Pattern Recognition 43 (2010) 1826–1834

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Bisection approach for pixel labelling problem

Dengfeng Chai [a,b,*], Hongwei Lin [b], Qunsheng Peng [b]

[a] Institute of Spatial Information Technique, Zhejiang University, No. 38, Zheda Road, Hangzhou, Zhejiang 310027, China
[b] State Key Lab of CAD&CG, Zhejiang University, No. 38, Zheda Road, Hangzhou, Zhejiang 310027, China

ARTICLE INFO

Article history:
Received 3 December 2007
Received in revised form
11 August 2009
Accepted 1 November 2009

Keywords:
Pixel labelling
Markov random fields
Graph cut
Stereo

ABSTRACT

This paper formulates pixel labelling as a series of two-category classification. Unlike existing techniques, which assign a determinate label to each pixel, we assign a label set to each pixel and shrink the label set step by step. Determinate labelling is achieved within $\log_2 n$ (n is size of label set) steps. In each step, we bisect the label set into two subsets and discard the one with higher cost of assigning it to the pixel. Simultaneous labelling of an image is carried out by minimizing an energy function that can be minimized via graph cut algorithm. Based on the bisection approach, we propose a bitwise algorithm for pixel labelling, which set one bit of each pixel's label in each step. We apply the proposed algorithm to stereo matching and image restoration. Experimental results demonstrate that both good performance and high efficiency are achieved.

1. Introduction

Many early vision tasks require assigning labels to pixels based on the observed images. The labels can denote quantities such as gray, disparity and so on. The label fields can be elegantly expressed as Markov random fields (MRFs). Then, the pixel labelling problem can be formulated as maximum *a posterior* estimation of the Markov random fields (MAP-MRF) in a Bayesian framework, it results in an energy minimization problem [1,2].

Simulated annealing is easy to implement and can optimize an arbitrary energy function. Theoretically, if annealing's temperature parameter is sufficiently low, it should eventually find the global minimum. Nevertheless, it gives results far from global minimum [3]. Moreover, it requires exponential time and it is very slow. Iterated conditional modes (ICM) adopts a greedy technique to find a local minimum [4]. It starts with an initial labelling and updates the labelling until a minimum is reached. Since only one pixel can change its label at each step, the results are extremely sensitive to the initial labelling and their quality is usually low.

Recently, people have developed graph cut for energy minimization. Graph cut methods [5–7] construct a weighted graph such that the minimum cut corresponds to a configuration minimizing the energy function. The minimum cut can be found efficiently by max flow algorithms such as the "push-relabel" [8]. When there are two labels involved, the global minimum can be found by a single minimum cut computation [3]. Usually, there are more than two labels involved, for example, image gray and disparity have many levels. Then, it is necessary to solve a multiway minimum cut problem. Unfortunately, multiway minimum cut problem is NP-hard [5].

Boykov et al. proposed $\alpha-\beta-$swap and $\alpha-$expansion algorithm to compute an approximate optimal solution [9]. These algorithms start with an initial labelling and update it iteratively. In each iteration, they perform a swap move for every pair of labels and an expansion move for every label, respectively, to minimize the energy. There are only two possible labels involved in the moves, and the optimal moves are found via standard minimum cut algorithm. Since a large number of pixels are allowed to change their labels simultaneously, they find local minimum with respect to very large moves and produce results with high quality [10]. The complexities of $\alpha-\beta-$swap algorithm and $\alpha-$expansion algorithm are $O(k*n^2)$ and $O(k*n)$, respectively, where k is the number of iterations and n is the size of the label set.

Moreover, people have developed belief propagation algorithms for energy minimization. Belief propagation algorithms [11,12] adopt a message propagating mechanism to assist pixel labelling. Each pixel receives messages from neighboring pixels. The message propagation is iterated until its convergence. Each pixel accepts the label that supports its maximal belief. Since the graphical model for pixel labelling consists of many loops, the belief propagation algorithm can eventually find an approximate solution. Belief propagation algorithms produce results with comparable quality as that of graph cut algorithms [10,13,14]. The complexity of the algorithm is $O(k*p*n^2)$, where p is the number of pixels in the image. The complexities of belief propagation and graph cut algorithms are either linear or

* Corresponding author. Tel.: +86 571 88206681; fax: +86 571 88206680.
E-mail addresses: chaidf@cad.zju.edu.cn (D. Chai), hwlin@cad.zju.edu.cn (H. Lin), peng@cad.zju.edu.cn (Q. Peng).

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.11.002

quadric in n, they are not efficient enough to support realtime applications.

This paper proposes an efficient method for pixel labelling. We treat labels as indicators of categories, i.e. pixels with same label belong to same category while pixels with different labels belong to different categories. Then, we formulate the pixel labelling problem as a classification problem, and classify pixels by a series of two-category classification. In other words, a label set instead of a determinate label is assigned to each pixel and it is shrunk step by step until the label set consists of only one label. It is not necessary to test every label or every pair of labels as that of $\alpha-\beta-$swap algorithm and $\alpha-$expansion algorithm. Using bisection technique, determinate label can be achieved within $\log_2 n$ steps. As a result, the whole process consists only $\log_2 n$ steps, and the complexity is reduced to $O(\log_2 n)$.

This rest of this paper is organized as follows. Section 2 formulates pixel labelling problem as a classification problem and Section 3 shows how to solve it using graph cut algorithm. A bitwise algorithm for pixel labelling is proposed in Section 4. Experimental results are presented in Section 5 and conclusions are drawn in Section 6.

2. Pixel labelling via classification

2.1. Pixel labelling problem

A MRF is a set of random variables $F=\{F_p, p\in P\}$ adhering to a field of pixels $P=\{p_1,\ldots,p_m\}$; each random variable F_p can take a value f_p in some label set $L=\{l_1,\ldots,l_n\}$. It has the *local characteristics*: the value F_p on pixel p only depends on the value F_q of its neighboring pixels $q\in N_p$; N_p is the set of neighbors of p and $N=\{N_p, p\in P\}$ is a neighborhood system on the field.

Pixel labelling problem is a common task in computer vision. It needs to assign one label l in L to each pixel p in P such that the configuration and the observed data are as consistent as possible. In the Bayesian framework, it is expressed as finding the maximum *a posterior* configuration of F corresponding to a realization of the random field [1]. It results in minimizing the following energy function:

$$E(f)=E_d(f)+E_s(f) \tag{1}$$

where data term $E_d(f)$, which measures the agreement between labelling and the observed data, reflects the *likelihood* of the labelling; smooth term $E_s(f)$, which measures smoothness of the labelling, reflects the *prior* probability of the labelling. This paper focuses on pairwise MRFs, i.e. the clique potentials involve only pairs of neighboring pixels. Then, $E_d(f)$ and $E_s(f)$ have the following forms:

$$E_d(f)=\sum_{p\in P} D_p(f_p) \tag{2}$$

$$E_s(f)=\sum_{p,q\in N} V_{p,q}(f_p,f_q) \tag{3}$$

where $D_p(f_p)$ measures how well f_p fits the observed data at p, and the penalty $V_{p,q}(f_p,f_q)$ makes f vary smoothly across neighboring pixels. Minimizing the data term forces the labelling to agree with observed data while as minimizing the smooth term forces the labelling varies smoothly everywhere. The effect of data term and the smooth term can be adjusted by weighting their contribution to the total energy. Moreover, discontinuity preserving energy functions can be adopted to deal with discontinuity at object boundary [15].

Since each pixel can take n labels, the total number of possible configurations is n^m. It is very difficult to search the optimal one among so many possible solutions. Simulated annealing samples the configuration space randomly to search the best configuration. It needs enough samples to find the optimal solution, therefore it is very slow. Iterated conditional modes adopt a greedy technique to find a local optimal solution. Since only one pixel can change its label at each step, they depend on the initial labelling greatly and usually cannot find good solution. $\alpha-$expansion algorithm perform expansion move to find local optimal solution. All pixels can change their labels simultaneously in each step. However, they are allowed to adopt one new label at each step. Therefore, it must test against all labels to find good solution. Similarly, $\alpha-\beta-$swap algorithm must test against every pair of labels. Both $\alpha-$expansion algorithm and $\alpha-\beta-$swap algorithm need to carry out graph cut algorithm iteratively. As a result, their complexities are $O(k*n^2)$ and $O(k*n)$, respectively.

The above approaches adopt what we called determinate labelling schema: every pixel has one determinate label in the process of labelling. Since the configuration space is very large, it costs much time to search the optimal configuration. This paper proposes a novel technique for pixel labelling: a label set instead of a label is assigned to the pixel. This schema reduces the complexity greatly.

2.2. Pixel labelling via classification

Let all pixels assigned with same label belong to one category, pixel labelling can be treated as a classification problem: classify each pixel $p\in P$ into one category $c\in C$ corresponding to one label $l\in L$. As shown in [16], classification can be carried out based on a decision tree.

2.2.1. Binary decision tree construction

Assumption 1. Assume that the given labels are

$$l_i=i-1,\quad i=1,\ldots,n \tag{4}$$

Otherwise, we can define a function mapping the original label to such labels.

We write each label $l\in L$ as a binary number and create a binary decision tree to assist the classification. First, we create a root node to denote L. Then, we create left and right child nodes to denote L^0 and L^1, respectively, L^0 and L^1 are two subsets of L: L^0 and L^1 consist of the labels with their highest bit being zero or one, i.e.

$$l\in\begin{cases} L^0, & bit(l,i)=0\\ L^1, & bit(l,i)=1\end{cases} \tag{5}$$

where i is the index of the highest bit and $bit(l,i)$ is a function returning the i th bit of label l. Similarly, we create two child nodes for each of above nodes. This process is repeated until every leaf node denotes a set contains only one label. In the example shown in Fig. 1, the whole label set is $L=\{0,1,2,3\}$ and the highest bit is 1th bit.

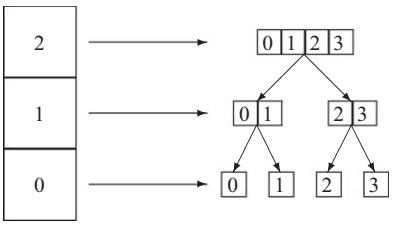

Fig. 1. Binary decision tree for classification. All brother nodes are aligned horizontally to be one layer, whose number is shown in left column. The classification is carried out layer by layer from top to bottom, one layer each time.

The binary tree is used as a decision tree for the classification. Each node denotes one subset $L' \subseteq L$, which also denotes one *category*. Each node denotes a super category of the categories corresponding to their child nodes, on the contrary, each child node denotes one subcategory of their parent node's category. Sometimes, we use such term as label set, category and node interchangeably, the meaning is indicated by their context. We call the category corresponding to a leaf node a *final category*, it corresponds to a determinate label.

2.2.2. Classification for a single pixel

Given a pixel p, it can be classified into one final category by finding a "root to leaf" path on the decision tree T. Each step towards this goal is a two-categories decision. We use a *binary-valued variable* $x \in \{0,1\}$ to represent the decision, i.e. $x=0$ and 1 indicates that $p \in L^0$ and $p \in L^1$, respectively. The decision is based on some cost function $E(x)$ measuring the probability of p belonging to the two categories:

$$x = \begin{cases} 0, & E(0) < E(1) \\ 1, & E(1) < E(0) \end{cases} \tag{6}$$

2.2.3. Classification for a pixel set

Given a label set P, all pixels can be classified into their final categories simultaneously. First, all pixels are classified into the root node, then they are classified into one of the nodes in the next layer on the decision tree. This is repeated until it reaches the bottom layer and all pixels are classified into their final categories.

Let us focus on one step of the classification corresponding to one layer on the decision tree. We use $x_p \in \{0,1\}$ to represent the decision for pixel p and $X = \{x_p, p \in P\}$ to represent the decision for all pixels. Determination of MRF X is based on the cost function as follows:

$$E(X) = E_d(X) + E_s(X) = \sum_{p \in P} E_p(x_p) + \sum_{p \in P} \sum_{q \in N_p} E_{p,q}(x_p, x_q) \tag{7}$$

where $E_d(X)$ and $E_s(X)$ are data term and smooth term, respectively. Since x_p does not indicate that p belongs to one final category and has one determinate label, the data term and smooth term cannot be defined in the traditional way.

Since there are a few layers in the decision tree, the whole classification process consists of only a few steps. All pixels can get their determinate labels within a few steps. Each step is a two-category classification and is formulated as bi-label labelling problem.

2.3. Energy construction in a probabilistic framework

As shown in Section 2.1, F_p is a random variable takes value in L. If p is classified into one category L_p, F_p can only take value in L_p. Therefore, the classifying process restricts the possible values of the random variable F_p step by step until F_p gets one determinate value. In other words, a set of possible labels instead of a determinate label is assigned to the pixels, and the set is bisected repeatedly until the pixels get their determinate label.

Suppose that L_p is assigned to p, then F_p takes any value in L_p. It is reasonable to assume that the chance of taking any label is equal. $E_p(x_p)$ can be defined by adopting "winner takes all" schema:

$$E_p(x_p) = \min(D_p(f_p), f_p \in L_p^{x_p}) \tag{8}$$

Since $D_p(f_p)$ measures the likelihood of pixel p having label f_p, $E_p(x_p)$ defined by the above equation can measure the likelihood of label f_p being in $L_p^{x_p}$, i.e. the likelihood of p belonging to the corresponding category.

Suppose that p and q are a pair of neighboring pixels, and that L_p and L_q are assigned to them. Then, F_p takes value in L_p and F_q takes value in L_q. Since the joint probability is proportional to the exponential of clique potential, we have

$$Pr(f_p, f_q) = \frac{\exp(-V_{p,q}(f_p, f_q))}{\sum_{f_p \in L_p^{x_p} f_q \in L_q^{x_q}} \exp(-V_{p,q}(f_p, f_q))} \tag{9}$$

where the denominator is a normalization constant. $E_{p,q}(x_p, x_q)$ can be defined as the expectation of $V_{p,q}(F_p, F_q)$:

$$E_{p,q}(x_p, x_q) = \sum_{f_p \in L_p^{x_p}} \sum_{f_q \in L_q^{x_q}} V_{p,q}(f_p, f_q) Pr(f_p, f_q) \tag{10}$$

Since $V_{p,q}(f_p, f_q)$ reflects the *prior* probability of p and q having f_p and f_q, $E_{p,q}(x_p, x_q)$ defined by the above equation reflects the *prior* probability of f_p and f_q being in $L_p^{x_p}$ and $L_q^{x_q}$, i.e. the probability of p and q belonging to the corresponding categories.

We can define $E_{p,q}(x_p, x_q)$ just by x_p and x_q. However, the classification process is a many to one mapping from f_p to x_p, therefore, one pair of x_p and x_q indicates many kinds of labelling and cannot distinguish the different cases. Eq. (10) calculates the expected distance between f_p and f_q. It assigns different values to different cases and makes a difference between them.

The constructed energy is an approximation to the original energy and the proposed method is an approximate approach. However, it works well in practice as demonstrated in Section 5.

3. Energy minimization via graph cut

Kolmogorov and Zabih presented a theoretic foundation on what energy can be minimized via graph cut [17]. The main result is that:

Theorem 1. *Let E be a function of m binary variables $x_i, i = 1, \ldots, m$*

$$E(x_1, \ldots, x_m) = \sum_i E_i(x_i) + \sum_{i,j} E_{i,j}(x_i, x_j) \tag{11}$$

Then E can be minimized via graph cut if and only if each term $E_{i,j}$ satisfies the inequality

$$E_{i,j}(0,0) + E_{i,j}(1,1) \le E_{i,j}(0,1) + E_{i,j}(1,0) \tag{12}$$

Following Assumption 1 and the classification schema described in Section 2.2, we have $L_p^s = \{l_{p_i}^s, i = 1, \ldots, k\}, l_{p_i}^s = i + l^s$, $s = 0, 1, l^1 = l^0 + k$. Moreover, we have:

Theorem 2. *If $L_p^s = L_q^s, s = 0, 1$ and $V_{p,q}(l_p, l_q) = |l_p - l_q|$, then*

$$E_{p,q}(0,0) + E_{p,q}(1,1) \le E_{p,q}(0,1) + E_{p,q}(1,0) \tag{13}$$

Since $V_{p,q}(l_p, l_q) = |l_p - l_q|$ denotes the distance between l_p and l_q, $E_{p,q}(s,t)$ is the weighted average of distances between labels from L_p^s and L_q^t, respectively, it can be interpreted as distance between the two label sets and written as D_{st}. As illustrated in Fig. 2, they satisfy $D_{00} + D_{11} \le D_{01} + D_{10}$. The mathematic proof is presented in the Appendix.

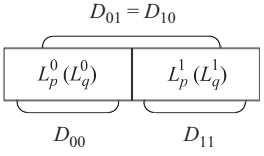

Fig. 2. Distance relationship 1. D_{st} is the distance between L_p^s and L_q^t, $s, t = 0, 1$, they satisfy $D_{00} + D_{11} \le D_{01} + D_{10}$.

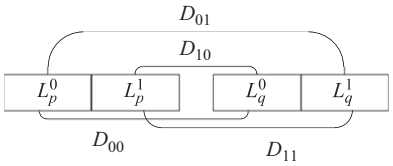

Fig. 3. Distance relationship 2. D_{st} is the distance between L_p^s and L_q^t, $s,t=0,1$, they satisfy $D_{00}+D_{11}\le D_{01}+D_{10}$.

Furthermore, we can remove the condition $L_p^s=L_q^s, s=0,1$ in Theorem 2 and have:

Theorem 3. *If* $V_{p,q}(l_p,l_q)=|l_p-l_q|$, *then*

$$E_{p,q}(0,0)+E_{p,q}(1,1)\le E_{p,q}(0,1)+E_{p,q}(1,0) \tag{14}$$

In this case, the label sets and their distances are shown in Fig. 3. It has two extremities according to the value of $d=l_{q1}^0-l_{p1}^0$:

(1) $d=0$, then $L_p^s=L_q^s, s=0,1$and $D_{00}+D_{11}<D_{01}+D_{10}$ as shown in Fig. 2.
(2) $d=\infty$, then $D_{00}=D_{11}=D_{01}=D_{10}=\infty$, and $D_{00}+D_{11}\le D_{01}+D_{10}$.

The rest cases are between these two and satisfy $D_{00}+D_{11}\le D_{01}+D_{10}$.

According to Theorems 1–3, Eq. (7) can be minimized via graph cut when the penalty $V_{p,q}(l_p,l_q)=|l_p-l_q|$. With a simple modification, we can prove that Eq. (7) can be minimized via graph cut when the penalty is Potts model. More importantly, we can remove the implicit assumption that the labels are one-dimensional and generalize the above theorems to multi-dimensional cases to deal with problems such as motion estimation. It involves more calculation but follows the same way. Therefore it is not presented in this paper.

4. Bitwise algorithm

As shown in Section 2.2.3, each step of the classification corresponds to one layer on the decision tree and determines one bit of the labels. Without loss of generality, let us assume that the labels have 8 bits and focus on the first step of the classification. As shown in Fig. 4, the highest bit of l_p is 0 and 1 when $l_p\in L_p^0$ and $l_p\in L_p^1$, respectively. Since x_p represents the decision for p, we can use x_p to set the highest bit of l_p. The rest bits of the labels can also be determined in the same way. Fig. 5 presents the *bitwise algorithm* for the pixel labelling problem.

Since the smooth term is determined by the current labels of all involved pixels and each pixel has a finite number of possible labels, the smooth term can be computed and stored in a table. Then, smooth term construction involves only table searching, the complexity is dominated by the graph cut computation. As shown, bitwise algorithm consists $nb=\log_2 n$ (n is size of label set) steps and each step involves only one graph cut computation. The complexity of the bitwise algorithm is $O(\log_2 n)$.

The higher bits are determined before the lower bits. Every set of possible labels is shrunk step by step, and contains only one label at last. Bitwise algorithm performs from coarse to fine like multi-resolution MRFs [18], but the refinement is conducted in the label field instead of site field. It suffers from the drawbacks of multi-resolution approaches: mistakes on the coarse level have strong negative impact on the fine level, and may impair the labelling quality.

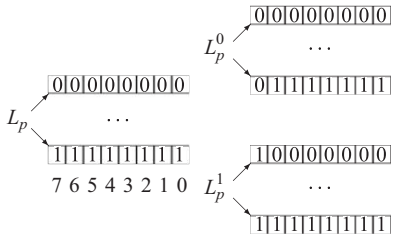

Fig. 4. Labels in the possible label sets.

1. Input nb, the number of bits of the label;
2. Let $i=nb$ and set $l_p=0$ for all $p\in P$;
3. Construct the energy (7) according to current l_p for $p\in P$;
4. Minimize the energy (7) to find $X=\{x_p, p\in P\}$;
5. Let $bit(l_p, i-1)=x_p$ for all $p\in P$;
6. Let $i=i-1$;
7. If $i=0$ finished, otherwise go to step 3;

Fig. 5. Bitwise algorithm for stereo matching.

5. Experimental results

Stereo matching and image restoration can be formulated as pixel labelling problem. Graph cut and belief propagation are the state of the art algorithms for pixel labelling. Since bitwise algorithm is built on graph cut algorithm, it is reasonable to compare bitwise algorithm with graph cut algorithms. α–expansion and α–β–swap algorithms have nearly the same performance while as α–expansion algorithm is more efficient than α–β–swap. Therefore, this section presents results on both stereo matching and image restoration, and compares it with α–expansion algorithm.

5.1. Stereo matching

In the presented experiments on stereo matching, the data terms and smooth terms are based on below functions:

$$D_p(f_p)=|I(p)-I'(p+f_p)| \tag{15}$$

$$V_{p,q}(f_p,f_q)=u_{p,q}|f_p-f_q| \tag{16}$$

$$u_{p,q}=\begin{cases}2*K & |I(p)-I(q)|\le 8\\ K & |I(p)-I(q)|>8\end{cases} \tag{17}$$

where $I(p)$ and $I'(p)$ are the gray of pixel p in each image of the stereo pair, the image sampling issue is taken into account by using measure presented in [19]. K is set to be 10.

Fig. 6 shows how bitwise algorithm determines the labels step by step. Figs. 6(a) and (b) show left and right images of tsukuba stereo pair. In this experiment, the disparity (label) range is $0\le l\le 15$, they are scaled by a factor of 16 for better illustration. Fig. 6(c) is the disparity image after one step of the bitwise algorithm, highest bit of the labels for the gray and dark pixels are 1 and 0, respectively. The disparities are refined step by step as illustrate in Figs. 6(d)–(f). The whole process consists only 4 steps. As shown, it produces good results efficiently.

The comparisons of bitwise algorithm and α–expansion algorithm are shown in Figs. 7–10. In each figure, (a) is the left image of the stereo pair and (b) is its true disparity image. (c) and (d) show disparity image computed using bitwise algorithm and

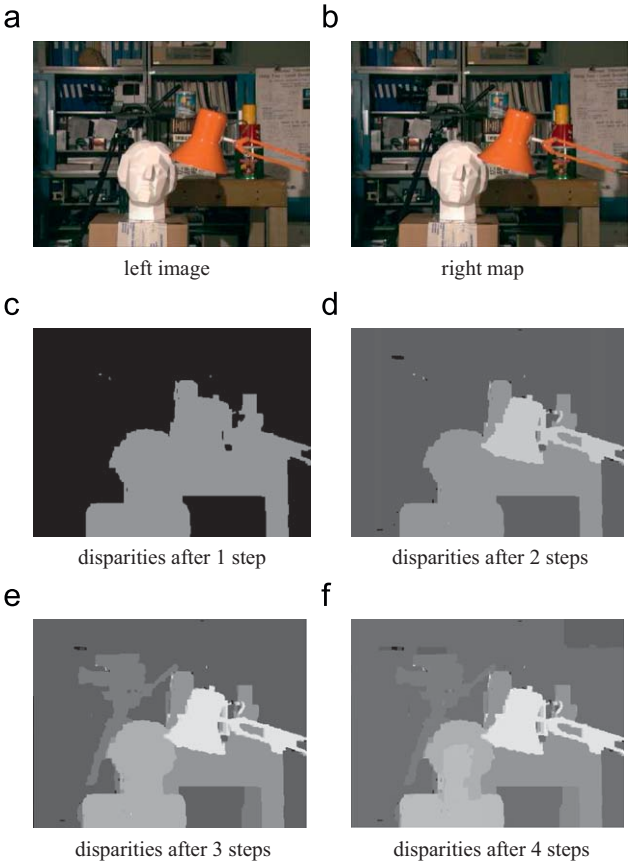

Fig. 6. Process of bitwise algorithm: (a) left image, (b) right map, (c) disparities after 1 step, (d) disparities after 2 steps, (e) disparities after 3 steps, (f) disparities after 4 steps.

$\alpha-$expansion algorithm, respectively. The computed disparity is compared with the true disparity. If the difference is larger than 1.0, it is reported as bad pixel. The percentage of bad pixels in non-occluded, textureless and discontinuity regions are recorded and presented in Tables 1 and 2. The evaluation methodology is described in [10]. As shown, these two algorithms have nearly the same performance on stereo computation. We should point out that these two algorithms do not incorporate such information as occlusion and segmentation into stereo computation, as a result, they produce results inferior to that of state of the art stereo algorithms [20–22]. Moreover, their computation times are recorded and presented in Table 3.

5.2. Image restoration

In the presented experiments on image restoration, the data terms and smooth terms are based on below functions:

$$D_p(f_p) = |I(p) - f_p| \tag{18}$$

$$V_{p,q}(f_p, f_q) = |f_p - f_q| \tag{19}$$

where $I(p)$ is the gray of pixel p.

The comparisons of bitwise algorithm and $\alpha-$expansion algorithm are shown in Figs. 11 and 12. (a) and (b) are the input image and the ground truth image. (c) and (d) are images restored by bitwise algorithm and $\alpha-$expansion algorithm, respectively. As shown, they have nearly the same performance. Their computation times are presented in Table 4.

The complexities of bitwise algorithm and $\alpha-$expansion algorithm are $O(\log_2 n)$ and $O(k * n)$, respectively. Therefore, bitwise algorithm is far more efficient than $\alpha-$expansion algorithm as demonstrated by Tables 3 and 4. The more labels, the more time is saved by bitwise algorithm.

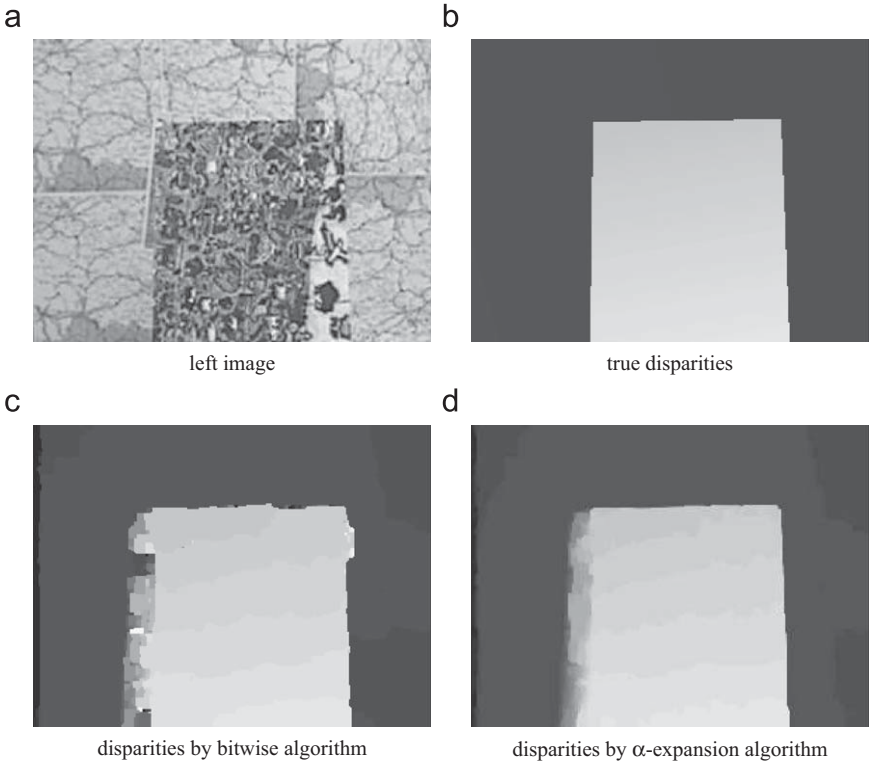

Fig. 7. Stereo matching on map image example: (a) left image, (b) true disparities, (c) disparities by bitwise algorithm, (d) disparities by $\alpha-$expansion algorithm.

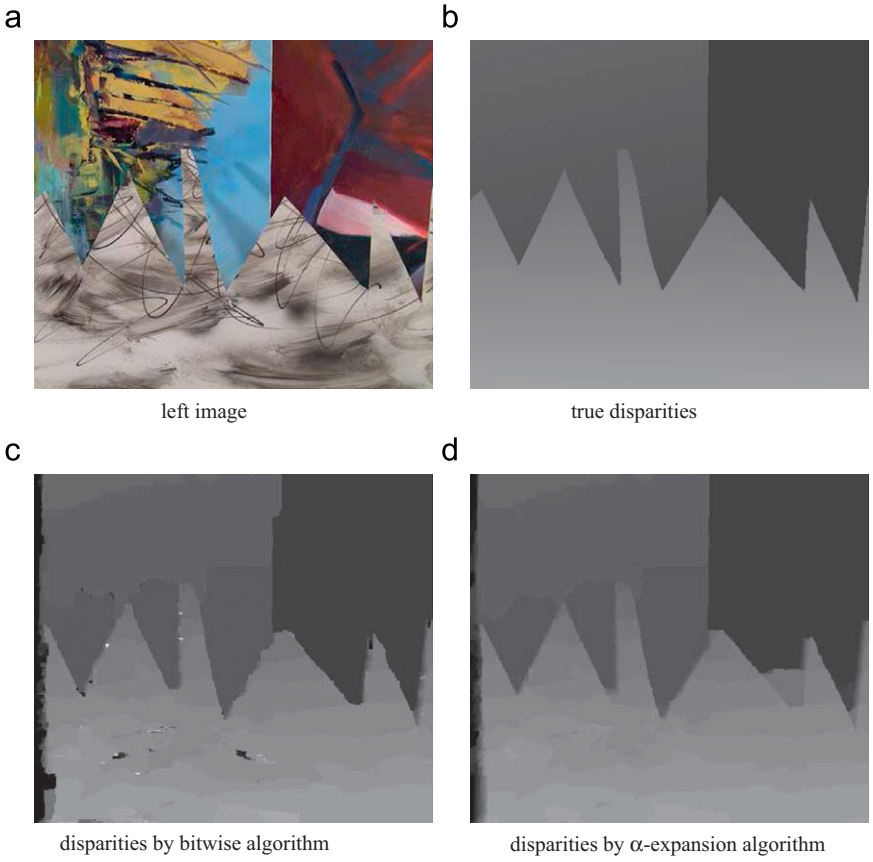

Fig. 8. Stereo matching on sawtooth image example: (a) left image, (b) true disparities, (c) disparities by bitwise algorithm, (d) disparities by α−expansion algorithm.

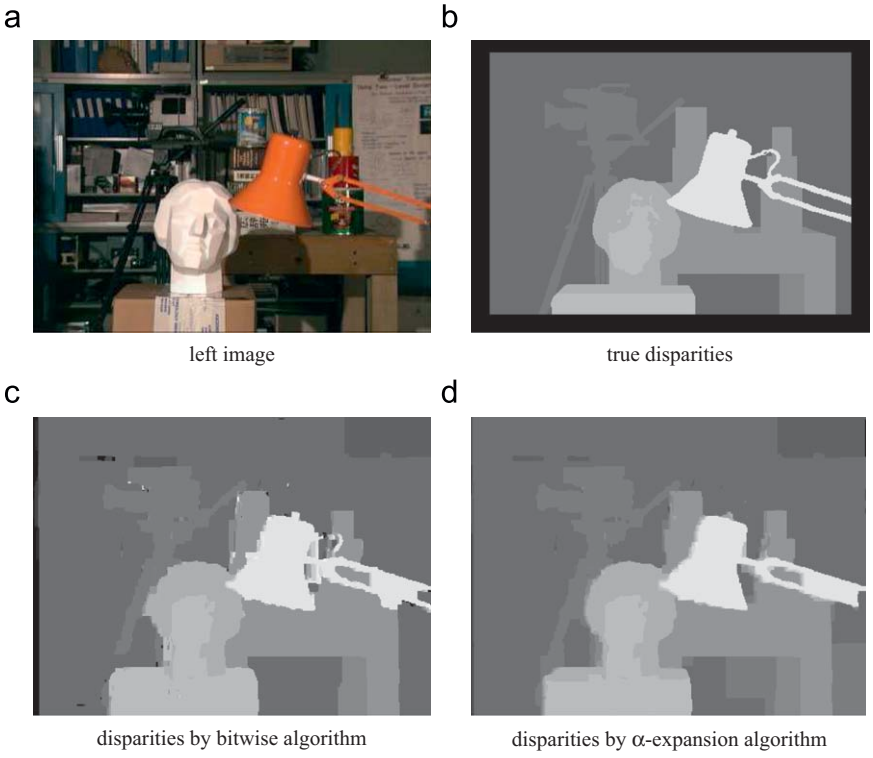

Fig. 9. Stereo matching on tsukuba image example: (a) left image, (b) true disparities, (c) disparities by bitwise algorithm, (d) disparities by α−expansion algorithm.

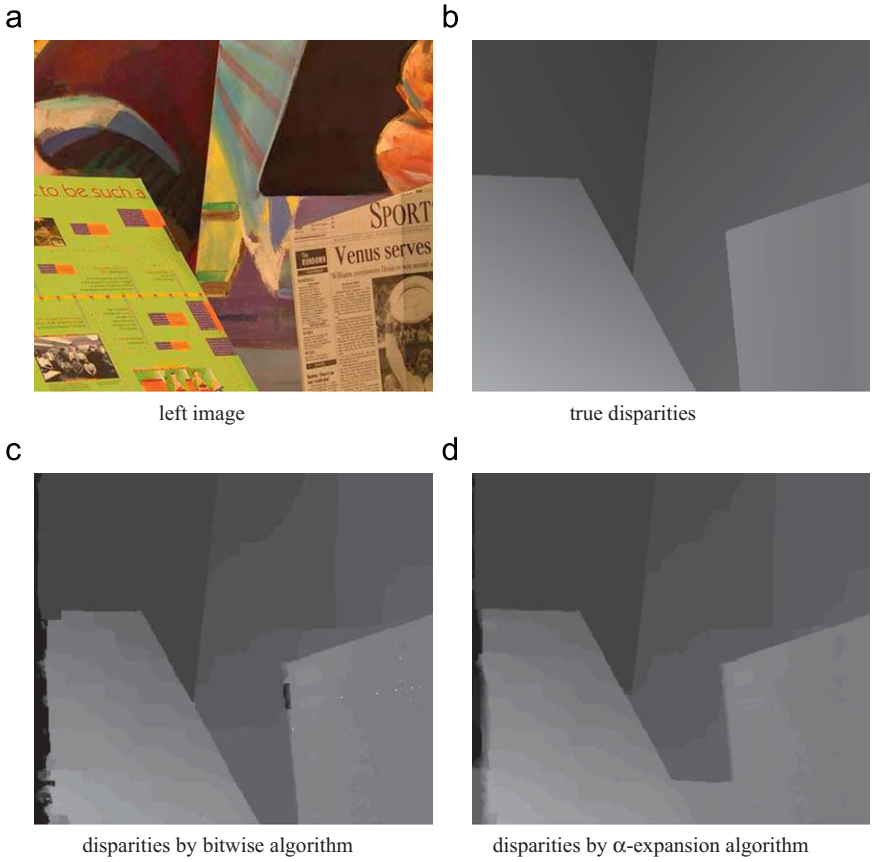

Fig. 10. Stereo matching on venus image example: (a) left image, (b) true disparities, (c) disparities by bitwise algorithm, (d) disparities by α−expansion algorithm.

Table 1
Quality evaluation of bitwise algorithm.

	Tsukuba	Sawtooth	Venus	Map
Non-occluded	3.55	3.57	1.37	0.86
Textureless	2.26	1.73	1.11	0.00
Discontinuity	15.72	10.23	6.99	9.70

Table 2
Quality evaluation of α−expansion algorithm.

	Tsukuba	Sawtooth	Venus	Map
Non-occluded	2.59	2.28	2.89	0.49
Textureless	1.57	1.78	3.69	0.72
Discontinuity	13.51	10.10	10.94	5.10

Table 3
Efficiency comparison of bitwise and α−expansion algorithms.

	Tsukuba	Sawtooth	Venus	Map
Width	384	434	434	284
Height	288	380	383	216
Disparity levels	16	32	32	32
Time (bitwise, seconds)	0.875	1.687	2.109	0.61
Time (α−expansion, seconds)	17.922	43.656	60.593	17.735

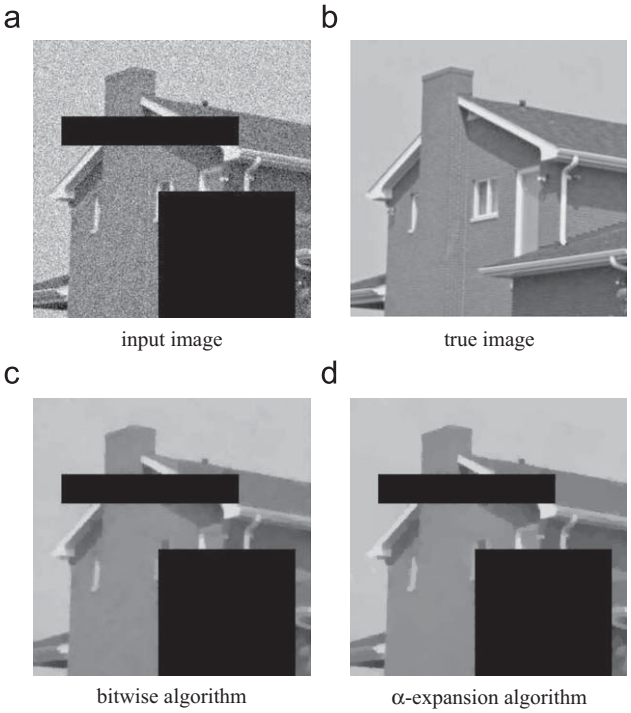

Fig. 11. Restoration on house image example: (a) input image, (b) true image, (c) bitwise algorithm, (d) α−expansion algorithm.

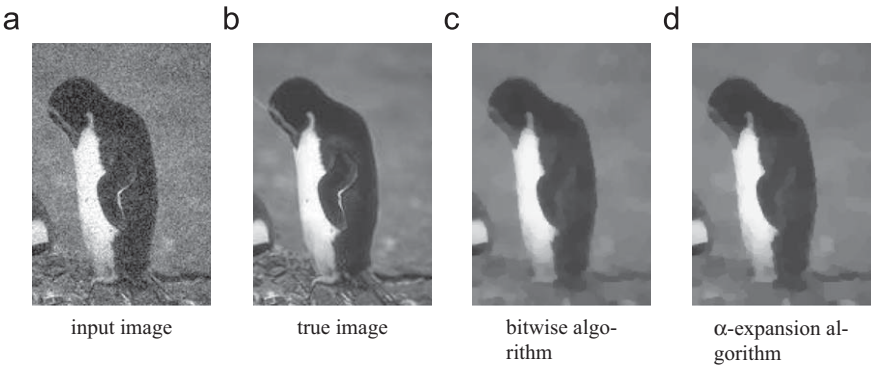

Fig. 12. Restoration on penguin image example: (a) input image, (b) true image, (c) bitwise algorithm, (d) α−expansion algorithm.

Table 4
Efficiency comparison of bitwise and α−expansion algorithms.

	House	Penguin
Width	256	122
Height	256	179
Gray levels	256	256
Time (bitwise, seconds)	0.985	0.39
Time (α−expansion, seconds)	208.968	55.063

6. Conclusion

This paper has demonstrated that pixel labelling problem can be expressed as a classification problem, and then formulated it as a series of two-category classification. In contrast with existing techniques, which assign a determinate label to each pixel, our new approach assigns a set of labels to each pixel and bisect the label set step by step until it contains only one label. Each step is formulated as a bi-label labelling problem and can be solved by standard graph cut method. Based on this formulation, we developed a bitwise algorithm for pixel labelling. The whole process consists of $\log_2 n$ steps, each step sets one bit of the labels and involves only one graph cut computation. As shown in our experiments, efficiency of the labelling is improved greatly while quality of the results is preserved.

The proposed method performs in a coarse to fine way like multi-resolution MRFs. But the refinement is conducted in the label field instead of site field. Like other multi-resolution methods, decision in the early steps has great impact on the later steps. As can be seen in Section 5, errors are usually introduced in the early steps. We plan to improve reliability of the labelling by incorporating reliable features into the labelling process. Reliable features act as control points and may improve quality of the results.

Acknowledgements

The authors would like to thank the anonymous reviewers for their critical and constructive comments and suggestions.

Appendix A. Proof of Theorem 2

First, let us compute $E_{p,q}(s,t)$ using Eqs. (10) and (9)

$$E_{p,q}(0,0)=\sum_{i=1}^{k}\sum_{j=1}^{k}V_{p,q}(l_{p_i}^0,l_{q_j}^0)Pr(l_{p_i}^0,l_{q_j}^0) \tag{A.1}$$

$$=\sum_{i=1}^{k}\sum_{j=1}^{k}|l_{p_i}^0-l_{q_j}^0|\frac{\exp(-|l_{p_i}^0-l_{q_j}^0|)}{\sum_{i=1}^{k}\sum_{j=1}^{k}\exp(-|l_{p_i}^0-l_{q_j}^0|)} \tag{A.2}$$

$$=\frac{\sum_{i=1}^{k}\sum_{j=1}^{k}|i-j|\exp(-|i-j|)}{\sum_{i=1}^{k}\sum_{j=1}^{k}\exp(-|i-j|)} \tag{A.3}$$

similarly,

$$E_{p,q}(1,1)=\frac{\sum_{i=1}^{k}\sum_{j=1}^{k}|i-j|\exp(-|i-j|)}{\sum_{i=1}^{k}\sum_{j=1}^{k}\exp(-|i-j|)} \tag{A.4}$$

$$E_{p,q}(0,1)=\frac{\sum_{i=1}^{k}\sum_{j=1}^{k}|k+j-i|\exp(-|k+j-i|)}{\sum_{i=1}^{k}\sum_{j=1}^{k}\exp(-|k+j-i|)} \tag{A.5}$$

$$E_{p,q}(1,0)=\frac{\sum_{i=1}^{k}\sum_{j=1}^{k}|k+i-j|\exp(-|k+i-j|)}{\sum_{i=1}^{k}\sum_{j=1}^{k}\exp(-|k+i-j|)} \tag{A.6}$$

From Eqs. (A.3)–(A.5), (A.6), we have $E_{p,q}(0,0)=E_{p,q}(1,1)$, $E_{p,q}(0,1)=E_{p,q}(1,0)$. If $k=1$, then $E_{p,q}(0,0)=0, E_{p,q}(0,1)=1$, $E_{p,q}(0,0)<E_{p,q}(0,1)$.

Now, we turn to cases when $k>1$. Since $i\le k$ and $j\ge 1$, we have $k+j-i\ge 1$ and

$$E_{p,q}(0,1)=\frac{\sum_{i=1}^{k}\sum_{j=1}^{k}|k+j-i|\exp(-|k+j-i|)}{\sum_{i=1}^{k}\sum_{j=1}^{k}\exp(-|k+j-i|)} \tag{A.7}$$

$$\ge\frac{\sum_{i=1}^{k}\sum_{j=1}^{k}\exp(-(k+j-i))}{\sum_{i=1}^{k}\sum_{j=1}^{k}\exp(-(k+j-i))}=1 \tag{A.8}$$

Let $S_i=\sum_{t=1}^{i-1}(t-1)\exp(-(t)), i=1,\ldots,k$, we have $S_1<S_2<\cdots<S_k$ and

$$S_k=\sum_{t=1}^{k-1}(t-1)\exp(-(t)) \tag{A.9}$$

$$S_k*e=\sum_{t=1}^{k-1}(t-1)\exp(-(t-1))=\sum_{t=0}^{k-2}t\exp(-t) \tag{A.10}$$

then, we have

$$S_k(e-1)=0+\sum_{t=1}^{k-2}\exp(-(t))-(k-2)\exp(-(k-1)) \tag{A.11}$$

$$\le\sum_{t=1}^{k-2}\exp(-(t))=\frac{e^{-1}(1-e^{-(k-2)})}{1-e^{-1}}=\frac{(1-e^{-(k-2)})}{e-1} \tag{A.12}$$

and

$$S_k = \frac{(1-e^{-(k-2)})}{(e-1)(e-1)} < \frac{1}{(e-1)^2} < 1/2 \tag{A.13}$$

Thus, $S_i < 1/2,\ i = 1,\ldots,k$, and then we have

$$E_{p,q}(0,0) - 1 = \frac{\sum_{i=1}^{k}\sum_{j=1}^{k}|i-j|\exp(-|i-j|)}{\sum_{i=1}^{k}\sum_{j=1}^{k}\exp(-|i-j|)} - 1 \tag{A.14}$$

$$= \frac{2\sum_{i=1}^{k}\sum_{j=1}^{i-1}(i-j)\exp(-(i-j))}{k+2\sum_{i=1}^{k}\sum_{j=1}^{i-1}\exp(-(i-j))} - 1 \tag{A.15}$$

$$= \frac{2\sum_{i=1}^{k}\sum_{j=1}^{i-1}(i-j-1)\exp(-(i-j)) - k}{k+2\sum_{i=1}^{k}\sum_{j=1}^{i-1}\exp(-(i-j))} \tag{A.16}$$

$$= \frac{2\sum_{i=1}^{k}\sum_{t=1}^{i-1}(t-1)\exp(-(t)) - k}{k+2\sum_{i=1}^{k}\sum_{j=1}^{i-1}\exp(-(i-j))} \tag{A.17}$$

$$= \frac{2\sum_{i=1}^{k}S_i - k}{k+2\sum_{i=1}^{k}\sum_{j=1}^{i-1}\exp(-(i-j))} \tag{A.18}$$

$$< 0 \tag{A.19}$$

Thus, $E_{p,q}(0,0) < 1 \le E_{p,q}(0,1)$ when $k > 1$.

Therefore, $E_{p,q}(0,0) < E_{p,q}(0,1)$ when $k \ge 1$ and Eq. (13) is satisfied.

References

[1] S. Geman, D. Geman, Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images, IEEE Transactions on Pattern Analysis and Machine Intelligence 6 (6) (1984) 721–741.

[2] S. Li, Markov Random Field Modeling in Computer Vision, Springer, Berlin, 1995.

[3] D. Greig, B. Porteous, A. Seheult, Exact maximum a posteriori estimation for binary images, Journal of the Royal Statistical Society, Series B 51 (2) (1989) 271–279.

[4] J. Besag, On the statistical analysis of dirty pictures (with discussion), Journal of the Royal Statistical Society, Series B 48 (3) (1986) 259–302.

[5] Y. Boykov, O. Veksler, R. Zabih, Markov random fields with efficient approximations, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, , 1998, pp. 648–655.

[6] S. Roy, I. Cox, A maximum-flow formulation of the n-camera stereo correspondence problem, in: Proceedings of International Conference on Computer Vision, , 1998, pp. 492–499.

[7] H. Ishikawa, Global optimization using embedded graphs, Ph.D. Thesis, New York University, 2000.

[8] A. Goldberg, R. Tarjan, A new approach to the maximum flow problem, Journal of the Association for Computing Machinery 35 (4) (1988) 921–940.

[9] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (11) (2001) 1222–1239.

[10] D. Scharstein, R. Szeliski, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, International Journal of Computer Vision 47 (3) (2002) 7–42.

[11] W. Freeman, E. Pasztor, O. Carmichael, Learning low-level vision, International Journal of Computer Vision 40 (2000) 25–47.

[12] J. Sun, H. Shum, N. Zheng, Stereo matching using belief propagation, in: Proceedings of European Conference on Computer Vision, , 2002, pp. 510–524.

[13] M. Tappen, W. Freeman, Comparison of graph cuts with belief propagation for stereo, in: Proceedings of International Conference on Computer Vision, , 2003, pp. 900–907.

[14] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, C. Rother, A comparative study of energy minimization methods for Markov random fields with smoothness-based priors, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (6) (2008) 1068–1080.

[15] D. Terzopoulos, Regularization of inverse visual problems involving discontinuities, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (4) (1986) 413–424.

[16] R. Duda, P. Hart, D. Stork, Pattern Classification, Wiley, New York, 2001.

[17] V. Kolmogorov, R. Zabih, What energy functions can be minimized via graph cuts?, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2) (2004) 147–159.

[18] A. Willsky, Multiresolution Markov models for signal and image processing, in: Proceedings of the IEEE, vol. 90, 2002, pp. 1396–1458.

[19] S. Birchfield, C. Tomasi, A pixel dissimilarity measure that is insensitive to image sampling, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (4) (1998) 401–406.

[20] L. Hong, G. Chen, Segment-based stereo matching using graph cuts, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, , 2004, pp. 74–81.

[21] J. Sun, Y. Li, S. Kang, H.-Y. Shum, Symmetric stereo matching for occlusion handling, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, , 2005, pp. 399–406.

[22] Y. Deng, Q. Yang, X. Lin, X. Tang, A symmetric patch-based correspondence model for occlusion handling, in: Proceedings of International Conference on Computer Vision, , 2005, pp. 1316–1322.

About the Author—DENGFENG CHAI received Bachelor's degree from Wuhan University in 1997, then received Master's degree from State Key Lab of Information Engineering in surveying, mapping and remote sensing at Wuhan University in 2000, and then received Doctor's degree from State Key Lab of CAD&CG at Zhejiang University in 2006. Since 2000, he serves as an assistant professor at College of Science in Zhejiang University. He has done research in computer vision and pattern recognition, and has published many papers on ICCV, ACCV and other related conferences.

About the Author—HONGWEI LIN received his B.Sc. from Department of Applied Mathematics at Zhejiang University in 1996, and Ph.D. from Department of Mathematics at Zhejiang University in 2004. He worked as a communication engineer from 1996 to 1999. Now, he is an associate professor in State Key Laboratory of CAD&CG, Zhejiang University. His current research interests are in computer aided geometric design, computer graphics, and computer vision. He has published over twenty papers on these areas.

About the Author—QUNSHENG PENG graduated from Beijing Mechanical College in 1970 and received a Ph.D. from the Department of Computing Studies, University of East Anglia, UK in 1983. He is a Professor of computer graphics at Zhejiang University. His research interests include realistic image synthesis, computer animation, scientific data visualization, virtual reality, bio-molecule modeling. Prof. Peng serves currently as a member of the editorial boards of several international and Chinese journals.