Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 817496, 11 pages
http://dx.doi.org/10.1155/2013/817496

Research Article

Approximated Slack Scaling for Structural Support Vector Machines in Scene Depth Analysis

Sheng Liu,[1,2] Binbin Zhai,[1] Sixian Chan,[1] Feng Li,[1] and Ye Zhan[3]

[1] *College of Computer Science & Technology, Zhejiang University of Technology, Hangzhou 310023, China*
[2] *Key Laboratory of Visual Media Intelligent Processing Technology of Zhejiang Province, China*
[3] *School of Accounting, Zhejiang University of Finance and Economics, Hangzhou, China*

Correspondence should be addressed to Sheng Liu; edliu@zjut.edu.cn

Received 31 January 2013; Accepted 31 March 2013

Academic Editor: Carlo Cattani

Based upon the framework of the structural support vector machines, this paper proposes two approaches to the depth restoration towards different scenes, that is, margin rescaling and the slack rescaling. The results show that both approaches achieve high convergence, while the slack approach yields better performance in prediction accuracy. However, due to its nondecomposability nature, the application of the slack approach is limited. This paper therefore introduces a novel approximation slack method to solve this problem, in which we propose a modified way of defining the loss functions to ensure the decomposability of the object function. During the training process, a bundle method is used to improve the computing efficiency. The results on Middlebury datasets show that proposed depth inference method solves the nondecomposability of slack scaling method and achieves relative acceptable accuracy. Our approximation approach can be an alternative for the slack scaling method to ensure efficient computation.

1. Introduction

Learning for stereo vision has been a challenging subject for a long time. Owing to the increment of ground truth datasets, considerable progress has been achieved, that is, using the scene structure of input images to learn a probability distribution model for matching [1–4] and adopting an expectation maximization algorithm to estimate disparity and then relearn the model parameters based on the estimation [5]. Although these methods have shown exciting results, the shortcoming is obvious, that is, the parameters must be preset or initialized manually on the basis of their prior knowledge. In [6], a new supervised machine learning method was proposed to handle such problem based on conditional random fields (CRFs), and the results had shown a promising future.

As mentioned above, supervised image labeling has been a long-lasting problem in computer vision. In recent years, CRFs have become a popular alternative to address this problem [7, 8], where the spatial correlations among neighboring pixels are incorporated by defining proper unary and pairwise potential functions on the related pixels. In addition, support vector machines have been widely used in image labeling [9], but they are less successful as noisy label results occurred for the absence of consideration of the spatial correlations.

Recently, structured prediction has caused widespread attention, and many new approaches have been proposed. Structured learning approaches solve the above-mentioned problems. In its computation process, both inputs and outputs are well structured, and strong internal correlations are revealed. It is formulated as the learning of complex functional dependencies between multivariate input and output representations. Structured learning has significant impact on addressing important computer vision tasks including image denoising [10], stereo [11], segmentation [12, 13], object localization [14, 15], and human pose estimation [16, 17]. A common way is to generalize the max-margin binary/multiclass classification to incorporate with structured information [14, 18–20]. It has been utilized in many respects, such as sequence labeling, image segmentation, grammar parsing, dependency parsing, bipartite matching, and text segmentation [21]. Furthermore, with the development of SVMs, structured information is introduced which

generated two new support vector machines named max-margin-based and slack-based SSVMs, respectively.

Max-margin method, with its decomposability of the error function, is possible to find the most violating constraint using the maximum a posteriori (MAP) inference algorithm for prediction [21]. But the shortcomings of the max-margin method are also obvious: it requires the error function being linearly comparable with the features, and it is sensitive to the most violating label. A label with large error would greatly decrease the separability of any other labels. An alternative choice is the slack scaling method. It has a fixed margin of 1 and reduces the violations in proportion to their errors which provide excellent accuracy. However, due to the nondecomposability of its error function, the slack method is not used widely. Therefore, we proposed an approximation method which modifies the slack method while reserving its normal properties. Depending on different given tasks, the proposed approximation method is effective to design most suitable loss functions and generate the corresponding solver.

This paper is organized as follows. In Section 2, we briefly discuss the principles of the SSVM. Our approach is proposed in Section 3 including steps to conduct the structural support vector machine, the typical max-margin method, and the expression of the improved slack method. Section 4 elaborates an approximation of the slack method. Section 5 provides the feature vectors which are utilized in our algorithm. As for Section 6, relative conditions and strategies for the training will be discussed and improved to make the training more efficient. Finally, we apply both methods for the depth restoration and make a detailed comparison between them.

2. Structural Support Vector Machine

Derived from statistical machine learning, the discriminative models focus on the posterior probability $p(y \mid x, \omega)$ and have been viewed as the most successful techniques for structural prediction. Here x is the input sample in the input space $\boldsymbol{\chi}$ and y is the associated output in the output space $\boldsymbol{\gamma}$. Given a feasible training set, for the training sample x_i and their associated truth output y_t, firstly a model for $p(y \mid x, \omega)$ will be learnt that the correct labels y_t have a higher probability than the wrong labels y, that is, $p(y_t \mid x, \omega) \geq p(y \mid x, \omega)$,and secondly, it can perform prediction by MAP estimation for a new sample x:

$$y^* = \arg\max_y p(y \mid x, \omega). \tag{1}$$

Under the framework of CRFs, $p(y \mid x)$ is modeled by a log linear model, which is often assumed as follows:

$$\log p(y \mid x, \omega) = \langle \omega, \Phi(x, y) \rangle - A_\omega(x), \tag{2}$$

where $\Phi(x, y)$ is a certain relationship between the input and its output; the second term, $A_\omega(x)$, is the normalization factor to make $p(y \mid x, \omega)$ a valid probability distribution.

By adopting the framework of max-margin method, the structural support vector machine tries to learn the weight vector, denoting the ω-parameterized model, to predict the correct output labels. And then, the optimization problem that results from the learning can be written as

$$\min_{\omega,\xi} \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{n}\xi_i \tag{3}$$

subject to

$$\langle \omega, \Phi(x_i, y_t) \rangle - \langle \omega, \Phi(x_i, y) \rangle \geq \Delta(y, y_t) - \xi_i, \tag{4}$$

here, i from 1 to n denotes different samples, y is the label that is not equal to the true label y_t and $\Delta(y, y_t)$ denotes the loss between the two labels, ξ_i is the slack variables. Thus, the most violated constrains can be found by solving

$$y^* = \arg\max_y \left(\Delta(y, y_t) + F(x, y)\right), \tag{5}$$

where $F(x, y) = \langle \omega, \Phi(x, y) \rangle$ is the discriminative function. Therefore, y^* is reformulated as the minimization problem of energy, that is, $\operatorname{argmax}_y F(x, y) = \operatorname{argmin}_y E(x, y)$.

3. Our Approach

3.1. Problem Formulation. In stereo matching tasks, stereo images are two (or more) images of the same object taken from different views, named the left image (reference image) and right image, respectively. Assume that the right view image is just a horizontal shift of the left view, and the two images are the same size $R \times C$. Denoting $I(r, c)$ is the pixel on the cross of rth row and cth column in reference image, and $I'(r, c)$ the pixel on the same position in right image. The matching is aimed at finding the pixel-wise disparity which minimizes the energy

$$\begin{aligned} Y^* &= \arg\min_{\Upsilon} E\left(I, I', Y\right) \\ &= \arg\min_{\Upsilon}\left(\sum_{r,c}\left\|I(r, c) - I\left(r, c - y_{r,c}\right)\right\|^2 + E_{\text{smooth}}(Y)\right), \end{aligned} \tag{6}$$

where $y_{r,c}$ denotes the local disparity and $E_{\text{smooth}}(Y)$ is the smooth term which usually takes the form of *Pot's Model*

$$E_{\text{smooth}}(Y) = \begin{cases} 0 & y_i = y_j \\ p & \text{otherwise}, \end{cases} \tag{7}$$

where i and j are the index of neighboring pixels, y_i and y_j represent the neighboring disparity label, and p is a constant for penalty.

Normally the features of I and I' represent certain categories of visual information, for example, color, texture, or gradient. However, each category suits different situations. Texture features work well in boundary regions which usually are rich-textured but not applicative in weakly textured regions. Gradient-based features have opposite characters in comparison with texture features. In addition, different categories of features are not easy to be combined for learning. Simply expanding the dimension of feature vectors to involve

more features from different categories is dangerous due to sampling effect and scale. The highly weighted features will greatly influence the final results, also suppressing other features. Therefore, the data term should be constructed in the form of $\langle \omega_n, \emptyset(I, I', Y)\rangle$, where ω_n is the unary weight parameter which can balance the components in the combination feature vector against the sampling effect and different scales. These parameters W can be learnt from training examples.

By expanding the squared difference in data term, we will get three terms; that is, $I^2(r,c)$, $I'^2(r,c-y_{r,c})$, and $-2I(r,c)I'(r,c-y_{r,c})$. We use $\emptyset(I,I',Y) - \emptyset(I,I',Y_t)$ as the constraints in training phase, where Y_t is the ground truth, the term $I^2(r,c)$ would be canceled out by the subtraction because of its independency of label Y. We use $\|I(r,c) - I'(r,c-y_{r,c})\|^2$ to take the place of $I^2(r,c)$. Parameters working on these terms can balance the difference between $I(r,c)$ and $I'(r,c-y_{r,c})$, which is caused by sampling effect and camera settings. Overall, the data term is built as $\emptyset(I,I',Y) = [\|I(r,c) - I'(r,c-y_{r,c})\|^2, I'^2(r,c-y_{r,c}), -2I(r,c)I'(r,c-y_{r,c})]^T$.

3.2. Max-Margin Formulation for Stereo Learning. Assuming a learnt pairwise weight $\omega_e = p$, then the parameter W can be denoted as $W = (\omega_n, \omega_e)^T$, and the energy is written as $E(I,I',Y) = \langle W, \Phi(I,I',Y)\rangle$. Here $\Phi(I,I',Y)$ is the vector including data term $\emptyset(I,I',Y)$ and also the smooth term. The energy on ground truth Y_t should be minimized, that is, for all possible Y we have $E(I,I',Y) \geq E(I,I',Y_t)$. By adopting the margin scaling and adding the slack variables ξ_t to account for violations, the optimization problem reads, for $\eta > 0$,

$$\min_{W,\xi_t} \frac{\|W\|^2}{2} + \frac{\eta}{T}\sum_{t=1}^{T}\xi_t,$$
$$\text{s.t.} \quad \left\langle W, \Phi\left(I,I',Y\right)\right\rangle - \left\langle W, \Phi\left(I,I',Y_t\right)\right\rangle \geq \Delta\left(Y_t,Y\right) - \xi_t \quad \forall t, Y, \xi_t \geq 0. \tag{8}$$

3.3. Slack Scaling Formulation. The margin rescaling method requires the label loss $\Delta(Y_t,Y)$ to be linearly comparable with the feature values $\Phi(I,I',Y)$. However, this is normally hard to be satisfied in structured learning, since $\Delta(Y_t,Y)$ counts the loss over each pixel in the image, and thus the aggregate value is much larger than feature values. Especially in stereo matching tasks, the pixel-wise loss may reach up to hundreds, which makes the overall loss even larger. Thus, we would like to adopt slack scaling, as it is invariant to the label loss scale. Nevertheless, the slack rescaling formulation is difficult to be solved, because no efficient approximation algorithm for Y^* exists. We follow the method introduced in [21] to solve this problem.

The slack rescaling optimization formulation is as follow:

$$\min_{W,\xi_t} \frac{\|W\|^2}{2} + \frac{\eta}{T}\sum_{t=1}^{T}\xi_t,$$
$$\text{s.t.} \quad \left\langle W, \Phi\left(I,I',Y\right)\right\rangle - \left\langle W, \Phi\left(I,I',Y_t\right)\right\rangle \geq 1 - \frac{\xi_t}{\Delta\left(Y_t,Y\right)} \quad \forall t, Y, \xi_t \geq 0. \tag{9}$$

4. The Approximation for Slack Scaling

For the slack scaling optimization formulation, the inference engine problem is to find

$$y^S = \arg\min_y \left(s(y) + \frac{\xi}{L(y)}\right), \tag{10}$$

where $y \in \{s(y) - s(y_i) < 1 - \xi/L(y)\}$ is the set of the most violating label, ξ is the slack variable, and $s(y) = \langle W, \Phi(I,I',Y)\rangle$, $L(y) = \Delta(Y_t,Y)$.

As it is seen in the formulation, because $L(y)$ must be considered entirely, the second part of the formula cannot be decomposed easily. Thus, an approximation y^A is used to take the place of y^S and make it possible to be decomposed into the local parts.

It should be noted that $s(y) + \xi/L(y)$ is concave, and it has been proved approximated in the form of a linear function with respect to $L(y)$ [22]. The linearization and to be approximation procedure will be shown in the following parts.

4.1. Linearization and Approximation. According to [22], a concave function can be expressed in a linear form. Therefore, (10) is expressed as

$$s(y) + \frac{\xi_i}{L(y)} \geq \max_{\lambda\geq 0}\left(s(y) - \lambda L(y) + 2\sqrt{\xi\lambda}\right). \tag{11}$$

The aim of the inference problem is to find the optimal label y which minimizes the left side of (11). Therefore, we have

$$\begin{aligned}
&\min_y \left(s(y) + \frac{\xi}{L(y)}\right) \\
&\quad = \min_y \left(s(y) - \min_{\lambda\geq 0}\left(\lambda L(y) - 2\sqrt{\xi\lambda}\right)\right) \\
&\quad = \min_y \max_{\lambda\geq 0}\left(s(y) - \lambda L(y) + 2\sqrt{\xi\lambda}\right) \\
&\quad = \max_{\lambda\geq 0}\min_y \left(s(y) - \lambda L(y) + 2\sqrt{\xi\lambda}\right).
\end{aligned} \tag{12}$$

Here, let $F'(y_\lambda;\lambda) = s(y) - \lambda L(y) + 2\sqrt{\xi\lambda}$, thus

$$F(\lambda) = \min_y F'\left(y_\lambda;\lambda\right), \tag{13}$$

which leads to the simplified formulation as

$$\min_{y}\left(s(y)+\frac{\xi}{L(y)}\right)=\max_{\lambda\geq 0}\min_{y}F'(y_\lambda;\lambda)=\max_{\lambda\geq 0}F(\lambda). \tag{14}$$

For a fixed λ, firstly the most optimal label y_λ can be computed through minimization

$$y_\lambda=\arg\min_{y}\left(s(y)-\lambda L(y)\right). \tag{15}$$

Then, y_λ can be substituted into the formula $F(\lambda)$. We can find a λ that enables $F(\lambda)$ to catch its maximum, because $F(\lambda)$ is a function which is convex with respect to λ. $F(\lambda)$ can be seen as the max of a set of convex functions; therefore, $F(\lambda)$ is convex as well.

With the help of linear search algorithm such as Golden Search, the maximum of $F(\lambda)$ can be acquired in an efficient way. During the search procedure, it will encounter many different λs. By evaluating the $F(\lambda)$ for each λ, we can get different labels. The goal is to find the optimal label to get a minimum of $s(y)+\xi/L(y)$, which is denoted as y^A.

4.2. The Determination of Interval for λ. Since a simple constrain has been given out, $\lambda \geq 0$, it is obvious that $\lambda = 0$ can be the lower bound of λ as λ_l. However, if $\lambda = 0$, it will be hard to distinguish the $F'(y_\lambda;\lambda)$ between different labels in the early iterations, due to the neglect for the different loss $L(y)$. Let $\lambda_l = \varepsilon/L_{\max}$, where $L_{\max}$ is the possible maximal label loss and ε is the tolerance of the difference between two continuous iterations for this algorithm. In this way, a proper correct λ_l is obtained.

Then we come to determine the upper bound of λ. It is sufficient to find an upper bound λ as λ_u such that it returns $F(\lambda) = F(\lambda_u)$ for any $\lambda \geq \lambda_u$. And it also satisfies

$$y'=\arg\min_{y}s(y)-\lambda_u L(y), \tag{16}$$

which leads to the following formula

$$s(y')-\lambda_u L(y')\leq\min_{y,L(y)<L(y')}s(y)-\lambda_u L(y). \tag{17}$$

Here, let $y_1 = \operatorname{argmin}_y s(y)$ and L_ε be the minimal difference between $L(y)$ and $L(y')$, such as $L_\varepsilon = 1$ for Hamming loss. Then the right side of the function becomes $s(y_1)-\lambda_u(L(y')-L_\varepsilon)$. That requires $\lambda_u \geq s(y')-s(y_1)$ ($L_\varepsilon = 1$).

Since $s(y') < s(y_i)+1-\xi/L(y') < s(y_i)+1-\xi/L_{\max}$, so λ_u can be set as $\lambda_u = s(y_i)+1-\xi/L_{\max}-s(y_1)$.

5. Construction of Feature Vector

Image features are the terms used to describe images, as well as the clues for distinguishing the differences of images. Some image features may be the basic visual features, while others are defined for specific applications. Three types of features are used in this paper, that is, color, texture, and edge features.

5.1. Color Features. Color features are the basic visual description of images. Generally, color features are based on the characteristics of pixels, and each pixel in the image or the image region makes its own contribution to the color features. However, as a global feature, it is not sensitive to the changes of the size of the image or image region and also the directions in image. In other words, color features cannot capture the local characteristics of the image. And due to its nonuniqueness, pixels in different objects may share the same color features. Two basic color descriptions are RGB color space and YCbCr color space. While RGB concentrates on the gray levels of the pixels, the YCbCr pays close attention to the intensity, chromaticity, and the color difference. In YCbCr color space, the channel Y represents the intensity of the color, while channels Cb and Cr denote the chromaticity for blue and red, respectively. YCbCr color space can be easily obtained just by a linear transformation from RGB color space. Both the RGB and YCbCr color features are shown in Figure 1. In this paper, we use both RGB and YCbCr as the color features in the training process.

5.2. Texture Features. Similar to color features, texture features are also global features. The major difference is that texture features describe the statistical characteristics of the pixels in the image region. And the texture features have the properties of rotational invariance and noise immunity, but they are sensitive to the revolution of images, if the revolution changes, different features may be generated. On top of that, the light and the reflection on the surface of the objects may make it hard for computing the texture features.

In [23], Laws developed a method for computing texture features. According to this method, different convolution kernels, which were named Laws' masks, will be applied to our images. And the results will give some characteristics of the images. Here, the 2D Laws' masks can be generated from the following small kernels both with the length 3 and 5: $L_3 = [1\ 2\ 1]$, $E_3 = [1\ 0\ -1]$, $S_3 = [1\ -2\ 1]$, $L_5 = [1\ 4\ 6\ 4\ 1]$, $E_5 = [-1\ -2\ 0\ 2\ 1]$, $S_5 = [-1\ 0\ 2\ 0\ -1]$, $W_5 = [-1\ 2\ 0\ -2\ 1]$, $R_5 = [1\ -4\ 6\ -4\ 1]$.

Here, L denotes the average gray levels, E denotes the edge features, S stands for extracting the spots in the image, W stands for extracting the wave feature, and R stands for extracting the ripples in the image.

In order to generate the 2-D Laws' masks, we adopted matrix multiplication by a vertical 1D kernel and a horizontal 1-D kernel, such as $L_5E_5 = L_5^T \times E_5$. Take the masks scaled 3×3, for example, all the possible masks were listed in Table 1. After the convolve operation with these masks on an image sized M × N, the gray-scale texture feature image sized (N-masks_size + 1) × (M-masks_size + 1) will be generated. Figure 2 demonstrates the texture feature results generated by the 3×3 Laws' masks.

5.3. Edge Features. The object edge is the visual features of the discontinuity in the local image region which has a significant change in intensity. Generally, in images, the pixels along the edge have a smooth change in gray levels; however, on the

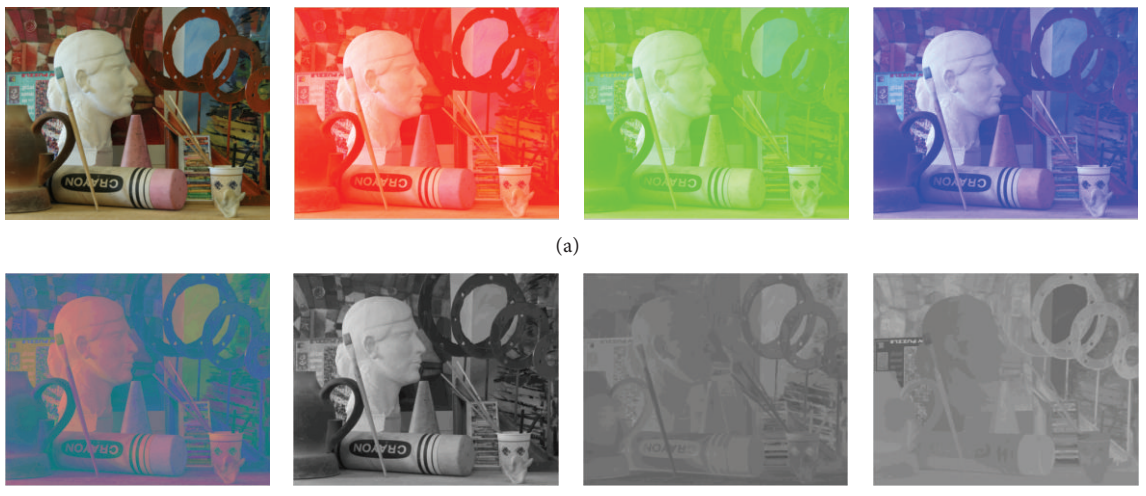


Figure 1: The color features of the image: RGB color features (first row) and YCbCr color features (second row). From left to right, first row: the original image in RGB color space, R channel, G channel, and B channel; second row: the original image in YCbCr color space, Y channel, Cb channel, and Cr channel.

Table 1: The possible Laws' masks scaled 3×3.

Masks	Method	Description
L_3L_3	$L_3^TL_3$	The gray level intensity within 3 neighboring pixels in both vertical and horizontal directions
L_3E_3	$L_3^TE_3$	In horizontal direction edge diction and in vertical direction gray level intensity
L_3S_3	$L_3^TS_3$	In horizontal direction spots detection and in vertical direction gray level intensity
E_3L_3	$E_3^TL_3$	In horizontal direction gray level intensity and in vertical direction edge diction
E_3E_3	$E_3^TE_3$	Edge detection in both vertical and horizontal directions
E_3S_3	$E_3^TS_3$	In horizontal direction spots detection and in vertical direction edge diction
S_3L_3	$S_3^TL_3$	In horizontal direction gray level intensity and in vertical direction spots detection
S_3E_3	$S_3^TE_3$	In horizontal direction edge diction and in vertical direction spots detection
S_3S_3	$S_3^TS_3$	Spots detection in both vertical and horizontal directions

direction which is vertical to the edge, the intensity of pixels change sharply.

The former denoted features are the local visual features. From the description, they are the surface features of the objects. On the other hand, the edge features are the measurement of the local compatibility. In this paper, 4 different Prewitt edge detectors which were directed in 0°, 45°, 90°, and 135° were adopted in order to extract the edge features. The detectors in different directions and corresponding results are shown in Figure 3. By applying the 4 detectors, almost all the edges in the images can be captured.

6. Parameter Learning and Inference Problem

6.1. Bundle Method for Parameter Learning. For parameter learning, this paper utilizes the bundle method. Due to the formulation, such as

$$\min_{W,\xi_t} \frac{\|W\|^2}{2} + \frac{\eta}{T}\sum_{t=1}^{T}\xi_t,$$
$$\text{s.t.} \quad \left\langle W, \Phi\left(I, I', Y\right)\right\rangle - \left\langle W, \Phi\left(I, I', Y_t\right)\right\rangle \geq \Delta\left(Y_t, Y\right) - \xi_t \quad \forall t, Y, \xi_t \geq 0. \tag{18}$$

In order to obtain the optimal parameter, the constraints can be rearranged in the following form:

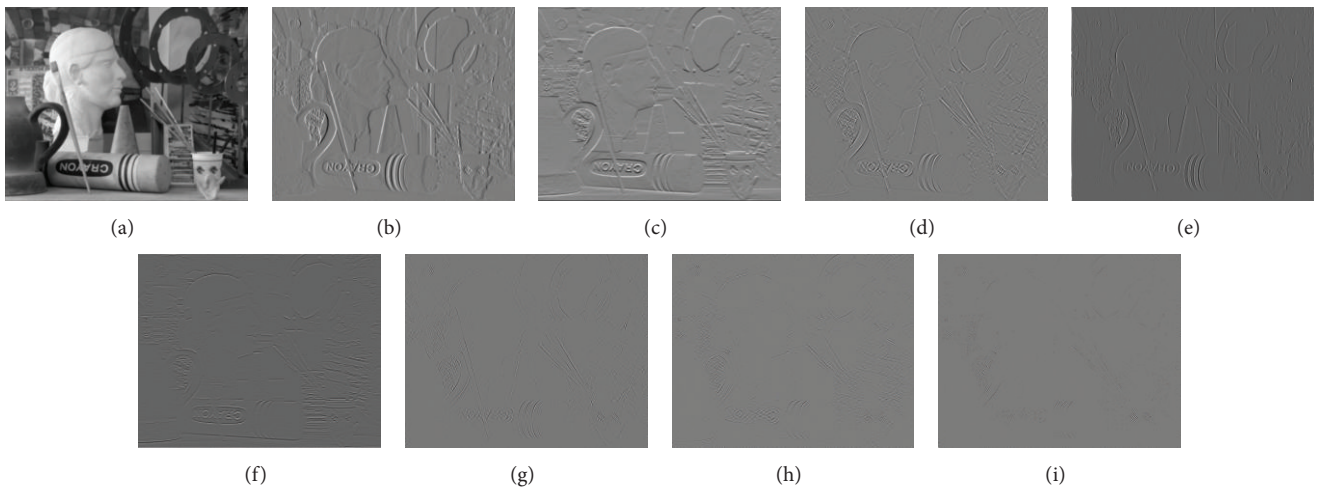
$$\left\langle W, \Phi\left(I, I', Y_t\right)\right\rangle \geq \left\langle W, \Phi\left(I, I', Y\right)\right\rangle - \Delta\left(Y_t, Y\right) + \xi_t. \tag{19}$$

This formula means that it is lower bounded by $\langle W, \Phi(I, I', Y_t)\rangle$. Then it generates the objective function to find the most violated constraints

$$Y^* = \arg\min_{Y}\left(\left\langle W, \Phi\left(I, I', Y\right)\right\rangle - \Delta\left(Y_t, Y\right)\right). \tag{20}$$

Thus, this forms an inference problem. And the bundle method can guarantee the optimal solution in a small number of iterations, so the problem can be solved efficiently. Algorithms 1 and 2 provide the parameter learning algorithm for both margin and slack method.

Both the margin and slack method refer to the optimal inference problems, so the best solution for them can be obtained via a standard graph-cuts algorithm (see reference [8] for detail). The frameworks seem to be the same, but in Algorithm 2, the inference engine is not similar to that in Algorithm 1. In this case, it needs to be approximated into a linear form, so that it searches for the best λ in the interval by the golden search algorithm.



(a) (b) (c) (d) (e)

(f) (g) (h) (i)

Figure 2: The outputs after the convolution of all the Laws' masks scaled 3×3.

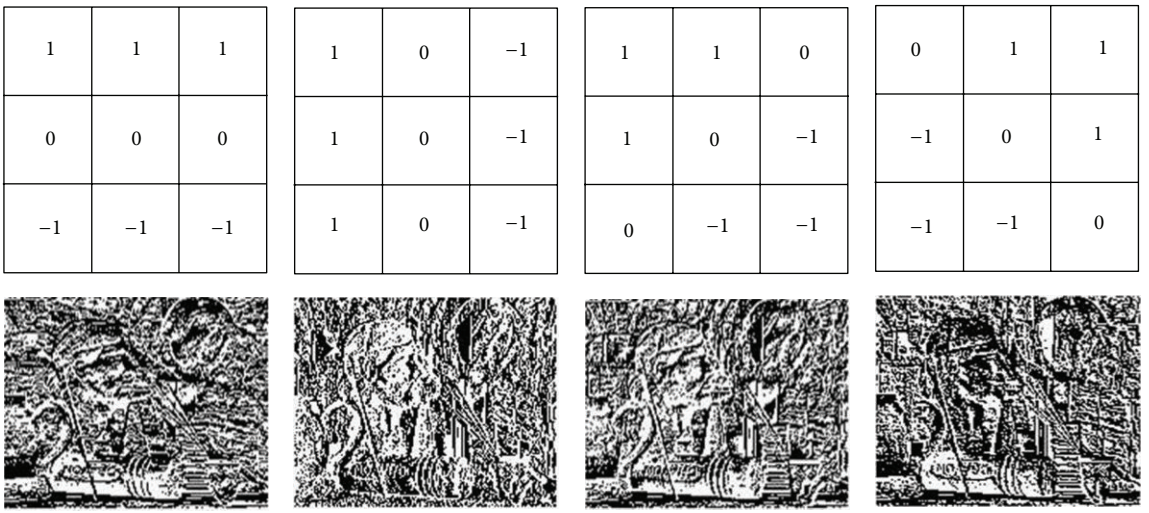


Figure 3: The results achieved by different edge detectors in 4 directions.

6.2. Golden Searching. In this paper, we adopted the golden searching algorithm during searching for the best approximation of the optimal label.

Firstly, suppose that there exists a continuous concave function $\mathbf{f}$ over the interval $[a, b]$, meanwhile it has only one minimum or maximum in the interval. Taking the minimum case for example, the binary searching algorithm is not the optimal algorithm for minimum searching, shown as follows:

Take the middle point as

$$m = \frac{a+b}{2}, \tag{21}$$

then two different points x_1 and x_2 are determined by

$$\begin{aligned} x_1 &= m - \frac{\delta}{2} \\ x_2 &= m + \frac{\delta}{2}, \end{aligned} \tag{22}$$

such that $f(x_1) \neq f(x_2)$. If $f(x_1) < f(x_2)$, the interval will be updated by $[a, x_1]$, otherwise $[x_2, b]$ will be the new interval. Obviously, each iteration step should call the binary searching for two times, which is not optimal.

In order to optimize the iteration process, there should be a factor which is capable of reducing the interval, named

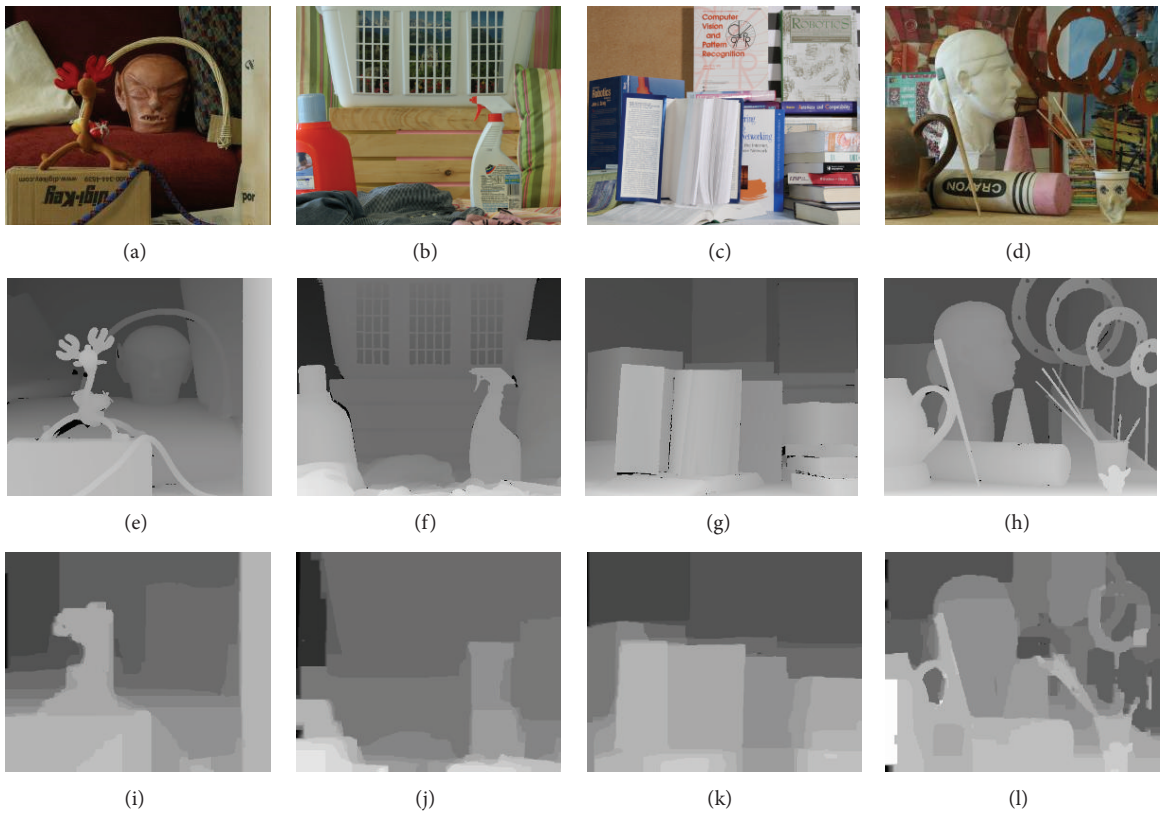


FIGURE 4: Inference depth maps by Max-margin method for different scene. From row 1 to 3: images, ground truth, and the obtained depth map. From column 1 to 4 are four scenes: 1st art, 2nd book, 3rd laundry, and 4th reindeer.

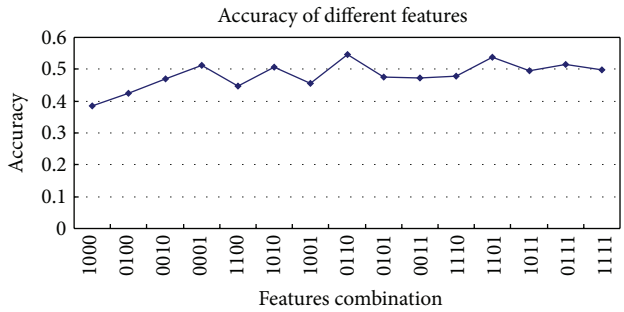


FIGURE 5: The inference accuracy of different features combination.

c. For x_1 and x_2 in the interval $[a, b]$, there are two different cases.

(1) If $f(x_1) < f(x_2)$, then the interval becomes $[a, x_2]$, and the interval size is compressed by c as follows:

$$\frac{x_2 - a}{b - a} = c, \tag{23}$$

as a result,

$$x_2 = (1 - c)\, a + cb. \tag{24}$$

(2) If $f(x_1) > f(x_2)$, similarly the interval is compressed by c and the new interval is $[x_1, b]$, then

$$\frac{b - x_1}{b - a} = c, \tag{25}$$

x_1 is obtained by

$$x_1 = (1 - c)\, b + ca. \tag{26}$$

Obviously, if the factor c is determined, it is easy to locate the points x_1 and x_2 in the interval. There are two rules for Cases (1) and (2), respectively, while Algorithm 3 shows the algorithm for golden searching.

Rule 1. If $f(x_1) < f(x_2)$, set $x_2 = x_1$, then compute another new x_1.

Rule 2. If $f(x_1) > f(x_2)$, set $x_1 = x_2$, then compute another new x_2.

7. Experiments and Results

We test the proposed methods on the Middlebury stereo datasets. The dataset contains many different scenes, that is, *art, books, dolls, laundry, moebius,* and *reindeer*, and each scene is consisted of 2 ground-truth images, related to view 1 and view 5 in each scene, and several different images which

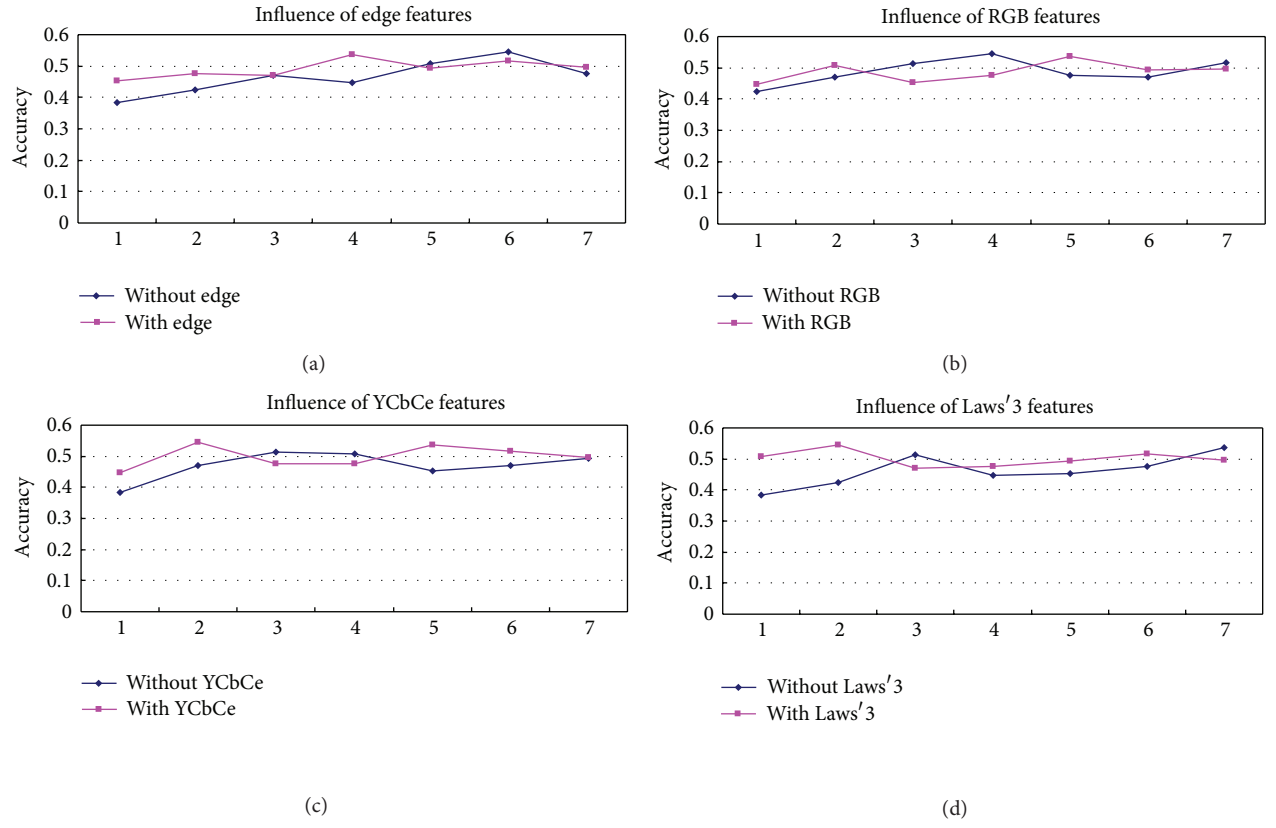


Figure 6: (a) the effect of the edge features. 1 to 7 means 1000, 0100, 0010, 1100, 1010, 0110, 1110. It shows that the edge features can boost the accuracy. (b) the effect of the RGB features. 1 to 7 means 0100, 0010, 0001, 0110, 0101, 0011, 0111. It shows that the RGB features can reduce the accuracy and the color feature with the texture features that can boost the inference accuracy. (c) the effect of the YCbCr features. 1 to 7 means 1000, 0010, 0001, 1010, 1001, 0011, 1011. It is easy to find that the YCbCr features have a similar effect on the accuracy with the RGB features. (d) the effect of the Laws' masks scaled 3×3. 1 to 7 means 1000, 0100, 0001, 1100, 1001, 0101, 1101. It is easy to find that the texture features can boost the accuracy in most of the situations.

were caught from different views. The ground-truth images are used as the label images of each scene, and its labels were compressed from 0–255 to 0–22 for the computing efficiency, and two neighbor view images are adopted to extract the different features.

Two groups of features are introduced in our experiments. The first group is local visual features, such as colors and textures, including the 3 dimensions of RGB color channels, the 3 dimensions YCbCr color channels, the 9 dimensions texture features, the outputs of Laws' masks scaled 3×3, and the 4 dimensions edge features, the outputs of the different Prewitt edge detectors. The second group is the graph edge features, which are the absolute difference between labels of neighboring pixels and one-dimensional bias constant. Practically, the method for conducting features may construct a large amount of dimensions, which can supply a rich set for choosing the suitable features to learn the parameters of the wanted model. By adopting the features and the Max-margin method, it may be easy for us to get the reasonable depth for different scenes, as shown in Figure 4.

7.1. Comparison on Inference Accuracy with Different Feature Combination. Suppose that the ground truth is denoted as y_t and the output results as y_o. Defining C_y as the number of the matched pixels in y_t and y_o and C_n as the number of different pixels in y_o from y_t, the inference accuracy can be denoted as

$$\text{acc} = \frac{C_y}{C_y + C_n}, \tag{27}$$

which stands for the ratio of the correct output.

In order to study the effects of different features, we have tested different combinations of image features. For the convenience of the expression, 1 denotes the state of the feature which was chosen, and 0 otherwise. Figure 5 shows the inference accuracy of different feature combinations for the 2nd scene *book*. Note, the order of the features arranged from left to right is RGB, YCbCr, laws' masks scaled 3×3, and the edge features. For example, 1000 denoted that only the RGB feature was chosen.

The combination of features does not always boost the accuracy of the results. In a word, some features have a negative effect on the results while others have a positive effect. In order to test it, a comparison between the set with a certain feature and another without it has been carried out. The results show that an offset effect does exist between

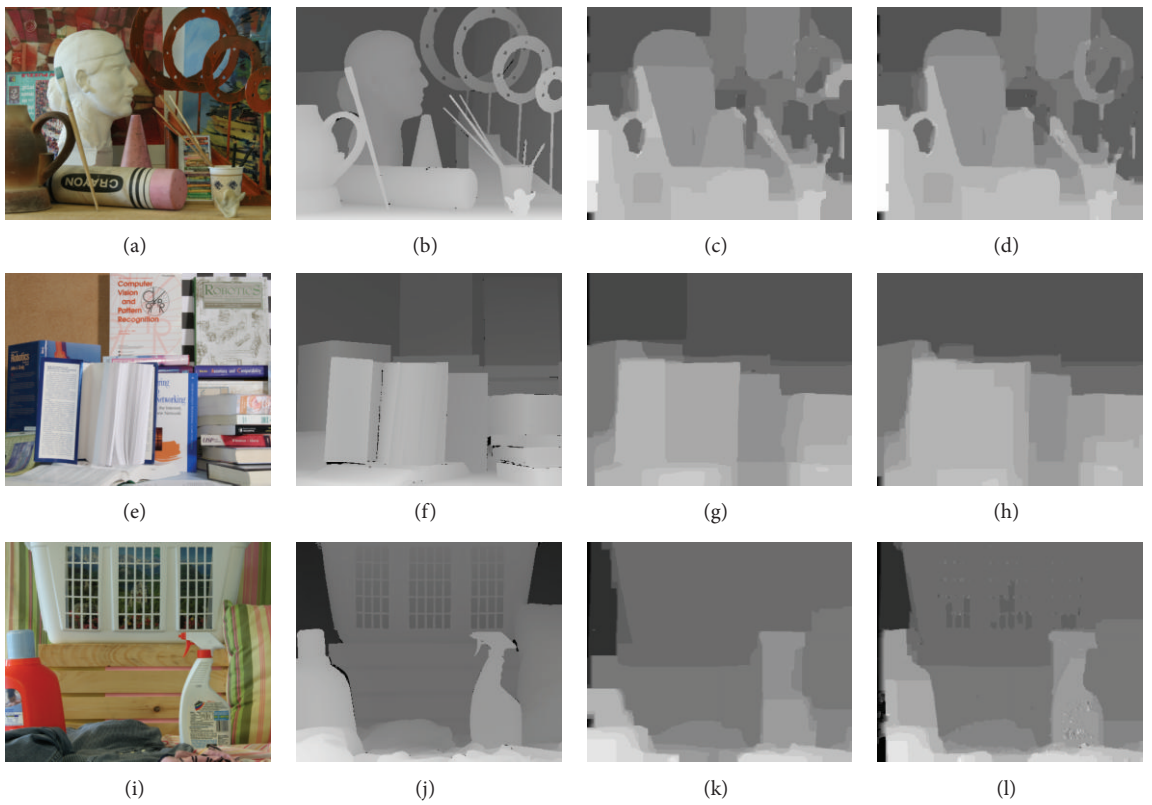


Figure 7: Depth inference results of different images by Max-margin and proposed slack method. From column 1 to 4: images, ground truth, the result of margin method, and the result of proposed slack method. And from row 1 to 3 are three scenes in Middlebury datasets: *art, book*, and *laundry*.

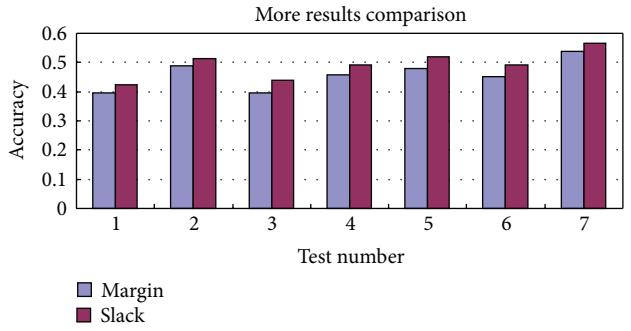


Figure 8: Different inference accuracy shows that the proposed slack method performs better than the margin method through the comparison of the inference accuracy.

Input: data X_t, label Y_t, size T, tolerance ε
Initialize parameter $W \rightarrow 0$, constraint set $R \rightarrow \emptyset$
Repeat
  for $t = 1$ to T
    $Y^* = \arg\min_Y [s(y) - \Delta(y_t, y)]$
  end for
  increase constraint set $R \leftarrow R \cup \{Y^*\}$
  $(W, \xi) \leftarrow$ solve the QP using all the existing Y^*
Until $\sum_t [\Delta(y_t, y) + s(y_t) - s(y)] \leq \xi + \varepsilon$

Algorithm 1: The parameter learning for margin method.

features, such as between color and edge feature, and also some features do boost the result, such as the textures in most situations (see Figures 6(a), 6(b), 6(c), and 6(d)).

7.2. Comparison between Margin and Slack Methods. To overcome the above-mentioned shortcomings of the Max-margin method, this paper adopts the slack scaling method to improve the results. In order to solve the nondecomposability problem, we introduce an approximated algorithm as described in Section 4 to make the slack method feasible. Both methods are tested on the Middlebury database, see Figure 7. As in Figure 8, the comparison results of inference accuracy for scene *art* show that the slack method performs better than the margin method.

7.3. Comparison on the Convergent Properties. To take a step further, the convergent property between margin and slack methods is compared. In the training procedure, the convergence of both margin and slack methods requires the use of the bundle method and *one-slack* trick. Take the

Input: data X_t, label Y_t, size T, tolerance ε
Initialize parameter $W \rightarrow 0$, constraint set $R \rightarrow \emptyset$
Repeat
  for $t = 1$ to T
    $Y^* = \arg\min_Y [s(y) + \xi/\Delta(y_t, y)]$
  end for
  increase constraint set $R \leftarrow R \cup \{Y^*\}$
  $(W, \xi) \leftarrow$ solve the QP using all the existing Y^*
Until $\sum_t [\xi/\Delta(y_t, y) + s(y_t) - s(y)] \leq \xi + \varepsilon$

Algorithm 2: The parameter learning for slack method.

Input: interval $[a, b]$, reduction factor c, tolerance ε
Initialize $x_1 \rightarrow x_1 = (1 - c)\, b + ca$,
  $x_2 \rightarrow x_2 = (1 - c)\, a + cb$,
Repeat
  If $(f(x_1) < f(x_2))$
    $b = x_2$, $x_2 = x_1$
    $f(x_2) = f(x_1)$
    $x_1 \rightarrow x_1 = (1 - c)\, b + ca$
  Else
    $a = x_1$, $x_1 = x_2$
    $f(x_1) = f(x_2)$
  $x_2 \rightarrow x_2 = (1 - c)\, a + cb$
  end If
Until $abs\,(x_1 - x_2) \leq \varepsilon$

Algorithm 3: The algorithm for golden searching.

margin method for example, the bundle method is used by rearranging the terms, then the constraints will be

$$\xi \geq \Delta(y, y_t) + s(y_t) - s(y). \tag{28}$$

This means that the constraints are up bounded by ξ. Given the current parameter, the objective function can be optimized using the bundle method, where the most violation constraint is

$$y^* = \arg\min_y (s(y) - \Delta(y_t, y)). \tag{29}$$

While the bundle method has the ability to achieve the optimal solution, the *one-slack* trick makes the procedure convergent in a small number of iterations. The computing process of the margin and the slack methods is examined to observe the convergence speed of the iteration. The error between two continuous iterations in the objective function is denoted as *itaeps*. Figure 9 shows the convergent property, indicating that both methods could converge in several iterations, while the slack method produces better accuracy without too much loss in convergence.

8. Conclusion

This paper presented two methods for the depth restoration of different scenes using structural vector machine. The proposed methods, including both margin and slack, have their own advantages and disadvantages, respectively. While the form of margin rescaling method can be decomposed into local parts easily, it is hard for the slack rescaling method to perform such operation. In contrast, the slack one outperforms the margin rescaling method in accuracy outstandingly. Besides the advantageous promotion in accuracy, there is no need for the slack rescaling method to abandon too many convergences while computing the parameters. The proposed approximation aiming at the slack rescaling approach manages to solve the decomposability problem successfully and make it computable in an efficient way. The pity is that the approximation method requires the formulation being concave which may be an over strong constraint. Our future works focus on these optimization algorithms, including improving the computing speed and enhancing the accuracy of the results.

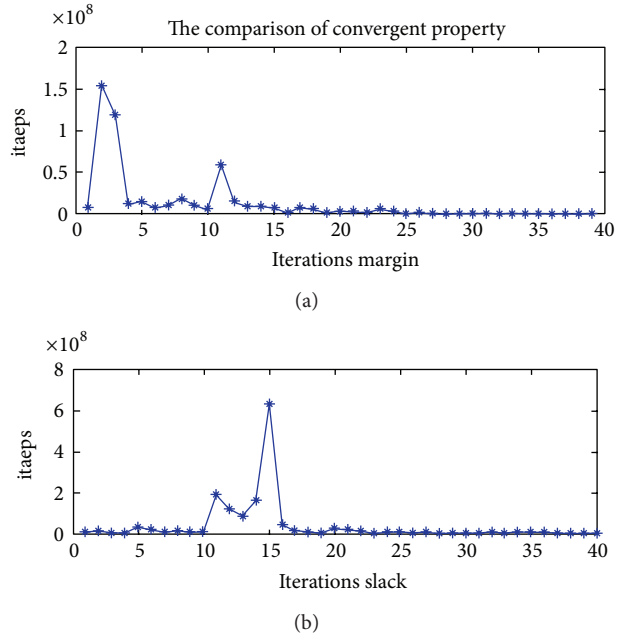


Figure 9: The comparison of the convergent property between margin and slack. Both the two methods converge in a small number of iterations. With the increasing accuracy, the slack method has a pronounced advantage in convergence compared to the margin method.

References

[1] D. Kong and H. Tao, "A method for learning matching errors in stereo computation," in *Proceedings of the British Machine Vision Conference (BMCV '04)*, 2004.

[2] S. Y. Chen and Z. J. Wang, "Acceleration strategies in generalized belief propagation," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 1, pp. 41–48, 2012.

[3] W. L. Wang, B. B. Xia, Q. Guan, and C. Shengyong, "Neural network based 3D model reconstruction with highly distorted stereoscopic sensors," in *Proceedings of the 8th International Workshop on Artificial Neural Networks (IWANN '05)*, vol. 3512 of *Lecture Notes on Computer Science*, pp. 661–668, June 2005.

[4] T. Fridman, J. Razumovskaya, N. Verberkmoes, G. Hurst, V. Protopopescu, and Y. Xu, "The probability distribution for a random match between an experimental-theoretical spectral pair in tandem mass spectrometry," *Journal of Bioinformatics and Computational Biology*, vol. 3, no. 2, pp. 455–476, 2005.

[5] L. Zhang and S. M. Seitz, "Parameter estimation for MRF stereo," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 288–295, June 2005.

[6] Y. Li and D. P. Huttenlocher, "Learning for stereo vision using the structured support vector machine," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, June 2008.

[7] D. Batra, R. Sukthankar, and T. Chen, "Learning class-specific affinities for image labelling," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, June 2008.

[8] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[9] N. Cristianini and J. S. Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, 2000.

[10] J. J. McAuley, T. S. Caetano, A. J. Smola, and M. O. Franz, "Learning high-order MRF priors of color images," in *Proceedings of the 23rd International Conference on Machine Learning (ICML '06)*, pp. 617–624, June 2006.

[11] P. Carr and R. Hartley, "Minimizing energy functions on 4-connected lattices using elimination," in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 2042–2049, October 2009.

[12] M. Szummer, P. Kohli, and D. Hoiem, "Learning CRFs using graph cuts," in *Proceedings of the European Conference on Computer Vision (ECCV '08)*, pp. 582–595, 2008.

[13] D. Anguelov, B. Taskar, V. Chatalbashev et al., "Discriminative learning of Markov Random fields for segmentation of 3D scan data," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, pp. 169–176, June 2005.

[14] B. Taskar, V. Chatalbashev, and D. Koller, "Learning associative Markov networks," in *Proceedings of the 21th International Conference on Machine Learning (ICML '04)*, pp. 807–814, July 2004.

[15] C. Ionescu, L. Bo, and C. Sminchisescu, "Structural SVM for visual localization and continuous state estimation," in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 1157–1164, September 2009.

[16] M. Blaschko and C. Lampert, "Learning to localize objects with structured output regression," in *Proceedings of the European Conference on Computer Vision (ECCV '08)*, pp. 2–15, 2008.

[17] M. Kim and V. Pavlovic, "Dimensionality reduction using covariance operator inverse regression," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, June 2008.

[18] L. Bo and C. Sminchisescu, "Structured output-associative regression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '09)*, pp. 1–8, 2009.

[19] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *Journal of Machine Learning Research*, vol. 6, pp. 1453–1484, 2005.

[20] B. Taskar, C. Guestrin, and D. Koller, "Max-margin Markov networks," *Advances in Neural Information Processing Systems*, vol. 16, 2003.

[21] S. Sarawagi and R. Gupta, "Accurate max-margin training for structured output spaces," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 888–895, July 2008.

[22] B. Taskar, C. Guestrin, and D. Koller, "Max-margin Markov networks," *Advances in neural information processing systems*, vol. 16, 2003.

[23] K. Laws, *Textured image segmentation [Ph.D. thesis]*, University of Southern California, 1980.