clustering results.

	topic 1	topic 2	topic 3	topic 4	topic 5
approach	0	2	0	2	**6**
carry	**8**	1	1	0	0
catch	1	2	**6**	0	1
collide	1	**5**	1	0	3
drop	2	0	1	**7**	0
purity	0.8	0.5	0.6	0.7	0.6

To evaluate classification accuracy we assign a unique action label to each topic. Here we use as the label the majority occurring action within the cluster. We compute precision and recall and also F1-measure as their combination for each class. Recall is defined as the fraction of the majority occurring action within a cluster to the whole number of the same label in the clustering result. Results are shown in Table 2 using the spatial features extracted extracted by AngledCORE-9 ($\text{CORE}^9_{\alpha_{MM},\alpha_a,\alpha_b}$).

Table 2: Quantitative evaluation for action classification.

action	Precision	Recall	F1-measure
approach	0.60	0.60	0.60
carry	0.80	0.67	0.73
catch	0.60	0.67	0.63
collide	0.50	0.50	0.50
drop	0.70	0.78	0.74

Figure 7 represents a comparison between action classification results with different angular directions: CORE-9 which uses the axis-aligned, $\text{CORE}^9_{0,\alpha_{MM}}$ using only two directions of horizontal and the one found by MM, $\text{CORE}^9_{0,\alpha_a,\alpha_b}$ using the frame direction and two PCA directions for objects a and b, and finally our method, $\text{CORE}^9_{\alpha_{MM},\alpha_a,\alpha_b}$, using the direction found by MM and the PCA directions for two objects.

Regarding the comparison, in most cases the MM technique improves the clustering results over both axis-aligned and PCA directions. The action *carry*, however, achieves better results with PCA directions. This shows that for the action *carry* size and direction, which are better captured by the PCA directions than the MM direction, is more important than topology alone. Adding the MM direction further improves results even for this action. Another interesting result is the action *drop*. Here the objects' MBBs are mostly axis-aligned so the original CORE-9 performs quite well. In all cases, however, our $\text{CORE}^9_{\alpha_{MM},\alpha_a,\alpha_b}$ performs the best due to the combination of extracted spatial information.

The results show that the directions used in our method

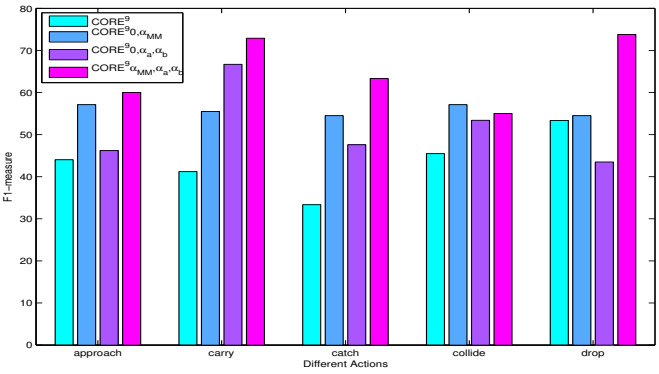


Figure 7: Quantitative comparison between CORE-9, $\text{CORE}^9_{0,\alpha_{MM}}$, $\text{CORE}^9_{0,\alpha_a,\alpha_b}$ and $\text{CORE}^9_{\alpha_{MM},\alpha_a,\alpha_b}$ for video classification on five different actions.

work better than other combinations. The MM technique helps to find the topological relation more precisely, so the results for our algorithm and $\text{CORE}^9_{0,\alpha_{MM}}$ are better than the two other methods. In comparison to CORE-9 in most cases $\text{CORE}^9_{0,\alpha_a,\alpha_b}$ classifies the actions more precisely, because the spread direction of objects gives the tightest bounding box which leads to a more accurate measure of size of objects.

7 Conclusion and Future Work

A good representation for modeling the behaviour of interacting objects over space and time is essential for automatic analysis and discovery of activities in videos. We focus on a qualitative representation that captures spatio-temporal aspects such as topology, size, distance and direction. By considering these as feature descriptors machine learning algorithms can by used to cluster or classify activities.

In this work, we proposed a new accurate model built on CORE-9 for extracting the aspects of space over time. We relaxed CORE-9 to allow arbitrary angles and thus obtain tighter bounding boxes around objects participating in activities. We chose three different directions to apply CORE-9 and showed the benefit of these directions. Specifically, the MM direction allows for better modeling of the topological relations, and the PCA directions provide better size and direction information. We used features derived from our AngledCORE-9 representation to describe video sequences, and quantitatively evaluated our model for action clustering against standard CORE-9.

One interesting trajectory for future work is to extend our representation to 3D space. By making use of current 2.5D imaging technologies, such as the Kinect sensor, we should be able to obtain an even better understanding of the relationship between objects. Here, in addition to points, lines and areas as the basic geometric entities, we would include the volume and depth making our representation significantly richer. Another interesting direction for future work is in applying our AngledCORE-9 model to noisy data obtained from object detectors or tracking algorithms. Such an approach would require an additional element in the spatio-temporal reasoning to deal with uncertainty.

References

[Blei *et al.*, 2003] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[Carlos *et al.*, 2008] Niebles Juan Carlos, Wang Hongcheng, and Fei-Fei Li. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision (IJCV)*, 79(3):299–318, 2008.

[Cohn and Renz, 2008] Anthony G. Cohn and Jochen Renz. *Qualitative Spatial Representation and Reasoning*, pages 551–596. F. van Hermelen, V. Lifschitz, B. Porter, eds., Handbook of Knowledge Representation, Elsevier, 2008.

[Cohn *et al.*, 2012] Anthony G. Cohn, Jochen Renz, and Muralikrishna Sridhar. Thinking inside the box: A comprehensive spatial representation for video analysis. In *The 13th International Conference on Principles of Knowledge Representation and Reasoning (KR)*, 2012.

[Cortes and Vapnik, 1995] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[Fei-Fei and Perona, 2005] Li. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. *Computer Vision and Pattern Recognition (CVPR)*, pages 524–531, 2005.

[Leal-Taixe, 2012] Laura Leal-Taixe. Branch-and-price global optimization for multi-view multi-target tracking. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1987–1994, 2012.

[Morariu and Davis, 2011] Vlad I. Morariu and Larry S. Davis. Multi-agent event recognition in structured scenarios. In *Computer Vision and Pattern Recognition (CVPR)*, 2011.

[Pearson, 1901] K. Pearson. *On Lines and Planes of Closest Fit to Systems of Points in Space*. University College, 1901.

[Phan and Nguyen, 2007] Xuan-Hieu Phan and Cam-Tu Nguyen. Gibbslda++: A c/c++ implementation of latent dirichlet allocation (lda), 2007.

[Randell *et al.*, 1992] David A. Randell, Zhan Cui, and Anthony G. Cohn. A spatial logic based on regions and connection. In *The 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR)*, pages 165–176, 1992.

[Renz and Mitra, 2004] Jochen Renz and Debasis Mitra. Qualitative direction calculi with arbitrary granularity. In *The 8th Pacific Rim International Conference on Artificial Intelligence*, pages 65–74, 2004.

[Sokeh and Gould, 2012] Hajar Sadeghi Sokeh and Stephen Gould. Towards unsupervised semantic segmentation of street scenes from motion cues. In *International Conference on Image and Vision Computing New Zealand (IVCNZ)*, 2012.

[Sridhar *et al.*, 2010] Muralikrishna Sridhar, Anthony G. Cohn, and David C. Hogg. Unsupervised learning of event classes from video. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2010.

[Sridhar *et al.*, 2011a] Muralikrishna Sridhar, Anthony G. Cohn, and David C. Hogg. Benchmarking qualitative spatial calculi for video activity analysis. In *IJCAI Workshop Benchmarks and Applications of Spatial Reasoning*, pages 15–20, 2011.

[Sridhar *et al.*, 2011b] Muralikrishna Sridhar, Anthony G. Cohn, and David C. Hogg. From video to rcc8: Exploiting a distance based semantics to stabilise the interpretation of mereotopological relations. In *Conference On Spatial Information Theory (COSIT)*, pages 110–125, 2011.

[Viola and Jones, 2004] Paul Viola and Michael Jones. Robust real-time face detection. *International Journal of Computer Vision (IJCV)*, 57(2):137–154, 2004.