

Efficient Image Denoising by MRF Approximation with Uniform-Sampled Multi-Spanning-Tree

Jun Sun

The Australian National University
National ICT Australia
Canberra, Australia
j.sun@anu.edu.au

Hongdong Li

The Australian National University
Canberra, Australia
hongdong.li@anu.edu.au

Xuming He

National ICT Australia
Canberra, Australia
xuming.he@nicta.com.au

Abstract—Traditionally, image processing based on Markov Random Field (MRF) is often addressed on a 4-connected grid graph defined on the image. This structure is not computationally efficient. In our work, we develop a multiple-trees structure to approximate the 4-connected grid. A set of spanning trees are generated by a new algorithm: re-weighted random walk (RWRW). This structure effectively covers the original grid and guarantees uniformly distributed occurrence of each edge. Exact maximum a posterior (MAP) inference is performed on each tree structure by dynamic programming and a median filter is chosen to merge the results together. As an important application, image denoising is used to validate our method. Experimentally, our algorithm provides better performance and higher computational efficiency than traditional methods (such as Loopy Belief Propagation) on a 4-connected MRF.

Keywords-MRF; image denoising; spanning tree; MAP inference;

I. INTRODUCTION

MRF is a popular tool in statistics inference, and has been widely used in image processing and computer vision [1]. While it is popular, how to balance the computational cost and visual effectiveness is still a challenging task. The most universal structure of MRF in image processing is the 4-connected grid, which makes the graph more general but not computational efficient for inference. To illustrate, graph cuts [2] and Loopy Belief Propagation (LBP) [3], as the most popular inference methods, are both iterative algorithms. Due to the poor computational efficiency, inference is slow. Our motivation for this paper is to build a multiple-trees structure, which is loop free and effectively approximates the 4-connected MRF.

There are three main contributions of our work: (i) We develop a new algorithm called re-weighted random walk (RWRW) to pick up a set of spanning trees of a 4-connected grid. This algorithm guarantees the occurrence of each edge in the original graph is uniformly distributed. (ii) We get exact MAP inference on each tree structure rapidly and use a computationally efficient but robust median filter to combine the results together. (iii) Experimentally, we demonstrate our RWRW spanning tree structure with

median filter outperform LBP in a 4-connected grid for image denoising.

The theory and application of using tree structures in image processing and computer vision are popular in recent years. Paper [4] provides a tree-reweighted (TRW) framework finding MAP based on message passing theory. Random spanning tree theory is used by [5] to approximate conditional random field. Our algorithm is different from all the previous work, as will be explained later.

In addition, our algorithm is distinct from those traditional work that use sparse representation for image processing, such as [6] and [7]. Papers referred above mainly focus on "learning atoms" and "unitary dictionary" to improve the results of image restoration. However, our work aims to build a combined sparse structure to approximate the 4-connected MRF and perform exact inference on it without any pre-process and learning work.

The rest of the paper is organized in the following way: We provide a statistical understanding to solve some image processing tasks and choose an estimator for inference in the next section. Section 3 provides the main procedures of RWRW algorithm to generate a set of spanning trees. In section 4 we analysis the method and computational cost of performing inference on the RWRW structure, as well as the merging strategy. We will show our experiment results and conclusion in section 5 and 6 respectively.

II. PROBLEM FORMULATION

A. Choosing an Estimator

Many image processing and computer vision tasks, such as image denoising, restoration and stereo are addressed by inference on an MRF. The joint probability of an MRF as:

$$P(x_1, x_2, \dots, x_N, y_1, y_2, \dots, y_N) = \prod_i \Psi(x_i, y_i) \prod_{(i,j)} \Psi(x_i, x_j), \quad (1)$$

where (i, j) is a pair of nodes in neighborhood, N is the number of nodes, y_n and x_n are the observed value and label value at location n respectively.

To inference on an MRF, we need to choose an estimator. The most universal one is MAP estimator. In MAP estimation, we label $x_1, x_2 \dots x_n$ to each node to maximize the Equation 1.

MAP estimation on Equation 1 is equivalent to minimizing an energy function in the following form:

$$E(X) = \sum_i D_i(x_i) + \sum_{(i,j)} S(x_i, x_j), \quad (2)$$

where $D(\cdot)$ and $S(\cdot)$ is data term and pairwise term respectively. $D(\cdot)$ is the cost of assigning a label to one node and $S(\cdot)$ penalize the different labels assigned in neighborhood nodes.

B. Relation to previous works

Our work is motivated but distinct from the TRW message passing framework, which uses a convex combination of distributions on tree structures to derive upper bounds on the cost of a MAP configuration. Tree structure is used only to determine the edge weight in the original grid. Inference is still performed on a 4-connected MRF, thus leading the same disadvantage as LBP and graph cuts.

In [5], researchers provide a Conditional Random Field (CRF) approximation work by spanning trees. However, they only use the algorithm cited in [8] to generate a set of trees. The original graph is not guaranteed to be completely covered by their method. As well, edge occurrence is not uniformly distributed. In addition, a CRF is discriminative, which is different from the generative MRF.

III. RANDOM SPANNING TREE GENERATION USING RE-WEIGHTED RANDOM WALK

In this section, we mainly focus on how to use a limited number of spanning trees to approximate the 4-connected MRF. As well, we guarantee the occurrence of each edge is uniformly distributed.

A. Model Definition

Let $G(V, E)$ denote the 4-connected grid, where $V = (1, 2, \dots, n)$ is the collection of nodes and $E(i, j)$ describes the pairwise relationship between two nodes in neighborhood.

Suppose we use a total number of Ω trees to approximate the original graph and ω represent the number of trees that has been constructed. Occurrence number of each edge is different among the set of spanning trees, ξ denotes edges occurrence for each edge.

The two most important attributes of the RWRW spanning trees structure is: (i) The collection of trees should cover the whole image, which means ξ for each edge must larger than or at least equal to one. (ii) The total occurrence of each edge in the original image is a uniform distribution. Fig. 1 provides an example of our proposed RWRW spanning trees structure for a 3×3 grid graph.

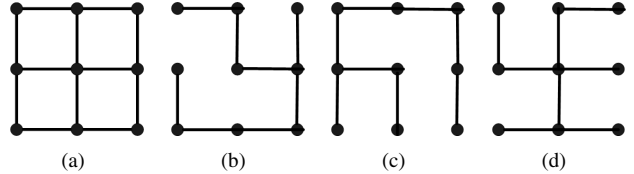


Figure 1. RWRW spanning trees: Tree (b), (c) and (d) are sub-graph of grid (a) and they contains all the possible edges in (a). The frequency of occurrence of each edge is $\xi = 2$ and occurrence probabilities is $\epsilon = 2/3$.

B. Tree Construction Using Re-weighted Random Walk

First of all, we pick up our first spanning tree randomly using Wilson’s method [8]. One arbitrary node r is chosen as the root and also marked as current tree τ at the beginning. Then an arbitrary node i is chosen as the start point, from which we do a loop-erasing random walk until it hit the current tree τ . Once any node encounters the current tree, all the edges are marked from node i to the current tree and all the nodes in this routine are added into the current tree structure. The tree is completed when all the nodes are marked “in tree” by a random walk.

Once the first tree is obtained, we do a depth-first search to get the relationship between each pair of parent and child. We add 1 to $\xi_{(i,j)}$ when there is one edge detected between node i and j . The collection of edges that exist in the first tree is defined $e_{(i,j)}$ and otherwise $\bar{e}_{(i,j)}$. As defined in the previous step, ω is the number of trees that has been constructed. We define $W_{(i,j)}$ as the weight of edge $E_{(i,j)}$:

$$W(i, j) = \xi_{(i,j)} / \omega, \quad (3)$$

$W(i, j)$ is used as the basis of building the next spanning tree.

Secondly, we use a iterative method to make the 4-connected grid covered by a set of trees and guarantees a uniform distribution of edge occurrence. At each iteration, another arbitrary node \tilde{r} is chosen as the root and marked as the current tree $\tilde{\tau}$. For any node i , we run the re-weighted walk to make it hit the current tree.

The most important procedure is how to choose the walking direction at each node. Due to the loop free nature of a tree, about half of edges in the original image are discarded in the first step, we aim to use our walk schedule to recover the missing items $\bar{e}_{(i,j)}$. At each node, there are four directions to select as the next target. $W(i, j)$ varies from each direction. Our “walk” schedule gives preference to direction having the minimum weight at each node. Once a cycle is formed by walking through one direction, we drop it and choose another direction with a second smallest weight.

Let $P_i^{\omega+1}$ denote the walking path from any beginning node towards i at tree $\omega + 1$, the edge selected in the next step is expressed as:

$$E_{(i,j)}^{\omega+1} := \arg \min W_{(i,j)}, j \notin P_i^{\omega+1} \quad (4)$$

Once it touches the current tree $\tilde{\tau}$, the “walk” finishes and all nodes in this path are added into the current tree structure.

After several iterations, all the edges in the original grid have been covered. Compared to the previous work [5] we take the preference of minimum weight into account. The most significant advantage compared is the uniformly distributed occurrence of each edge. Figure 2 (in experiment part) provides an example result of the advantage of our algorithm.

IV. DYNAMIC PROGRAMMING ON TREE AND MERGING STRATEGY

A. Dynamic Programming on Tree

Tree dynamic programming is an efficient algorithm to estimate MAP on a tree structure. It is used by [9] for object detection and improved by [10]. According to our energy formulation, each node i (except for the root) passes a minimum energy to its parent j in this form (C_i is the collection of children of i):

$$E_i(x_j) = \min(D_i(x_i) + S(x_i, x_j) + \sum_{w \in C_i} E_w(x_i)) \quad (5)$$

Energy is summed at the root then a label x_r is decided by the minimum energy. Labels of the other nodes are recursively decided by their parents.

The computational complexity of dynamic programming on a tree is linear with the number of nodes and quadratic with the number of labels, totally $O(NL^2)$. (L is the length of label space). This cost can be optimized to $O(NL)$ by the distance transform strategy used in the smoothness penalty computation referred in [11].

B. Merging Strategy

Inference result on single spanning tree is limited for the whole 4-connected grid, losing much pairwise information. Therefore, it is necessary to find a applicable strategy to combine those results obtained from each spanning tree together. According to our algorithm, a median filter is chosen as the merging strategy.

Median filter: Let $P(x|y, \tau)$ be defined as a posteriori probability obtained by exact inference on tree τ , $\tau \in T$.

$$x^{MAP}(\tau) := \arg \max P(x|y, \tau) \quad (6)$$

The merging strategy works by taking the median value of $x^{MAP}(\tau)$ for the observed value y :

$$x^*|y = \text{Median}\{x_i, x_i \in x^{MAP}|y\} \quad (7)$$

Notice that our inference strategy provides more robust results and have better edge-preserving property than ordinary Loopy BP. The robustness mainly comes from the sparse yet uniform edge distribution and the median operator. Specifically, sparse connections in the chosen trees leads to less smoothing effect on average, and for those pixels nearby object boundaries, the median operation guarantees

that the smoothing effect would come from the same side of boundaries with high probability. Compared with another voting method referred in [5], which takes $x^*|y$ as the one that has the most frequent occurrence in collection $x^{MAP}|y$. The advantage of median filter provides a more robust results experimentally. For those pixels in the interior of regions, the uniform edge property would still give similar isotropic smooth effect as the original random field. We will compare the results between merging strategy in [5] and our median filter in the experimentation part.

Considering the computing complexity and performance, the median operator is simple but very effective. At the meantime, based on the ensemble learning framework [12], we can choose other complicated methods, such as adaptive weighting and bagging instead. Using those trainable algorithms, we can analysis the importance of each spanning tree and develop a more complicated and stronger combination method.

V. EXPERIMENTAL RESULTS

A. Edges Distribution

The re-weighted random walk aims to cover a 4-connected MRF in a few iterations and provides a uniform distribution for each edge. First of all, we compare the edge distribution of our algorithm with Wilson’s method used in [5].

As shown in Fig. 2 and Fig. 3, after 20 iterations, all the edges in original grid have been covered by RWRW and most importantly edges distribution center around 10. Compared with ours, Wilson’s algorithm not only fail to cover the whole image, but provides a quite nonuniform distribution of edges.

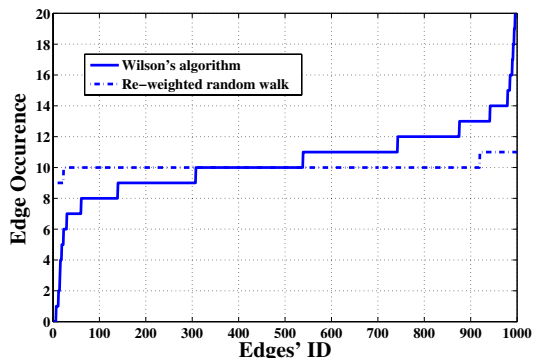


Figure 2. Edges occurrence after 20 iterations for both Wilson’s algorithm and re-weighted random walk. Edges are sampled from a 300×400 grid (For better visual convenient, edges’ ID is sorted by edges occurrence). Our algorithm well covers the original graph and provides a uniformly distributed edges occurrence.

B. Image Denoising

We choose image denoising as our task to test the proposed RWRW spanning trees structure. According to the inference method, we first define our data term and pairwise

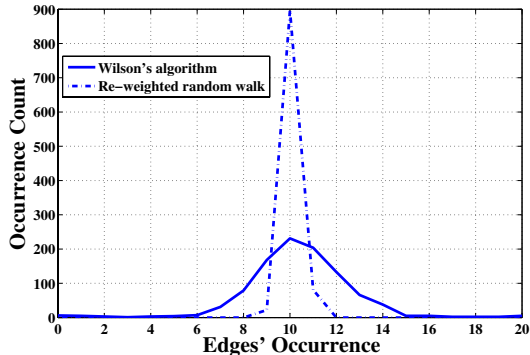


Figure 3. Most of the edges occurrence ranges from 9 to 11 times after 20 iterations by the RWRW algorithm, compared with that of Wilson’s from 6 to 15 times.

term of energy function. We use truncated L_1 norm for both the data term and pairwise term:

$$D_i(x_i) = \min(|\tilde{I}_i - x_i|, t) \quad (8)$$

$$S(x_i, x_j) = \min(\lambda|x_i - x_j|, \gamma) \quad (9)$$

\tilde{I}_i is the pixel i ’s intensity in the noisy image, λ is a scale factor used to represent the relative importance of data term and pairwise term. t and γ are truncated values, which are chosen variably from different images.

Table 1
COMPARISON OF DIFFERENT ALGORITHMS

		<i>Penguin</i>	<i>Castle</i>	<i>Mushroom</i>
PSNR (dB)	RWRW+median	30.33	29.18	29.73
	RWRW+majority voting	29.98	28.78	29.14
	Accelerated LBP	24.26	24.06	25.34
Timing	RWRW+median	2s	8s	8s
	RWRW+majority voting	2s	8s	8s
	Accelerated LBP	17s	70s	65s

We compared our algorithm with accelerated loopy belief propagation both on accuracy and timing, image data is chosen from [1] and [13]. Peak signal-to-noise ratio ($PSNR = 20 \log_{10}(255/RMSE)$) is used to evaluate denoising results. RMSE is root mean square errors, defined as:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (I_i - \hat{I}_i)^2} \quad (10)$$

I_i and \hat{I}_i is intensity of node i for original and restored image respectively.

According to our experiment, the results of proposed method on RWRW spanning trees structure outperforms LBP on a 4-connected MRF both in accuracy and computational cost. We test different images with different level of Gaussian noise ($\sigma = 20$ and $\sigma = 25$), the results are shown in Fig. 4 and Fig. 5 respectively.

Compared with LBP on a 4-connected MRF, our algorithm not only effectively removes the noise, but also well keeps details of the original image. However, images restored by LBP are oversmooth, some details around the edges are missing. The oversmooth results is mainly due to the message passing rules with the 4-connected structure. Pixels tend to be consistent with their neighbors which lead to a partially fuzzy result and also a high root mean square error.

For our proposed algorithm, generating RWRW spanning trees structure costs the largest component of time; despite that, it is still much fast than accelerated LBP with distance transform. To balance the the computational cost and denoising performance, the number of spanning trees are suggested between 10 and 20, since we experimentally find there is only slightly improvement although we use a large number of trees. However, if the input image size is known in advance, we can just generate a set of spanning trees by our algorithm and fix it before denoising. The computational complexity for each inference is $O(NL^2)$ (L is the length of label space). In addition, in our experiment, we choose 15 spanning trees to do the MAP inference. We have compared the running time in table 1, the timing for our proposed method does not include the tree building time.

We also compare the denoising performance between our merging strategy and the method referred in [5]. As shown in table 1, using the same multi-spanning-tree structure, our median filter performs better than the majority voting in [5].

Notice that our method have addressed the image denoising problem in a distinct way from some classical methods, such as [14]. Instead of focusing on kernel based work or training methods [6], we mainly demonstrate that our RWRW spanning trees structure with a median filter performs good in approximating the traditional 4-connected grid in image processing tasks.

VI. CONCLUSION

In this paper, we have proposed a new algorithm to generate a RWRW spanning trees structure to approximate the 4-connected MRF. This structure is easy to understand and to code. This RWRW spanning trees structure jointly covers all the edges in the original 4-connected MRF in a few iterations and provides a uniform distribution of each edge. In the future, we will focus on the ensemble learning framework and analyses other strategies to combine inference result obtained at individual tree.

REFERENCES

- [1] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, “A comparative study of energy minimization methods for markov random fields with smoothness-based priors,” *IEEE Transactions on PAMI*, pp. 1068–1080, 2007.

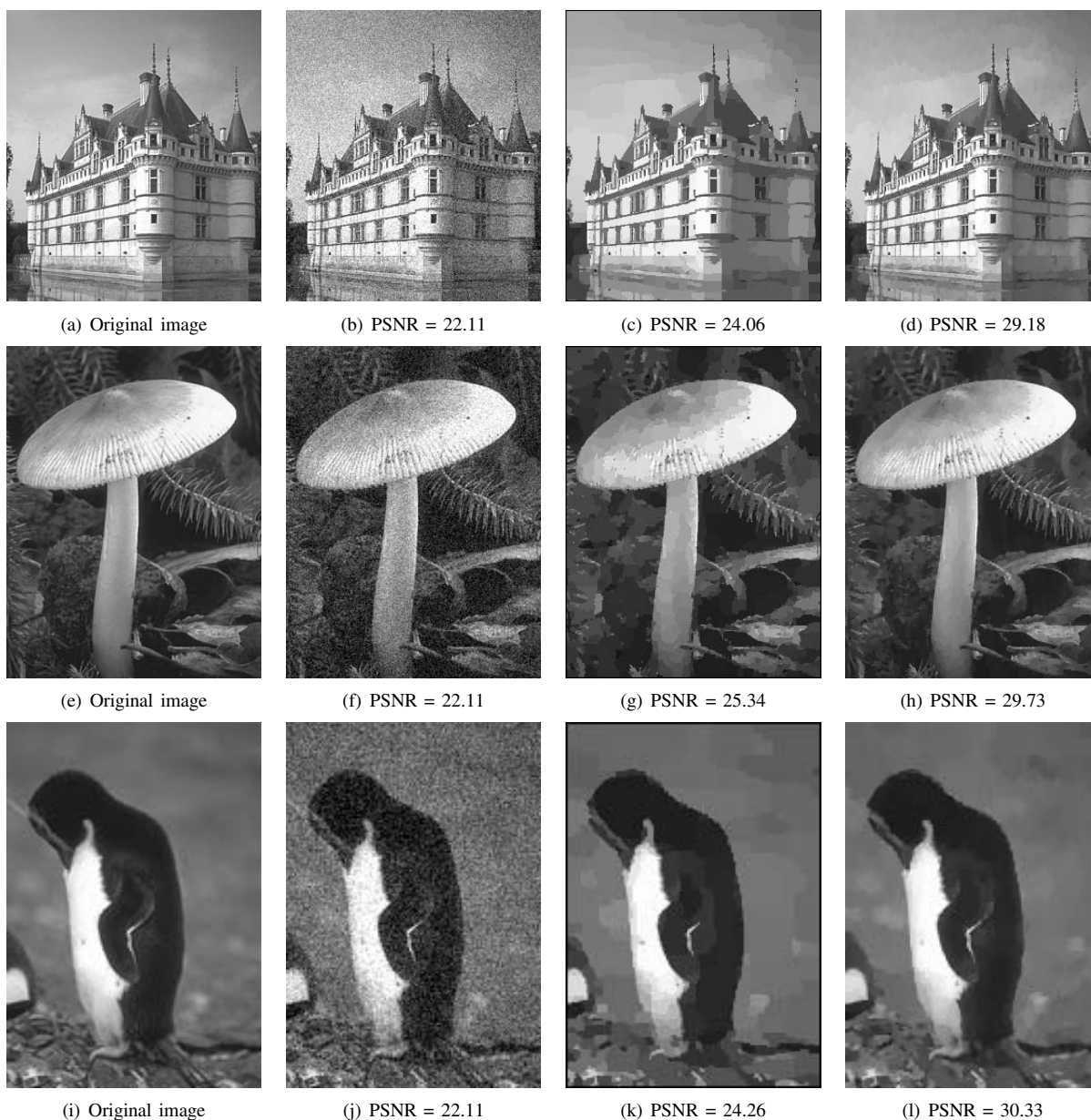


Figure 4. Denoising results in different images : The first and second collums are original image and noisy image with Gaussian noise ($\sigma = 20$) respectively, the third collum is the restored image by LBP and the fourth column is results generated by proposed method

- [2] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on PAMI*, vol. 23, no. 11, pp. 1222–1239, 2002.
- [3] J. Yedidia, W. Freeman, and Y. Weiss, "Generalized belief propagation," *NIPS*, pp. 689–695, 2001.
- [4] M. Wainwright, T. Jaakkola, and A. Willsky, "Tree-reweighted belief propagation algorithms and approximate ML estimation by pseudomoment matching," in *Workshop on AISTATS*, vol. 21, 2003.
- [5] P. Pletscher, C. Ong, and J. Buhmann, "Spanning tree approximations for conditional random fields," in *AISTATS*, vol. 5, 2009.
- [6] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Transactions on Image Processing*, vol. 17, no. 1, pp. 53–69, 2008.
- [7] M. Protter, I. Yavneh, and M. Elad, "Closed-form MMSE estimation for signal denoising under sparse representation modeling over a unitary dictionary," *IEEE Transactions on Signal Processing*, vol. 58, no. 7, pp. 3471–3484, 2010.
- [8] D. Wilson, "Generating random spanning trees more quickly than the cover time," in *STOC*, 1996, pp. 296–303.

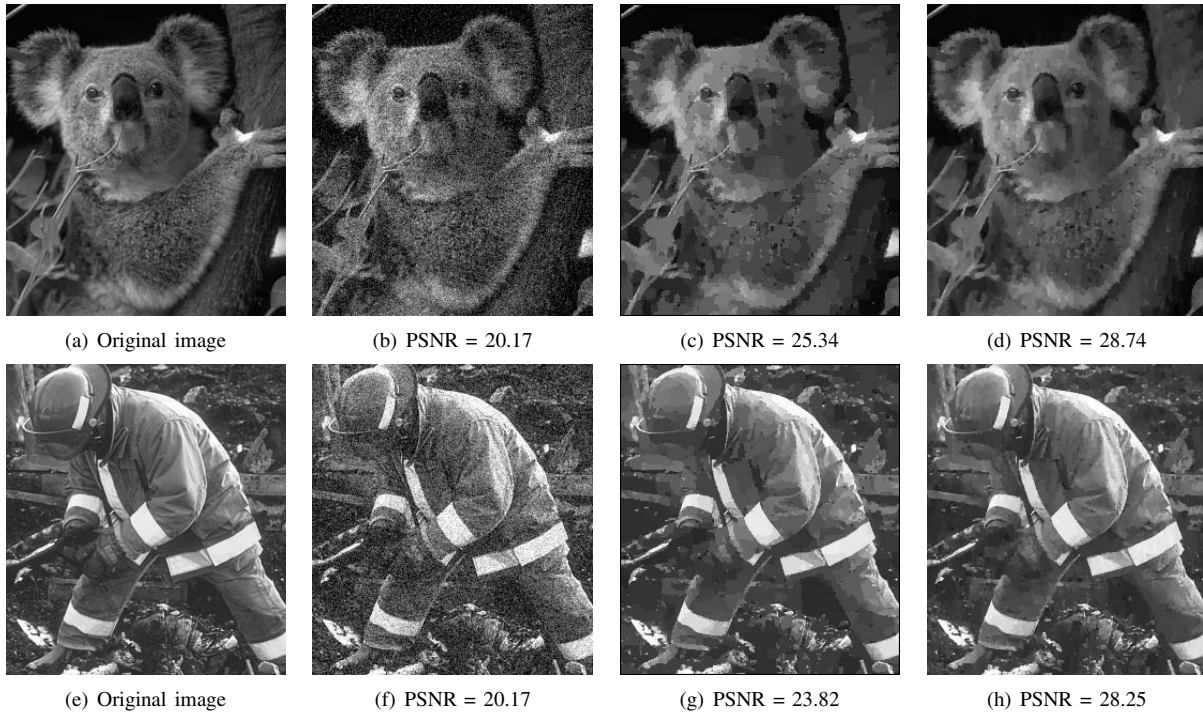


Figure 5. Denoising results in different images : The first and second collums are orginal image and noisy image with Gaussian noise ($\sigma = 25$) respectively, the third collum is the restored image by LBP and the fourth column is results generated by proposed method

- [9] P. Felzenszwalb and D. Huttenlocher, "Pictorial structures for object recognition," *IJCV*, vol. 61, no. 1, pp. 55–79, 2005.
- [10] O. Veksler, "Stereo correspondence by dynamic programming on a tree," in *CVPR*, vol. 2, 2005, pp. 384–390.
- [11] P. Felzenszwalb and D. Huttenlocher, "Efficient belief propagation for early vision," *IJCV*, vol. 70, no. 1, pp. 41–54, 2006.
- [12] T. Dietterichl, "Ensemble learning," *The Handbook of Brain Theory and Neural Networks, Second edition*, pp. 3697–3717, 2002.
- [13] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *ICCV*, vol. 2, 2001, pp. 416–423.
- [14] J. Portilla, V. Strela, M. Wainwright, and E. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain," *IEEE Transactions on Image Processing*, vol. 12, no. 11, pp. 1338–1351, 2003.