Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 568034, 20 pages
http://dx.doi.org/10.1155/2014/568034

Research Article

Self-Similarity Based Corresponding-Point Extraction from Weakly Textured Stereo Pairs

Min Mao,[1] Kuang-Rong Hao,[1,2] and Yong-Sheng Ding[1,2]

[1] *College of Information Sciences and Technology, Donghua University, Shanghai 201620, China*
[2] *Engineering Research Center of Digitized Textile & Fashion Technology, Donghua University, Ministry of Education, Shanghai 201620, China*

Correspondence should be addressed to Kuang-Rong Hao; krhao@dhu.edu.cn and Yong-Sheng Ding; ysding@dhu.edu.cn

Received 13 March 2014; Revised 15 August 2014; Accepted 15 August 2014; Published 3 September 2014

Academic Editor: Yi Chen

For the areas of low textured in image pairs, there is nearly no point that can be detected by traditional methods. The information in these areas will not be extracted by classical interest-point detectors. In this paper, a novel weakly textured point detection method is presented. The points with weakly textured characteristic are detected by the symmetry concept. The proposed approach considers the gray variability of the weakly textured local regions. The detection mechanism can be separated into three steps: region-similarity computation, candidate point searching, and refinement of weakly textured point set. The mechanism of radius scale selection and texture strength conception are used in the second step and the third step, respectively. The matching algorithm based on sparse representation (SRM) is used for matching the detected points in different images. The results obtained on image sets with different objects show high robustness of the method to background and intraclass variations as well as to different photometric and geometric transformations; the points detected by this method are also the complement of points detected by classical detectors from the literature. And we also verify the efficacy of SRM by comparing with classical algorithms under the occlusion and corruption situations for matching the weakly textured points. Experiments demonstrate the effectiveness of the proposed weakly textured point detection algorithm.

1. Introduction

Local interest points can be used for many applications, such as video analyzing, object detection, localization, and identification. Techniques which only use images to reconstruct 3D scenes are now the most popular of its applications. Since many algorithms based on the diversity of concepts such as graph cuts [1] and minimal path search [2] can only handle short-baseline stereo matching, that is, they cannot be used at the wide-baseline situation, on the other hand, gradient detectors can be used for wide-baseline matching well; in general, the process for matching the discrete image points can be divided into three main steps. First, extracting the "interest points" from each image, such as T-junctions, corners, and the point detector should have the property of repeatability, which guarantee finding the same physical interest points under different viewing conditions. Next, each interest point can be represented by a feature vector through the descriptor, which should also be distinctive, finally, matching the descriptor vectors between different images.

Almost all of detectors are based on the gradient map of image; for example, the Harris corner detector [3] is based on the second moment matrix, which describes the gradient distribution in a local neighborhood of a point in image, but corners detected by this method are not scale invariant. Mikolajczyk and Schmid [4] proposed two scale-invariant methods, that is, Harris-Laplace and Hessian-Laplace, which are based on the concept of automatic scale selection [5], and the location is selected by Harris measure or the determinant of the Hessian matrix; scale is selected by Laplacian; Lowe [6] speeds up the above methods by using the difference of Gaussians (DoG) to approximate the Laplacian of Gaussians (LoG). There are lots of different detectors which have been proposed in the literature [7–10]. However, problems of methods based on interest point still remain, such as image blurring, magnification, and illumination; one of the most serious weaknesses between these is that these methods could not get point on the low texture areas. Since these areas can be defined by its gradient below some constant thresholds, it is difficult to extract points on them by classical interest-point detectors.

As a guiding principle for describing shapes this characteristic has a rich history. Animals, man-made objects, and plants are usually with symmetry characteristic. Many techniques have been proposed to analyze this characteristic. For example [11] describes an algorithm to segment objects into terms of points, line segments, and circles. Loy and Zelinsky [12] use the local radial symmetry to highlight points of interest within a scene, which need not consider the contribution of a local neighborhood to a central pixel. In this paper we focus on the characteristic of strong self-similarity. The approach proposed in [13] inspires us that the region of weakly textured area has the characteristic of strong self-similarity; hence the weakly textured points can be located by this characteristic; moreover the locations with self-similar structure of local pattern are also distinguishable in different images. Our goal is to develop both a detector and matching algorithm for getting corresponding-point in weakly textured regions under wide-baseline situation. Our work in this paper can be divided into three parts, namely, candidates of weakly textured point detection, refining the set of candidates, and weakly textured point matching algorithm based on sparse representation.

In the detecting part, the proposed detector is based on the entities: circumferences and radii; it sums the value on circumferences and radii; the information from image is very first level of data processing, which is contrary to gradient-based methods; hence, for giving the scale invariant as the gradient-based methods we introduce a mechanism for automatic radial selection. The following two approaches can be identified; the first one is used to obtain the symmetry maps for each pixel in different radius scales, which is also used in [14]. The second approach selects the radius scale for each pixel.

In the refinement part, the set of weakly textured points can be refined by computing the gradient magnitude in scale-space [15], which is widely used in image processing community, such as feature description, point detection, and image structure analysis. Here we use this theory to measure the texture strength and for reducing the points which do not belong to weakly textured point; we also propose a threshold selection mechanism.

In the point matching part, the correspondence relationship of points which contain the same scene in two images will be determined. The matching between two points set is a one-to-one mapping between the points in these two sets. There are many algorithms for computing the similarity between two point features, such as zero-mean normalized cross correlation (ZNCC) [16] and Hausdorff distance [17], which many algorithms have been suggested for. We here introduced the sparse representation concept [18] for weakly textured point matching, which has been used for human face recognition in [19]. This approach considers all possible supports (here is the set of weakly textured points in the second image) and can automatically choose the minimal number of samples needed to represent each feature of point in the first image.

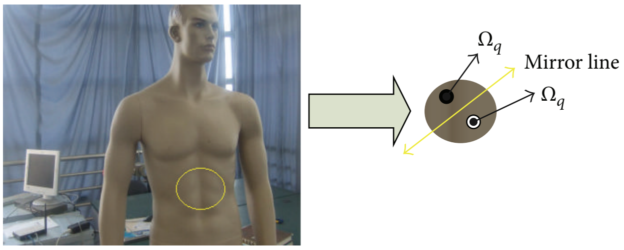


Figure 1: Example of a weakly textured region in human model.

The approach proposed in this work gives a new detector for extracting the weakly textured points from image in which the objects have weakly textured characteristic. Moreover, since the detector is based on radial symmetry, it is not sensitive to variations in image illumination. In the experiment for detectors, the results obtained on image sets of different image sets under different types of geometric and photometric transformations show that the extracted points by this detector are the complement of classical detectors and high robustness with intraclass variations. In the point matching algorithms test experiment, the method based on sparse representation outperforms other algorithms under the pixel corruption and block occlusion situations, respectively, and the results also show that the extracted weakly textured points are also distinguishable from locations; finally we use our algorithm to promote the quasi-dense matching algorithm which was proposed in [20] to verify the proposed approach under wide-baseline situation.

This paper is organized as follows: in Section 2 the method is proposed to detect the candidate of weakly textured regions. In Section 3, we sketch out a mechanism for points refinement via scale-space selection theory. In Section 4, the sparse representation theory is used for matching weakly textured point. In Section 5, the proposed method is tested in accordance with two protocols [21, 22] for evaluation of local region detectors and matching algorithm. Section 6 provides concluding remarks and possible extensions of the proposed method.

2. Weakly Textured Feature Points Detection

Given an image I with the weakly textured region Ω_{WT} and texture region Ω_T, it is clear that there has no corner-point or have few edge-points in the neighbor region of the points in Ω_{WT} (see Figure 1). Let $I(x, y)$ denote the gray value of image; then the points' characteristic in the region Ω_{WT} can be represented by its first- and second-order partial derivatives as

$$I_x, I_y, I_{xx}, I_{yy}, I_{xy}, I_{yx} \approx 0, \quad \{(x, y) \in \Omega_{\mathrm{WT}}\}. \tag{1}$$

For the above mentioned, the detectors based on the gradient feature such as the Hessian, Harris, and SIFT detectors could not be used for detecting these points. So these points should be detected in different ways. As shown in Figure 1, the region formed by weakly textured pixel and its neighborhoods can be segmented in many fragments and the pixel's gray value distribution in each fragment is similar.

Let $I(x)$ be the intensity value at location; Ω_p denotes the reference fragment in the image; then the self-similar fragment Ω_p can be measured by the normalized correlation coefficient as follows:

$$\operatorname{ncc}(P,T) = \frac{\sum_i \left(I(x_i)-\bar{I}\right)\left(I(T(x_i))-\bar{I}\right)}{\sqrt{\left(\sum_i \left(I(x_i)-\bar{I}\right)^2\right)\left(\sum_i \left(I(T(x_i))-\bar{I}\right)^2\right)}}, \quad \{p \neq q \mid q = 1,2,\dots,N\}, \tag{2}$$

where $I(T(x)) = a + bI(x)\ \forall x \in \Omega_p$.

$\bar{I}$ denotes the average intensity value of fragment Ω_p. The purpose of the formula $I(x_i) - \bar{I}$ is to reduce the influence caused by intensity; $T(\cdot)$ denotes a geometrical transformation.

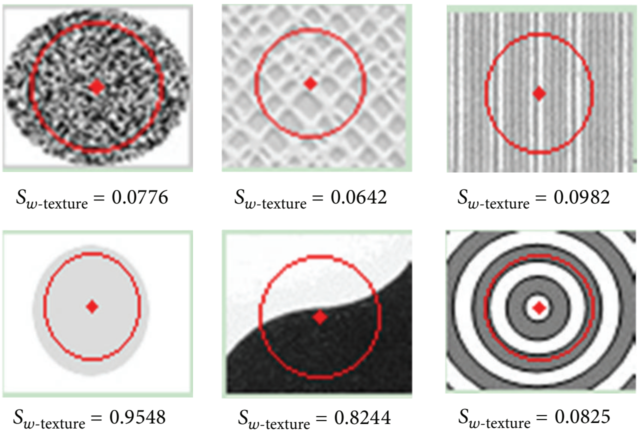
From Figure 1, it is clear that the weakly textured region has strong mirror symmetry; let point $p = (r,\theta)$ be represented in polar coordinates, let $\varphi \in [0,\pi)$ denote the mirror line orientation, and then the symmetric point of p about the mirror line can be represented as $p' = (r, 2\varphi - \theta)$. For measuring the mirror symmetry about the region Ω_p, it should fulfill all mirror line orientation $\varphi \in [0,\pi)$; here the symmetry of region Ω_p can be obtained as follows [23]:

$$S_{w\_\text{texture}}\left(\Omega_p\right) = \frac{1}{N}\sum_{i=0}^{N-1} \operatorname{ncc}\left(\Omega_p, T\left(\Omega_p, \varphi_i\right)\right), \tag{3}$$

where $T(\cdot)$ denotes the symmetry transformation function, and it transforms the Ω_p to its symmetric region about the mirror line orientation $\varphi_i \in \{\varphi_0, \varphi_1, \dots, \varphi_{N-1}\}$, $\Delta\varphi = \pi/N$, and $\varphi_i = i \times \Delta\varphi$. It is easy to prove that the $\max(S_{w\text{-texture}}(\Omega_p)) = 1$ and $\min(S_{w\text{-texture}}(\Omega_p)) = 0$, $\Omega_p \in I$.

2.1. Same Radial for $S_{w\text{-texture}}$. In the example shown in Figure 2 the strength of $S_{w\text{-texture}}$ with the same radial in different images is affected by the similarity of its region, as we can see that the strong texture region has small value of $S_{w\text{-texture}}$; in contrast, the value of $S_{w\text{-texture}}$ in weakly textured region is close to 1. So $S_{w\text{-texture}}$ can be used for detecting the weakly textured region. However the middle-down part of Figure 2 shows that the region with strong similarity but not weakly textured region has high value of $S_{w\text{-texture}}$; this case indicates that the weak texture cannot be detected by $S_{w\text{-texture}}$ only. And in Section 3, we will solve this problem by the differential expressions at the center point of region Ω_p.

2.2. Radial Selection for $S_{w\text{-texture}}$. According to the above discussion, we use $S_{w\text{-texture}}$ as our weakly textured detector (WTD); however, there are many factors which can affect the performance of WTD, including the structure complexity distributed within the local region, variety illumination for the local region, and radials chosen by WTD. Here we investigate the performance at which WTD measures region's texture strength with different radials.


$S_{w\text{-texture}} = 0.0776$ $S_{w\text{-texture}} = 0.0642$ $S_{w\text{-texture}} = 0.0982$

$S_{w\text{-texture}} = 0.9548$ $S_{w\text{-texture}} = 0.8244$ $S_{w\text{-texture}} = 0.0825$

Figure 2: $S_{w\text{-texture}}$ in different image structures with the same radius.

As shown in Figure 3, the texture strength is reflected by the intensity of the pixel in the similarity map; that is, the pixel with lower texture strength has higher intensity. When the radial change from small to large the intensity of the pixel with low texture strength will become lower, whereas the intensity of the pixel with high texture strength will become higher.

The points detect by WTD with different radials are shown in Figure 4. Obviously, the result is largely affected by the radials of the operator.

Here we proposed a mechanism for automatic radial selection. The radial selection mechanism should follow heuristic principles: first, the radial must be large enough to contain the weakly textured region; that is, the radial should extend the weakly textured region until the region bound close to the texture feature. Second, the points chosen by this mechanism should have the maxima value of $S_{w\text{-texture}}$ with its radial. Third, the mechanism should have robustness to small variations caused by fragments; that is, the radial can ignore small fragments in the weakly textured region.

As shown in Figure 5, $S_{w\text{-texture}}$ varies with the radius, and let S_R be the derivative of $S_{w\text{-texture}}$ with radius. In differential geometric terms, the fine-radius for each weakly textured points should simply satisfy $S_{RR} = 0$, which is the second-order derivative of $S_{w\text{-texture}}$. On the other hand, when considering the influence caused by neighboring fragments, we here use the following *γ-parameterized normalized derivative:* $\partial_{RR}(R^{\gamma} S_{w\text{-texture}}) = 0$. According to this formula the influence caused by the fragments can be eliminated under the fitted parameterized γ, with $\gamma = 0.5$. Figure 6 shows the results with this radius selection method.

3. Weakly Textured Points Filter for Fine Positioning

In Section 2 we have analyzed the weakly textured feature and use WTD with proposed adaptive radial mechanism to position the weakly textured point. However the problem

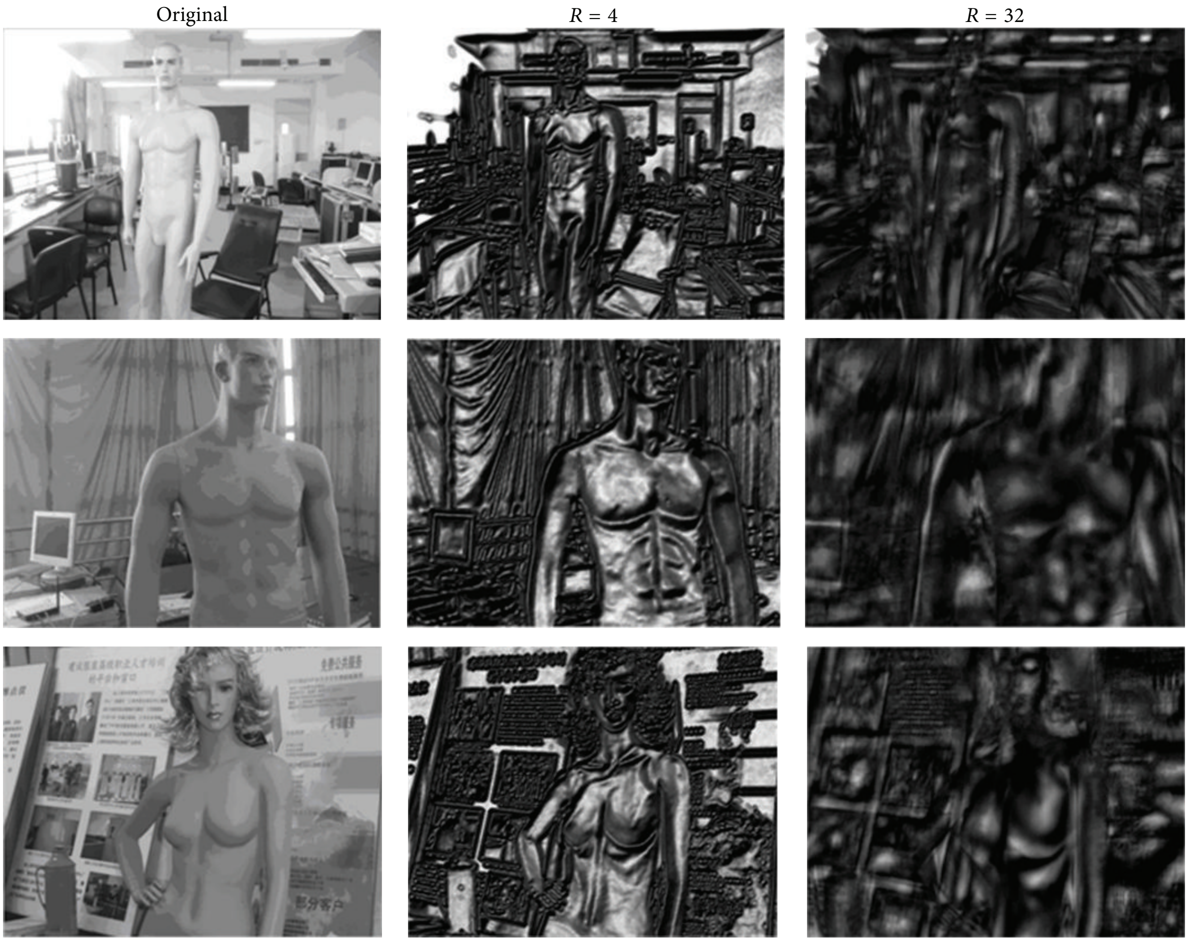


Figure 3: Similarity maps under radii $R = 4$ and $R = 32$.

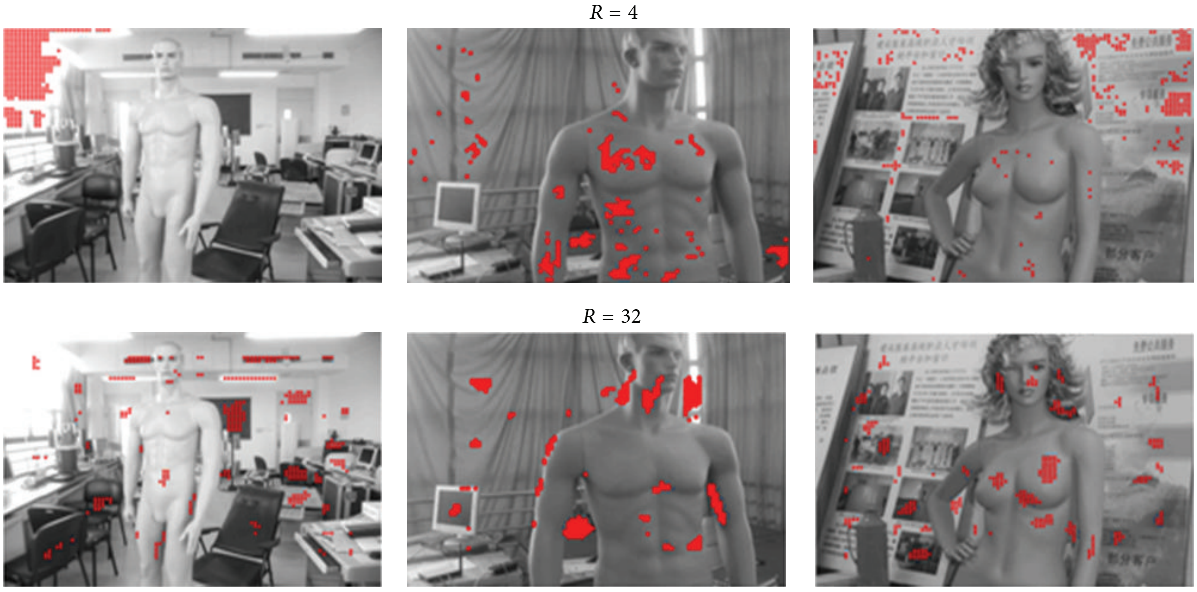


Figure 4: Extracted points under radii $R = 4$ and $R = 32$.

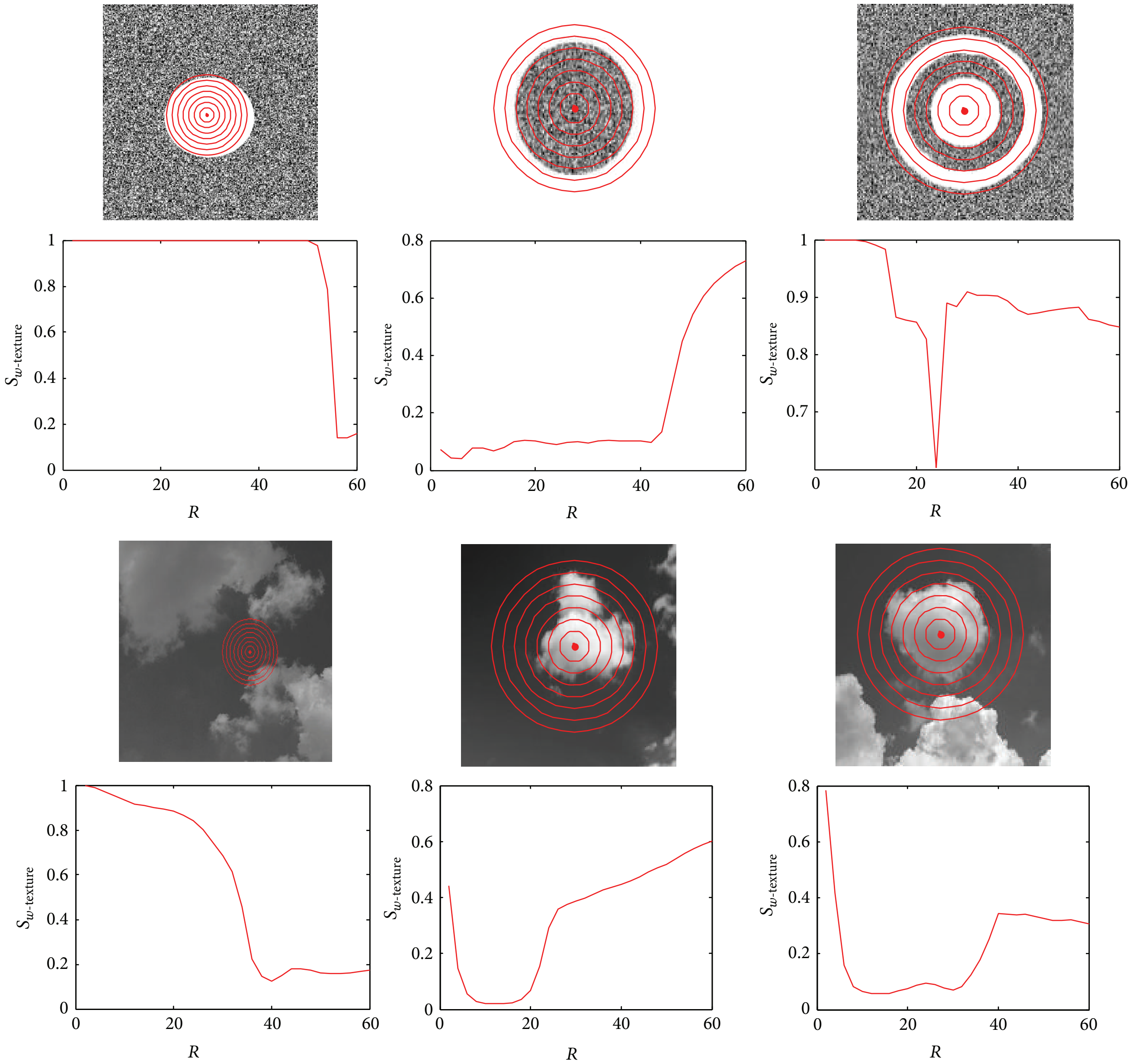


Figure 5: $S_{w\text{-texture}}$ varies on a scale from $R = 2$ to $R = 60$.

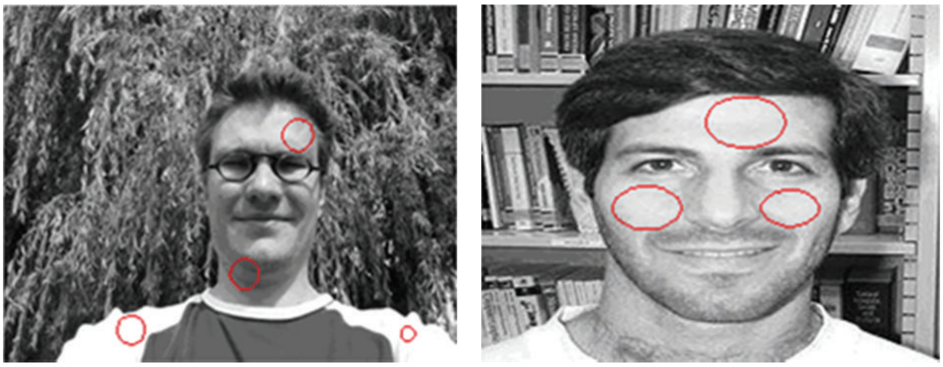


Figure 6: Radius selected for weakly textured points.

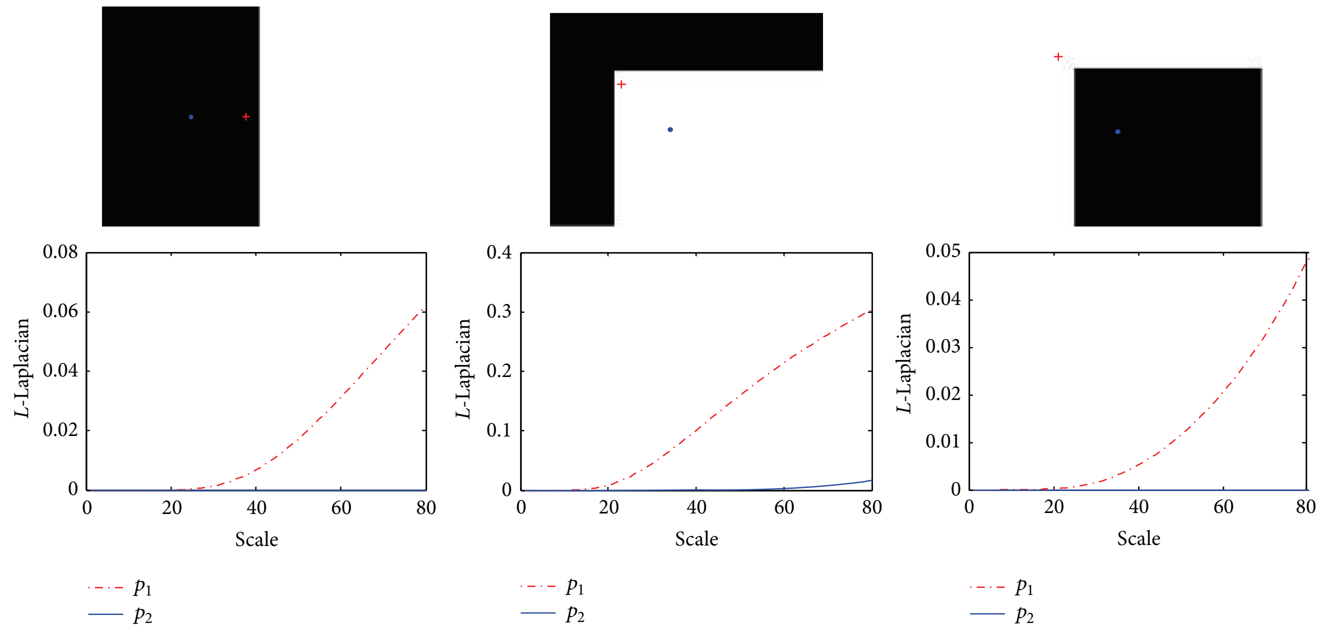


FIGURE 7: L_{grad} varies on scale-space from $t = 10$ to $t = 80$.

in the middle-down part of Figure 2 shows that the point has large value of $S_{w\text{-texture}}$ which is not the weakly textured point. On the other hand, the state-of-the-art edge detector corresponds to detecting points with the maximum gradient magnitude in the gradient direction, so edges can be detected in scale-space; here we extend these ideas to remove these points which do not have the weakly textured property.

3.1. Improve WTD in Scale-Space. When giving a 2D signal f, the definition of the scale-space representation $L : R^2 \times R_+ \to R$ is the solution of the diffusion equation [24]:

$$\partial_t L = \frac{1}{2}\nabla^2 L = \frac{1}{2}\left(\partial_{xx} + \partial_{yy}\right) L. \quad (4)$$

It can be proved $L(\cdot, t) = g(\cdot, t) * f(\cdot)$, where $g : R^2 \times R_+ \to R$ is the Gaussian kernel; that is,

$$g(x, y; t) = \frac{1}{2\pi t} e^{-(x^2+y^2)/(2t)}. \quad (5)$$

For removing the point with strong texture feature, we should detect its texture strength in its neighbor region; here we use the gradient magnitude to represent it. Based on the *γ-parameterized* normalized derivative concept in the scale-space, we will consider the following differential expression:

$$L_{\text{grad}}\left(\cdot; \sqrt{t}\right) = t^{\gamma} \left| \left(\frac{\partial L(\cdot; t)}{\partial x}\right)^2 + \left(\frac{\partial L(\cdot; t)}{\partial y}\right)^2 \right|. \quad (6)$$

As shown in Figure 7 when using this operator to measure the points' texture strength, it is clear that L_{grad} will change along with the scale t. Moreover the more close the point is to the edge the bigger the change to L_{grad} magnitude it has (the cross-point "+" in each picture denotes the one more close to the edge).

3.2. Point Texture Strength Based on Local Maximum over Scales of Derivative. As above mentioned, the point texture strength can be reflected by L_{grad} magnitude over scales; hence we can define it by the local maximum over scales of derivative, and here we use following step function which is the close-form theoretical analysis for edge models, and it also can be convenient for further analysis to instead the actual edges. Consider

$$f(x) = \begin{cases} 0 & \text{for } x < x_0 \\ 1 & \text{for } x \geq x_0. \end{cases} \quad (7)$$

On the other hand, Gaussian kernel is a local kernel which responds to near neighbor of the input variables; hence, we here restrict the kernel size to the interval $[-3\sqrt{t}, 3\sqrt{t}]$. Consider

$$g(x; t) = \begin{cases} \dfrac{1}{2\pi t} e^{-x^2/(2t)} & x \in \left[-3\sqrt{t}, 3\sqrt{t}\right] \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

By substitution from (7) and (8) in (6), we obtain the L_{grad} magnitude of $f(x)$ with the following equation:

$$L(x, t) = \begin{cases} 0 & x - x_0 \leq -3\sqrt{t} \\ \displaystyle\int_{-3\sqrt{t}}^{x - x_0} g\left(x'; t\right) dx' & -3\sqrt{t} < x - x_0 < 3\sqrt{t} \\ \displaystyle\int_{-3\sqrt{t}}^{3\sqrt{t}} g\left(x'; t\right) dx' & x - x_0 \geq 3\sqrt{t}. \end{cases} \quad (9)$$

According to above equation, magnitude L_{grad} of $f(x)$ can be represented as the following expression:

$$L_x(x,t) = \begin{cases} g(x - x_0, t) & |x - x_0| < 3\sqrt{t} \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

where $|x - x_0|$ represents the distance between x and the discontinuous point x_0 which is the point on the edge, and we use the symbol Δl instead of it. This formula shows that when $\Delta l < 3\sqrt{t}$, $L_{\text{grad}} \neq 0$, and as mentioned above, the local maximum over scales of derivative will be used as the point's texture strength. But from Figure 7 it can be shown that there is no guarantee for getting the local maximum of the derivative over scales. And the derivative of $L_x(x,t)$ can be gotten from formula (10) as follows:

$$\frac{\partial L_x}{\partial t} = \begin{cases} \dfrac{1}{2\sqrt{2\pi}} t^{-5/2} e^{-\Delta l^2/2t} \left(\Delta l^2 - t\right) & \Delta l < 3\sqrt{t} \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

It is clear that the local maximum of L_x depends on scale t; that is, if $t_0 = \Delta l^2$, then $\partial L_x/\partial t|_{t=t_0} = 0$. Hence the range of scale $[1, t_{\max}]$ is difficult to select; for example, when $\Delta l = 20$, then $t_{\max}$ must satisfy $t_{\max} \geq 400$, and when $t_{\max}$ is set too large, it will increase in computation complexity for finding the local maximum of L_x. Obviously, we want to find the way to ignore the influence caused by Δl; that is, t_0 is not proportional to Δl^2.

For this reason we need to transform $L_x(x,t)$ to the function in which local maximum does not depend on Δl^2; let $L'_x = p(t)\cdot L_x$; then $\partial L'_x/\partial t = (\partial p(t)/\partial t)L_x + p(t)(\partial L_x/\partial t)$; that is, the derivative of $\partial L'_x/\partial t$ is

$$\frac{\partial L'_x}{\partial t} = \begin{cases} \dfrac{1}{2\sqrt{2\pi}} t^{-5/2} e^{-\Delta l^2/2t} \left(2p(t)\,t^2 + p(t)\left(\Delta l^2 - t\right)\right) \\ \qquad \Delta l < 3\sqrt{t} \\ 0 \quad \text{otherwise.} \end{cases} \tag{12}$$

For the purpose of reducing the incremental of $t_{\max}$ when Δl increases, hence $p(t)$ here should be the decreasing function of scale t. On the other hand, despite the fact that $t_{\max}$ should not be proportional to Δl^2, it has to guarantee that the local maximum position t_0 increases with increasing Δl; that is, it should satisfy

$$\left.\begin{aligned} \frac{\partial L'_x(\Delta l_1, t)}{\partial t} &= 0, \quad (t = t_1) \\ \frac{\partial L'_x(\Delta l_2, t)}{\partial t} &= 0, \quad (t = t_2) \end{aligned}\right\} \Longrightarrow t_1 > t_2. \tag{13}$$

For the constraint to $p(t)$, it can be assumed that $p(t) = \alpha - t^{\gamma}/\beta$, where $\alpha, \beta > 0$, and it also should satisfy $\log \alpha\beta > \gamma \log t$ (this constraint can guarantee that if $t \geq 1$, then $L'_x > 0$); hence, for the above mentioned when $t \geq 1$, then $p(t)$, $p'(t) \neq 0$; if we insert $p(t)$ into (13) and give $\gamma = 1$, then the local maximum position t_0 is given by

$$\begin{aligned} \frac{\partial L'_x}{\partial t} = 0 \Longrightarrow t_0 &= \frac{-\left(\Delta l^2 + \alpha\beta\right) + \sqrt{\left(\Delta l^2 + \alpha\beta\right)^2 + 4\alpha\beta\Delta l^2}}{2} \\ &= \frac{2\alpha\beta\Delta l^2}{\left(\Delta l^2 + \alpha\beta\right) + \sqrt{\left(\Delta l^2 + \alpha\beta\right)^2 + 4\alpha\beta\Delta l^2}} \\ &= \frac{2\alpha\beta}{\left(1 + \alpha\beta/\Delta l^2\right) + \sqrt{\left(1 + \alpha\beta/\Delta l^2\right)^2 + 4\left(\alpha\beta/\Delta l^2\right)}}. \end{aligned} \tag{14}$$

It is clear that when $\alpha\beta/\Delta l^2 \to 0$, then $t_0 \to \alpha\beta$; moreover it is an increasing function of Δl; that is, it has the same monotonicity property with one in original function $\partial L_x/\partial t|_{t=t_0} = 0$. Figure 8 shows the result improved by $L'_x(x,t)$.

3.3. Threshold Selection for Weakly Textured Points. A method for threshold selection to remove points which are not weakly textured points is needed, if their local maximum of texture strength is beyond this threshold.

On the other hand, there is no a priori information that can be used for removing the point which has no weakly textured property, and to illustrate the method for threshold selection, here, let us consider the distribution of weakly textured points detected by the method in Section 2.

Figure 9 shows the weak points detected by WTD and the 3D histogram of their distribution in the image. Here, the image is divided into $M \times N$ fragments; that is, if I is the image, then $I = \{R_1, R_2, \ldots, R_{M\times N}\}$; according to this distribution we find that regions have more concentrated distribution than others which are weakly textured regions. It also indicates that the local maximum texture strength of points distributed in these regions can be used to detect the threshold.

Through the above result, the properties of weakly textured region can be assumed as follows.

(i) The points which are detected by WTD in the region with weakly textured feature have more concentrated distribution than other regions.

(ii) The mean of local maximum texture strength in the weakly textured region is lower than that in other regions.

An important consequence of these properties is that it allows the threshold to vary with different images. We here use the Otsu threshold algorithm to segment the point sets detected by WTD into two classes, that is, the sets belonging to weakly textured region and the sets belonging to other regions. Let T_O be the threshold defined by Otsu threshold algorithm; $N(R_i)$ is the number of the points in fragment R_i (as shown above, the image is divided into $M \times N$ fragments), and according to property (i), if $N(R_i) \geq T_O$, then fragment R_i is selected as the part of weakly textured region which will be used for determining the threshold of texture strength later. On the other hand, when the weakly textured region is

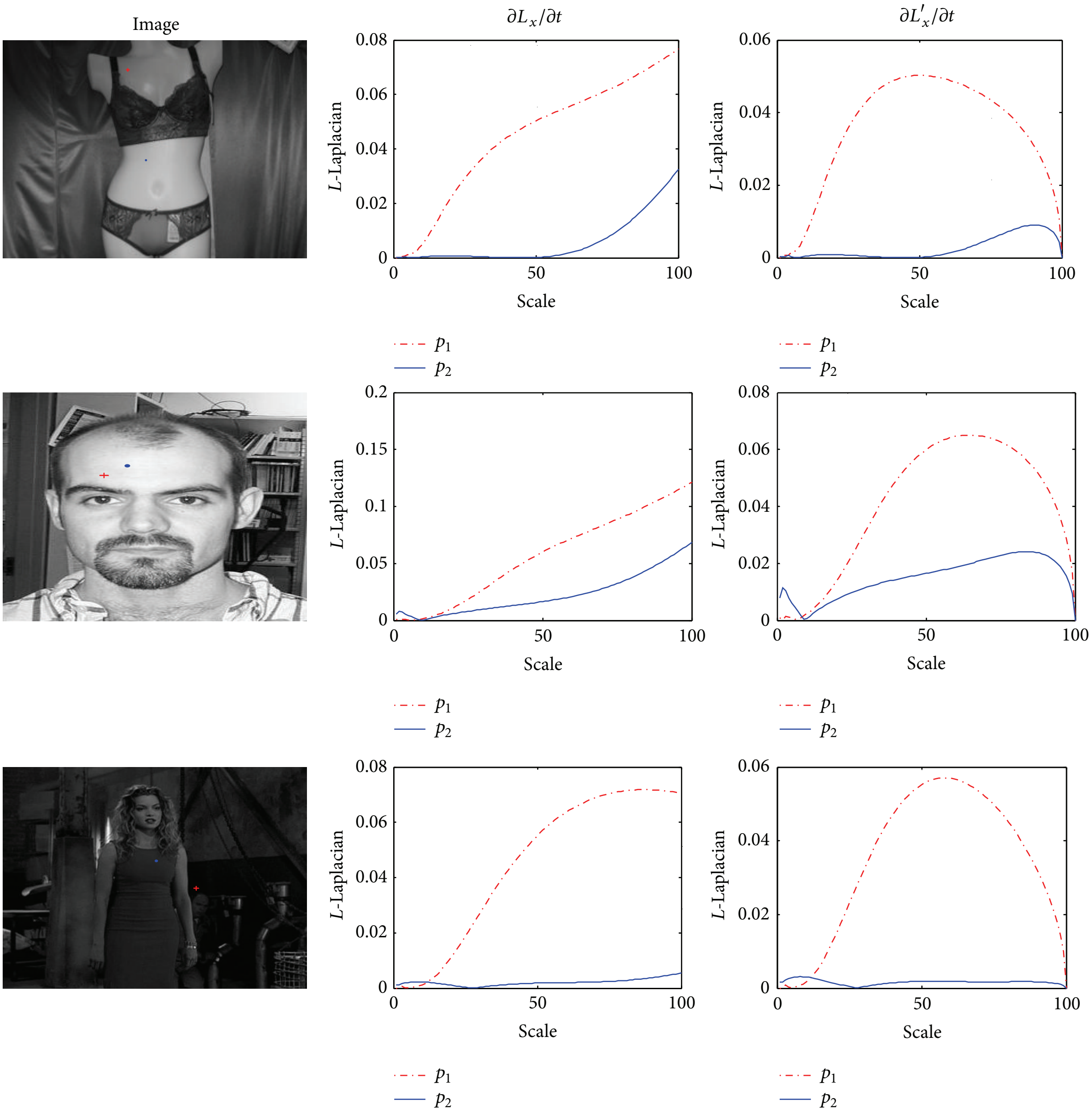


Figure 8: Results from improved texture measure method under different images.

determined, then according to property (ii), the threshold T_s can be defined as follows:

$$T_s = \min_i \left(\max_j \left\{ t\left(x_j\right) \mid x_j \in R_i \right\} \right), \qquad \left\{ R_i \in I \mid N\left(R_i\right) \geq T_O \right\}, \tag{15}$$

where $t(x)$ is texture strength which is detected by the local maximum of the derivative over scales at point $x \in R_i$. Since the points in the weakly textured region should be retained as much as possible, we here use $\max_j\{t(x_j) \mid x_j \in R_i\}$ as the subthreshold detected in region R_i; on the other hand, there also exist points which are not weakly textured points in some parts of the weakly textured region, so for removing these points the global threshold here can be obtained as $\min_i(\max_j\{t(x_j) \mid x_j \in R_i\})$.

3.4. Composed Algorithm. The mechanisms for weakly textured point detection and fine positioning were proposed, respectively. And they can be used in various ways in point detection and point removing. For the purpose of experiments and validation, we have here integrated these two modules into a composed algorithm, which can be expressed in the following four-step procedure.

Stage 1. Given a constant integer M, which is the number of points needed to be detected by WTD, M can be adjusted according to the size of image. Formulas (2) and (3) are then

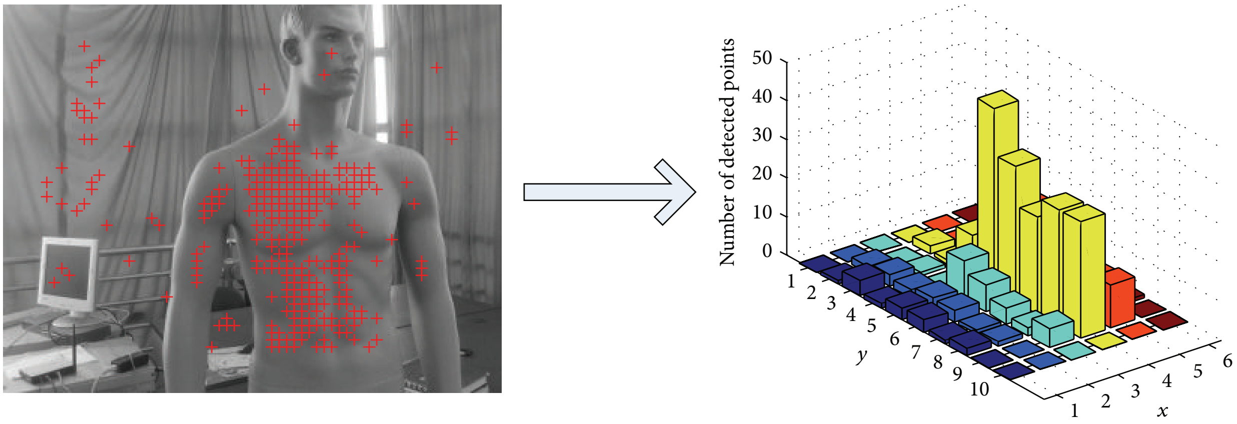


Figure 9: 3D histogram of weakly textured points distribution.

used to compute the vector in the radial-space $\{r_1, r_2, \ldots, r_N\}$ for each point p_i of image:

$$v_S(p_i) = \{S_{w\text{-texture}}(p_i, r_1), S_{w\text{-texture}}(p_i, r_2), \ldots, S_{w\text{-texture}}(p_i, r_N)\}. \quad (16)$$

Stage 2. Use the mechanism proposed in Section 2 to select the suitable radial r_{fit} for each point p_i of image I. Take $S_{w\text{-texture}}(p_i, r_{\text{fit}})$ as the similarity strength of p_i. And arrange p_i of image I in order of decreasing $S_{w\text{-texture}}(p_i, r_{\text{fit}})$. Then choose the first M points as the candidates of the weakly textured points.

Stage 3. Given a constant integer T, then scale t varies within $[1, 2, \ldots, T]$. Use $L'_x = p(t) \cdot L_x$ defined in formula (12) for computing the vector of each point which is obtained from Stage 2:

$$v_t(p_i) = \{L_{\text{grad}}(p_i, t_1), L_{\text{grad}}(p_i, t_2), \ldots, L_{\text{grad}}(p_i, t_T)\}, \quad t_i \in [1, 2, \ldots, T]. \quad (17)$$

Choose the biggest element from $v_t(p_i)$ as the texture strength of point p_i; that is,

$$v_{t\_\max}(p_i) = \max_{t_i \in [1,2,\ldots,T]} L_{\text{grad}}(p_i, t_i). \quad (18)$$

Stage 4. Using the threshold selection mechanism to detect the threshold T_s of texture strength and remove the points by T_s, that is, if $v_{t\_\max}(p_i) < T_s$, then p_i is the weakly textured point.

Figure 10 shows the performance of above composed algorithm; it is clear that this algorithm can efficiently extract the weakly textured points, and through the scale-space texture strength concept, the points have strong textureness which can be removed completely.

4. Sparse Representation for Point Correspondence

In this section, we study the feature matching problem of the weakly textured points detected by the proposed mechanism in the above two sections. Since the resulting feature matches can be used in wide-baseline stereo matching problem, it is important to find a matching algorithm for weakly textured points correspondence. On the other hand, since the point has the feature of weak texture, it is difficult to choose features for distinguishing different weakly textured points. Traditionally, methods of point-matching are based on the Euclidean distance; namely, if the distance between the candidate point and the object point is minimum, then those points are correspondence; moreover, the algorithms such as support vector machine (SVM) [25] and nearest neighbor algorithm [26] are largely dependent on the choice of features. Hence, we here match the feature via sparse representation, within the proposed framework; the precise choice of feature space is no longer critical. In [19], the algorithm used for human face recognition is similar to our method. And the efficacy of sparse representation in solving classification problems has been demonstrated.

4.1. Analyze the Feature of Weakly Textured Points. Since features of the same point in two different images are almost equal, the feature of the point in the first image can be thought as lying on the linear subspace composed by all features of weakly textured points in the second image. As proposed in [19], we here use the same hypothesis; that is, the feature from one point lies on a subspace. This is the only prior knowledge about our feature matching algorithm.

Given the feature set of points $S_2 = [s_{2,1}, s_{2,2}, \ldots, s_{2,n_2}] \in \mathfrak{R}^{m \times n_2}$ in the second image, where $s_{2,j}$ denotes feature of point p_j, if p_j exists in the first image, then its feature $s_{1,j}$ will be approximately in the linear span of S_2 associated with object j:

$$s_{1,j} = \alpha_1 s_{2,1} + \alpha_2 s_{2,2} + \cdots + \alpha_{n_2} s_{2,n_2}, \quad (19)$$

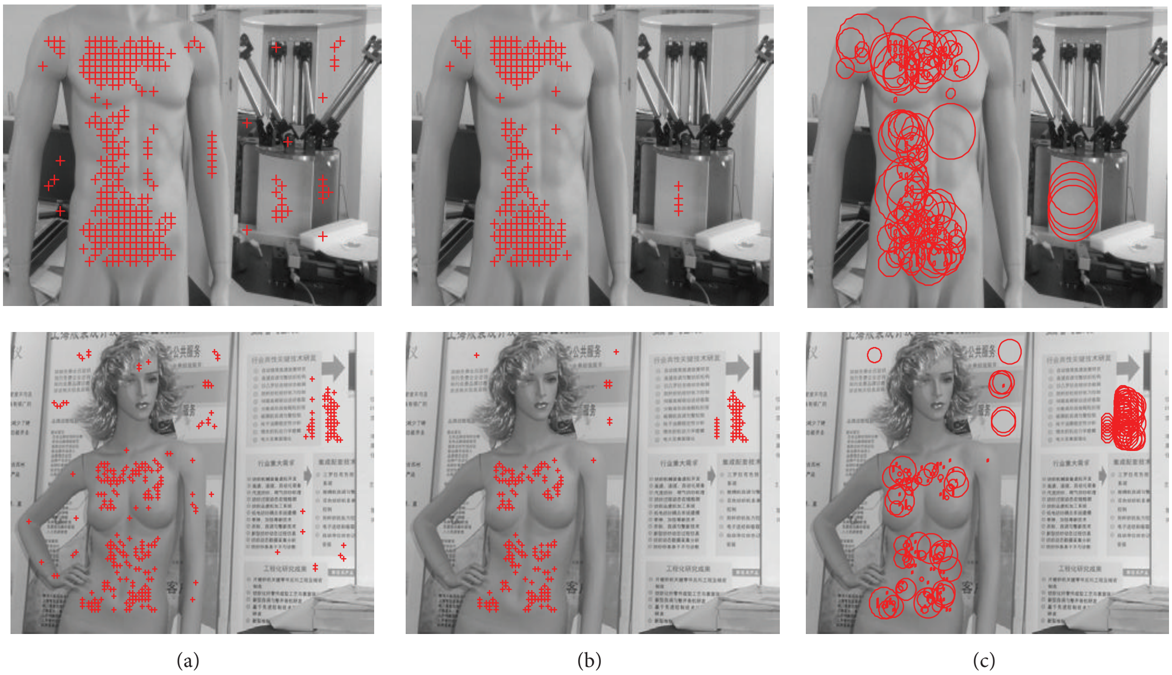


Figure 10: Results from threshold mechanism for different images. (a) Points detected by WTD. (b) Points detected by WTD + threshold mechanism. (c) Regions belong to detected points.

where $\alpha_i \in \Re$, $j = 1, 2, \ldots, n_2$. The linear representation $s_{1,j}$ also can be rewritten in terms of S_2 as

$$s_{1,j} = S_2 x \in \Re^m. \quad (20)$$

Since the point p_j existed in the two images, its feature in these two images should be closed enough. Hence, the coefficient vector x could be assumed as $x = [0, \ldots, 0, \alpha_j, 0, \ldots, 0]^T \in \Re^{n_2}$ where its entries are zero except those associated with jth point. And here we should solve the linear equation $s_{1,j} = S_2 x$; here $S_2 \in \Re^{m \times n_2}$, and n_2 is the number of the weakly textured points in the second image, and m is the dimension of the features. Here in order to reduce the computational complexity, the dimension of features should be chosen as small as possible; that is, it has $n_2 \gg m$, so the equation $s_{1,j} = S_2 x$ is underdetermined, and its solution is not unique. On the other hand, since the ideal solution x_0 of $s_{1,j} = S_2 x$ should be dense, as assumed above. So the denser x_0 is, the easier will it be to determine point p_j in the first image.

Through the above analysis, we here seek the sparsest solution to $s_{1,j} = S_2 x$, that is, solving the following optimization problem:

$$x_0 = \arg\min \|x\|_0 \quad \text{subject to } s_{1,j} = S_2 x, \quad (21)$$

where $\|\cdot\|_0$ denotes the ℓ_0-norm, which counts the number of nonzero entries in a vector. However, this problem is NP-hard and even difficult to approximate [27]: there are no more efficient procedures for finding the sparest solution than exhausting all subsets of the entries for x. But in the theory of sparse representation and compressed sensing [28], it is proved that if the solution of ℓ_0-minimization is sparse enough, then it is equal to the solution of the ℓ_1-minimization problem:

$$x_0 = \arg\min \|x\|_1 \quad \text{subject to } s_{1,j} = S_2 x. \quad (22)$$

We here use a simple example for interpretation of the reason why should we use the ℓ_1-minimization rather than ℓ_2-minimization to get the sparest solution. As shown in Figure 11, the left and right images show the geometry of ℓ_2-minimization algorithm and ℓ_1-minimization algorithm to solve $s_{1,j} = S_2 x_0$ under two-dimensional situation, respectively. According to Figure 11 the solution of ℓ_1-minimization is also the solution of ℓ_0-minimization; however, the one gotten from ℓ_2-minimization does not satisfy ℓ_0-minimization.

4.2. Feature Matching Based on Sparse Representation. So far, we have used the simple example for interpreting the sparse representation theory and before we give the algorithm, all the symbols used latter should be defined as follows.

I_1: the image having the points which should be matched.

I_2: the image having the reference points for those in image I_1.

$\{q_i\}_{I_1}, \{p_i\}_{I_2}$: the weakly textured points sets, which are detected in the I_1, I_2, respectively.

$F(\cdot) : \Re^2 \rightarrow \Re^m$: feature extraction function as the position of point; that is, it can extract the m-dimension feature for the point.

$\delta_i(\cdot)$: $\Re^n \rightarrow \Re^n$: the function that can get a new vector from the vector $x \in \Re^n$, and the new one

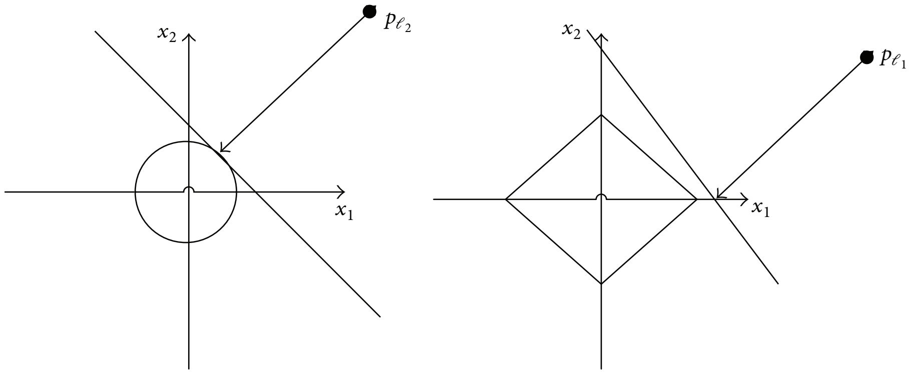


Figure 11: Geometry of sparse representation with ℓ_2-minimization and ℓ_1-minimization.

has the only nonzero entries in $x \in \mathfrak{R}^n$ which are associated with index i; moreover because the point in image I_1 can only have one corresponding point, so here the new vector gotten from $\delta_i(\cdot)$ should have only one nonzero entry.

Sparse Representation-Based Feature Matching Algorithm

(1) Input the feature set of reference points in image I_2. That is, $S_2 = \{F(p_1), F(p_2), \dots, F(p_n)\} \in \mathfrak{R}^{m\times n_2}$ which is regarded as the matrix for ℓ_1-minimization problem. And the feature of test point $F(q)q \in I_1$, which can be regarded as a test sample.

(2) Normalize the columns of S_2 and $F(q)$ to have unit ℓ^2-norm; here $\overline{S}_2$ and $\overline{F}(q)$ are the normalized S_2 and $F(q)$, respectively.

(3) Solve the following ℓ_1-minimization problems:

$$x_0 = \arg\min \|x\|_1 \quad \text{subject to } \overline{F}(q) = \overline{S}_2 x \tag{23}$$

or

$$x_0 = \arg\min \|x\|_1 \quad \text{subject to } \left\|\overline{S}_2 x - \overline{F}(q)\right\|_2 \leq \varepsilon, \tag{24}$$

where if $\overline{F}(q)$ have noise term, and the bounded energy of this noise is less than ε, then we will use the extended ℓ_1-minimization problem [19].

(4) Compute the residuals $r_i(\overline{F}(q)) = \|\overline{F}(q) - \overline{S}_2\delta_i(x_0)\|_2$ for $i = 1, \dots, n$.

(5) Output identity $(\overline{F}(q)) = \arg\min_i r_i(\overline{F}(q))$, then the position of corresponding point can be determined by this result.

4.3. Feature Extraction for Weakly Textured Point. As above shown, we have proposed the feature matching algorithm for the weakly textured points based on the sparse representation classifier, but there still remains a problem, that is, what feature should be extracted. As assumed above the point feature in image I_1 should have the sparest solution in the feature space of image I_2; this means that if the feature satisfies this sparsity condition then it can be used for matching weakly textured points, that is, as in [19]: "the choice of features is no longer critical." On the other hand, this condition is satisfied by the feature of texture points, because these points have their own structure in their neighbor regions, and this feature cannot be represented by other points. However, the weakly textured points might not satisfy this condition, for the reason that these points usually have no obvious characteristics. So, in the experiment part we will use the different descriptors including LBP, random, and downsampled features to investigate their performance to the proposed matching algorithm. Here the random feature can be expressed as [29]

$$x_0 = \arg\min \|x\|_1 \quad \text{subject to } \hat{y} = RSx, \tag{25}$$

where $R \in \mathfrak{R}^{d\times m}$ is a random matrix independently sampled from zero-mean normal distribution; moreover each row of R is normalized to unit length, and $\hat{y} = Ry \in \mathfrak{R}^d$, when $d \ll m$; then using this feature can reduce the computation complex, and the polytope geometry as the analysis indicates the following: if the solution x_0 is sparse enough, then it can be correctly recovered by ℓ_1-minimization from any sufficiently large number d of linear measurements $\hat{y} = RSx$ with the probability:

$$d \geq 2t \log\left(\frac{n}{d}\right), \tag{26}$$

where t is the number of nonzero entries in vector x.

For ending this section, we will give the method to decide the validation of the sample. In the real-world situation, when a point is detected in image I_1 might not be detected in image I_2 it means that the feature of this point could not be described by using only one point feature in image I_2; that is, the nonzero entries of its sparse representation in image I_2 are not concentrated on one subject; in contrary, it has widely spread sparse coefficients among multiple points in image I_2; hence, the sparsity concentration index (SCI) [19] to determine the validation of the sample is as follows:

$$\mathrm{SCI}(x) = \frac{k \cdot \max \|\delta_i(x)\|_1 / \|x\|_1 - 1}{k - 1} \in [0, 1]. \tag{27}$$

And according to this index then the valid point p can be decided if $\mathrm{SCI}(p) \geq \tau$, where $\tau \in (0, 1)$ is a threshold.

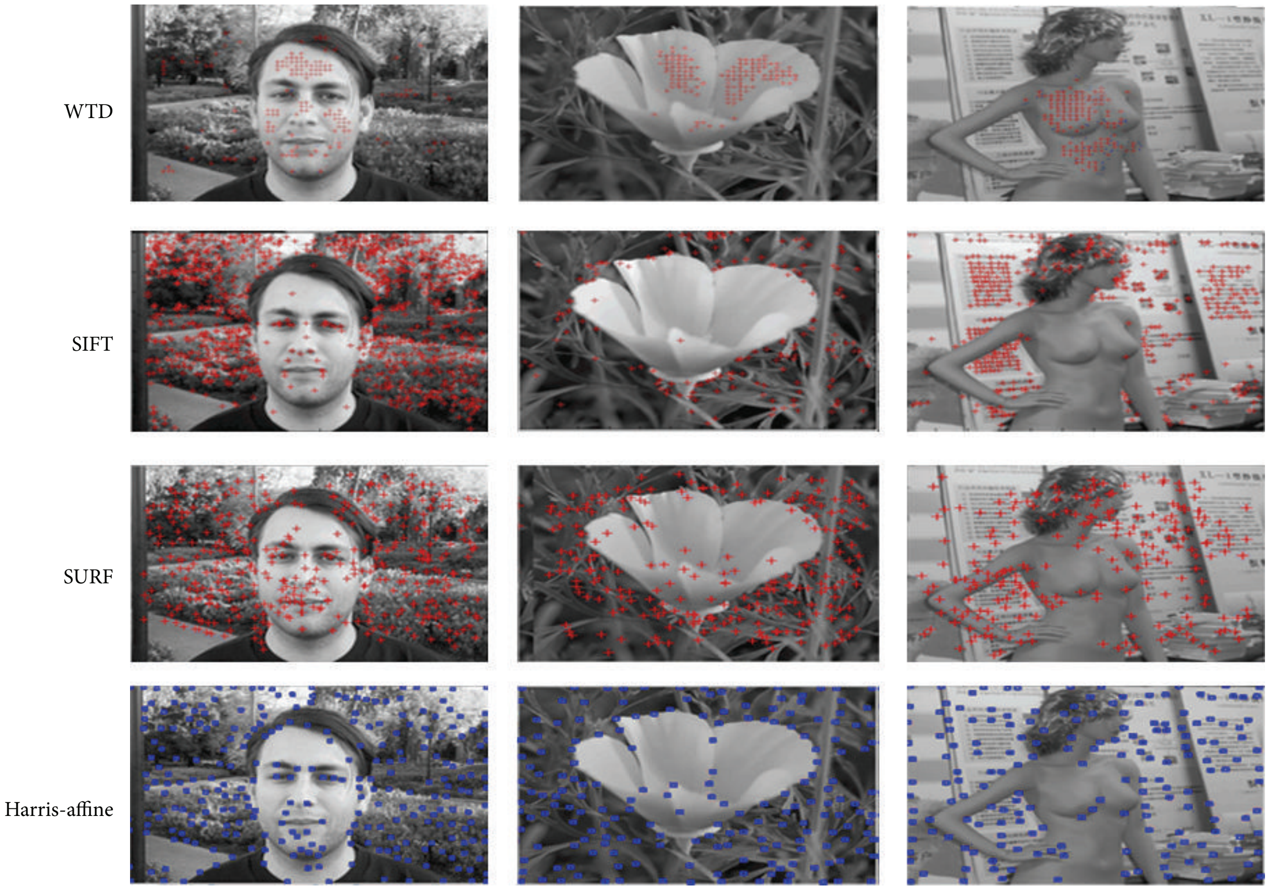


Figure 12: Examples of detected points distributed on different objects with different detectors.

Table 1: Performance under different objects.

Test images	Harris-affine		SIFT		SURF		WTD	
	OP/TP	OP	OP/TP	OP	OP/TP	OP	OP/TP	OP
Human face	0.1187	28.0000	0.05464	32.1429	0.1255	46.4286	0.5841	85.2857
Flower	0.1837	28.2857	0.0983	26.1428	0.2387	50.5714	0.7253	107.1429
Human model	0.1315	22.8571	0.0122	5.0000	0.1629	36.2857	0.8240	113.0000
Average	0.1446	26.38	0.05504	21.09523	0.1757	44.4285	0.7111	101.8095

5. Experiment Results

In this section, we present experiments on publicly available databases for weakly textured points detecting and the feature matching algorithm based on sparse representation, and the efficacy of these two algorithms can be demonstrated by these experiments. The experiments can be divided into three parts, that is, points detection, feature matching, and promoting the quasi-dense matching algorithm [20]; in the first part we will examine the performance of the detection algorithm, comparing with different images in various situations and comparing to several popular feature detectors (as shown in Sections 5.1 and 5.2). In the second part we will examine feature matching algorithm, comparing performance across various features spaces and feature dimensions and comparing to several popular classifiers in Section 5.3. Finally, we will demonstrate the proposed method for improving the quasi-dense matching algorithm in Section 5.4.

5.1. WTD with Image Perturbations. We here measure the WTD efficiency for weakly textured points detection under images with different objects and the robustness across with the same object has different backgrounds. The data we choose here include 200 images from Caltech human faces set, 102 images from category flower dataset, and 100 images from human model. And first we will compare our detector with three classical detectors in weakly textured points detection, namely, Harris-affine, SIFT, and SURF. For measuring the performance here we use the proportion between the number of detected points (Figure 12) on the objects (OP) and total number of detected points (TP) and set the radial of WTD from $R = 2$ to $R = 32$. Table 1 compares WTD to the other three detectors.

Table 2: Performance under different objects with different backgrounds.

Test images		Human face	Flower	Human body	Average
Harris-affine	RB	0.1242	0.0758	0.0380	0.0794
	FB	0.1111	0.0765	0.0234	0.0703
	TB	0.1228	0.0556	0.0208	0.0664
SIFT	RB	0.0778	0.0059	0.0009	0.0282
	FB	0.0281	0.002	0.0042	0.0114
	TB	0.4242	0.0136	0.0167	0.1516
SURF	RB	0.1925	0.1843	0.2135	0.1968
	FB	0.2131	0.1808	0.1566	0.1835
	TB	0.2157	0.1767	0.0948	0.1624
WTF	RB	0.8750	1	1	0.9583
	FB	0.9123	1	1	0.9708
	TB	0.8361	1	1	0.9454

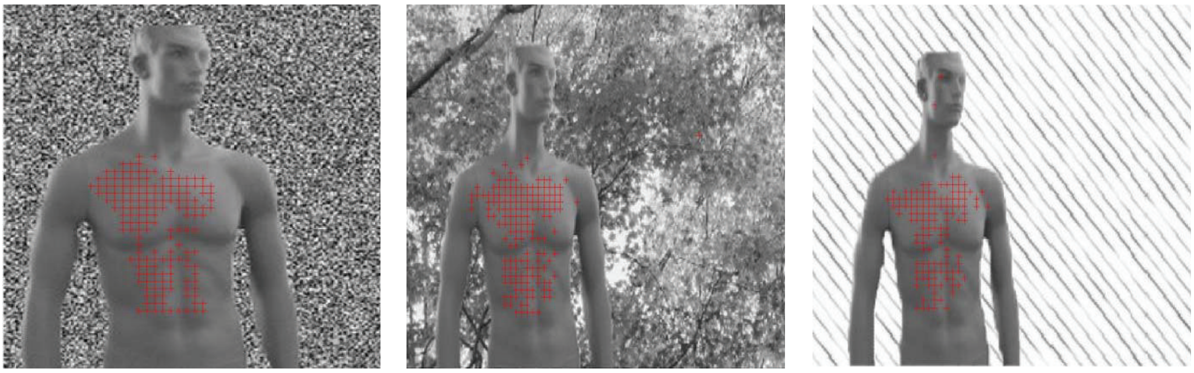


Figure 13: Performance of WTD on different backgrounds.

Based on the results on the data, we draw the following conclusion.

(1) For all data, the performances of WTD consistently exceed the other three detectors in weakly textured detection. It means that if the object has large region of weak texture, then WTD can detect more points in this object than other detectors.

(2) The results obtained by WTD and classical detectors are complementary; namely, the points which are not detected by classical detectors can be detected by WTD, so if points detected by those classical detectors are not enough for the stereo reconstruction, then WTD can be used as a complement.

As a second set of this experiment, the robustness of WTD for the same object with three different texture backgrounds is tested. In order to demonstrate the performance of WTD, we here use artificial backgrounds as the extreme situations, namely, random background (RB), forest background (FB), and texture background (TB), respectively. And the parameters of WTD are the same as in the above experiment. Table 2 and Figure 13 show the performance of WTD comparing to classical detectors.

It is clear that the points detected by WTD are almost concentrated on the object with weak texture (human model) no matter what texture background takes (the WTP/TP close to 1). On the other hand, this property of WTD is contrary to classical detectors, whose detected points are almost concentrated on the texture objects, and it is once again demonstrating the relationship between WTD and classical detectors as the above experiment.

5.2. Performance under Varying Blur, Lighting Change, Rotating, and Viewpoint Change. For this experiment, we test the repeatability of four detectors under different photometric and geometric transformations, as the protocol suggested in [13]. Test images sets and results are shown in Figures 14–17 and each set changes in 6 levels. Because of space constraints, we here only give three of the six images in each type of transformation, that is, first, third, and sixth. Each figure presents the repetition rate which includes three parts, namely, total number of matched points, the number of matched points on the objects, and the matching score on the objects. The range of radius scale is set from $R = 2$ to $R = 30$, and the repetition rate here we just use the region matches, and the matching conditions follow as recommended in [22]. The matching score S_i, here, is given by the proportion of

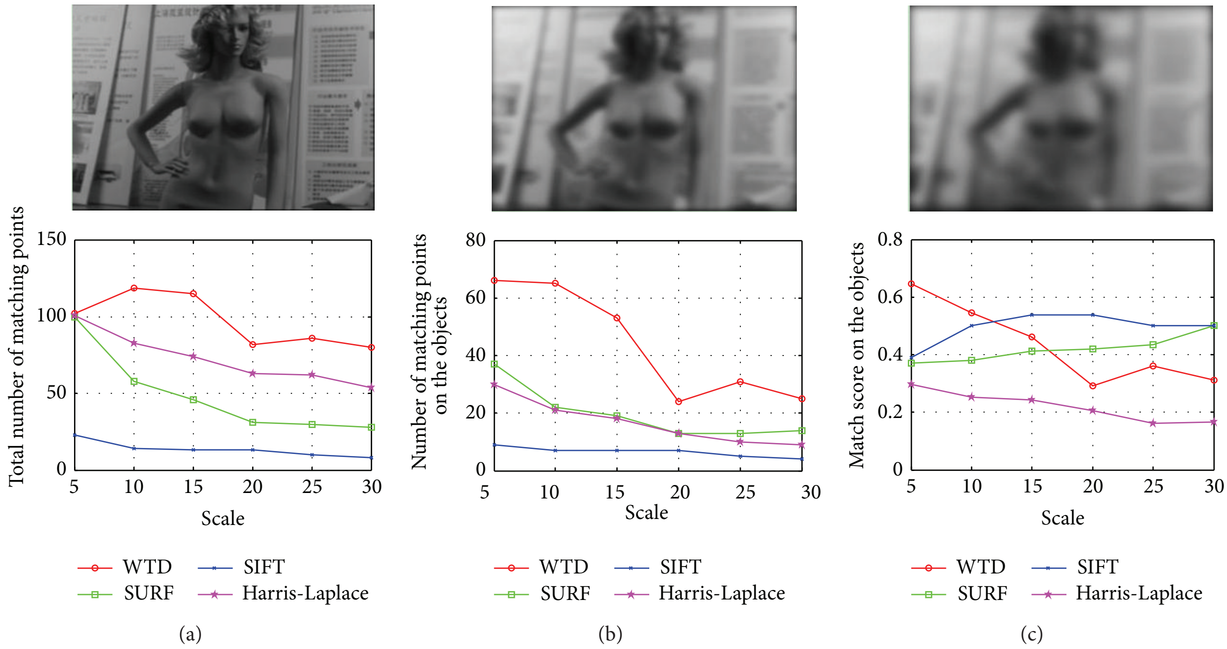


Figure 14: Blur for image. (a) Total number of matched points for image blurs. (b) Number of matched points on the objects. (c) Matching score on the objects.

correspondences with detected regions on the objects for all correspondences in the test image; that is,

$$S_i = \frac{\text{Total number of matches}}{\text{Total number of detected regions}} = \frac{N_i}{N}, \quad (i = 1, 2, \ldots, M). \tag{28}$$

5.2.1. Blur. Figure 14 shows the results detected by four detectors undergoing increasing amounts of image blur, and we here use the Gaussian kernel to blur the image, and the scale parameter t (scale selection) is set as $t = 0, 10, \ldots, 60$; according to the results of all detectors, the number of matching points declined with the blur increasing; on the one hand, the reason for classical detectors is the image texture reduced by image blur. On the other hand, the number of match points detectors by our detector should increase with the blur, but in fact the threshold selection mechanism discussed in the previous section would get the small threshold because the blur causes the number of candidate regions increasing, so according to formula (15), the threshold will become smaller than before; hence the number of points detected by WTD will decline.

5.2.2. Lighting Changes. Figure 15 shows the results for lighting changes with four different detectors. From this result we can find the following: (a) the total number of match points gotten by WTD is lowest and (b) the match points on the object (human face) are higher than other detectors. This is caused by following reasons.

(1) The background of image chosen by us is clatter, so the total number of match points chosen by classical detectors is higher than WTD.

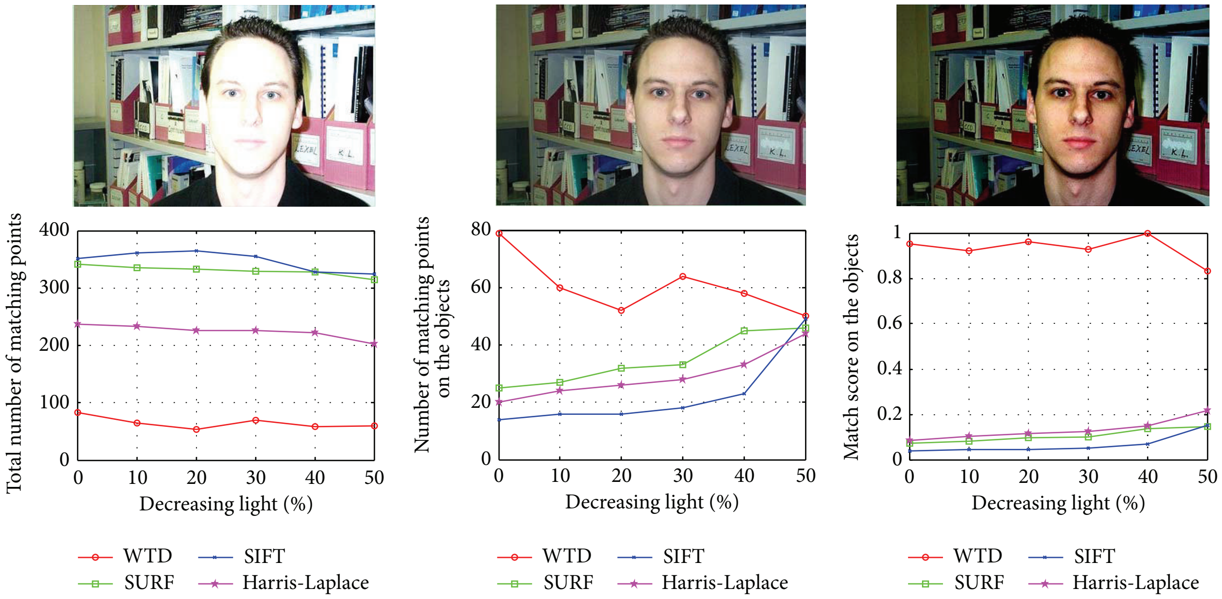
(2) When the intensity decreases, then the texture of background will increase, and the texture in the object is also increased with light intensity decreasing, but it is slower than the increase in the background; hence, the number of match points on the object by WTD is staying close to its mean value.

(3) The points detected by WTD are almost concentrated on the object, so its match score is large than others.

5.2.3. Rotation. Figure 16 shows the performance for image rotation; the number of match points on the object detected by WTD is not stable with the image rotation. This is because the object scale here will vary with the rotating process. And the same radial (see Section 2) used to measure the same weakly textured point in these images will get the different values. On the other hand, the backgrounds of these images are also textureless, and if the scale changed the threshold for reducing the texture points will also change; hence, points detected by WTD will concentrate on the background. Although being in such serious situation, the match score of WTD is also higher than other detectors.

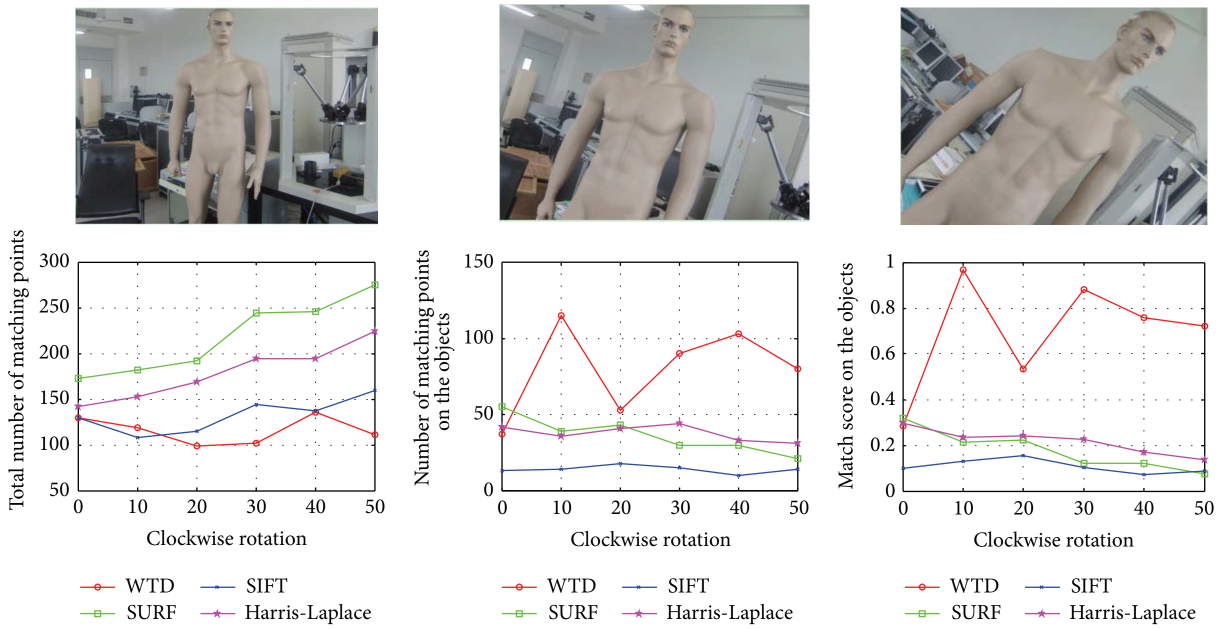
5.2.4. Viewpoint Changes. Figure 17 shows the performance for viewpoint changes. It is clear that when viewpoint changes the matching score of WTD is always close to 1. This implicates that the matching points gotten from WTD are almost concentrating on the object with the viewpoint changes in 15° degree limitation.

5.3. Feature Matching Algorithm Experiment. In this part, we will test the performance of matching algorithm based on sparse representation with the detected weakly textured



(a) (b) (c)

Figure 15: Lighting changes for image. (a) Total number of matched points for lighting change. (b) Number of matched points on the objects. (c) Matching score on the objects.



(a) (b) (c)

Figure 16: Rotation for image. (a) Total number of matched points for rotation. (b) Number of matched points on the objects. (c) Matching score on the objects.

points. We here compare performance across various feature spaces and feature dimensions with two popular classifiers, namely, linear SVM and ℓ^2-min. Moreover, we will also test our matching algorithm under random pixel corruption and random block occlusion, respectively.

5.3.1. Sparse Representation Based Weakly Textured Point Matching. We match 150 weakly textured points detected by WTD in each image as shown in Figure 18. Here we only test the matching algorithm under the WTD, and we use conventional features LBP (local binary pattern) and two unconventional features: random and downsampled region features. The window size of LBP is set to be 30 × 30 pixels, and we compute the matching rate with the feature space dimensions $10, 20, \ldots, 60$. The dimension of random feature space discussed in the previous section is set the same as

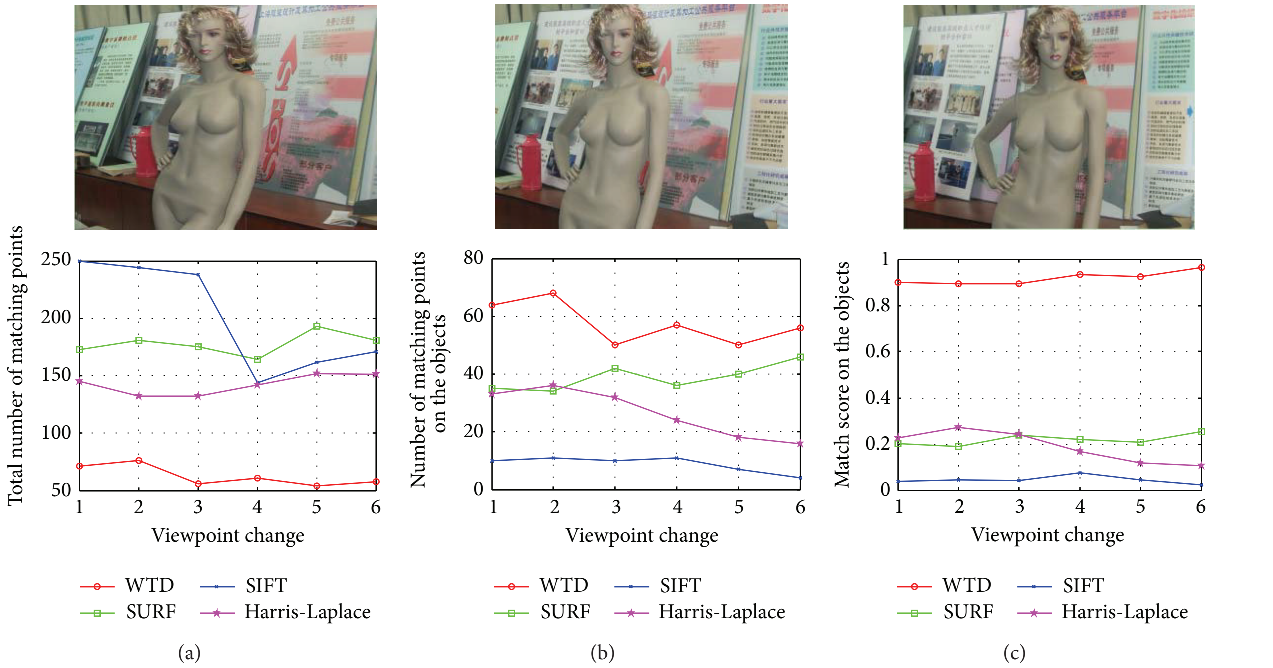


Figure 17: Viewpoint change for image. (a) Total number of matched points for viewpoint change. (b) Number of matched points on the objects. (c) Matching score on the objects.

Table 3: Variances of matching rate.

Algorithm	LBP	Downsample	Random
SRM	1.15×10^{-2}	9.8×10^{-3}	7.9540×10^{-4}
ℓ^2-min	2.47×10^{-2}	7.2×10^{-3}	5.6×10^{-3}
SVM	2.63×10^{-2}	4.29×10^{-2}	2.63×10^{-2}

LBP, namely, the random matrix $R^{d\times 60}$ ($d = 10, 20, \ldots, 60$); finally the window size of downsampled feature is set to be 30×30 with dimensions 60, 45, 30, and 12. Those numbers correspond to downsampling ratios of 1/15, 1/20, 1/30, and 1/75, respectively. Figure 18 shows the match performance for the various features, in conjunction with three different algorithms: SRM, ℓ^2-min, and SVM.

Figure 18 also shows results under LBP descriptor with 40 dimensions. The algorithm returns enough matched points on the low texture region that lets us get more sufficient points than traditional sparse matching on these area.

Based on the results on different feature spaces, the following conclusion can be drawn.

(1) Here we use the variance of feature dimensions to measure the stability of matching rate on different feature spaces. Table 3 shows the variances of matching rate for SRM, ℓ^2-min, and SVM. It is clear that SRM is more stable than others for feature dimensions on the LBP and random spaces.

(2) The biggest matching rate of SRM exceeds the best performances of others on the LBP and downsample spaces. More specifically, the best performance for SRM on the LBP is 80.86%, compared to 71.35% for ℓ^2-min and 69.70% for SVM. The best rate for SRM on the downsample is 80.38%, compared to 59.64% for ℓ^2-min and 55.97% for SVM.

(3) The results on the LBP and random spaces suggest that when the feature dimension is 40, it is sufficed for sparse recovery. Moreover when the dimensions are beyond 40, the performances on these two feature spaces will converge.

5.3.2. Matching despite Random Pixel Corruption. In this experiment, we test the robustness of SRM under the pixel of description region occluded by random noise. We use the extended ℓ_1-minimization problem [19] at the third step of SRM for solving this occlusion problem. Here the error tolerance ε is set to be the bounded energy of the random noise. In order to eliminate the influence caused by WTD, we use the random noise to corrupt the located region in which the feature is extracted (see Figure 19). On the other hand, since there are no efficient descriptions for weakly textured point, the description we used is LBP, because it has the best performance as shown above. For demonstrating the performance of SRM, we compare it to the following algorithms: $\ell^2_{\min}+\text{PCA}$, $\ell^2_{\min}+\text{LNMF}$, and $\ell^2_{\min}+\text{ICA}$, respectively. Here PCA, LNMP, and ICA are principal component analysis, local nonnegative matrix factorization (LNMF), and independent component analysis (ICA), respectively. These algorithm are used for feature preprocessing and the ℓ^2-min we used here because it has the second highest matching rate in the above experiment. According to the experiment result, it can be seen that when the corruption is up to 25 percent the number of matching points with each algorithm will drop to zero. So, here we only show the result with corruption from 0 percent to 25 percent. According to the performance of

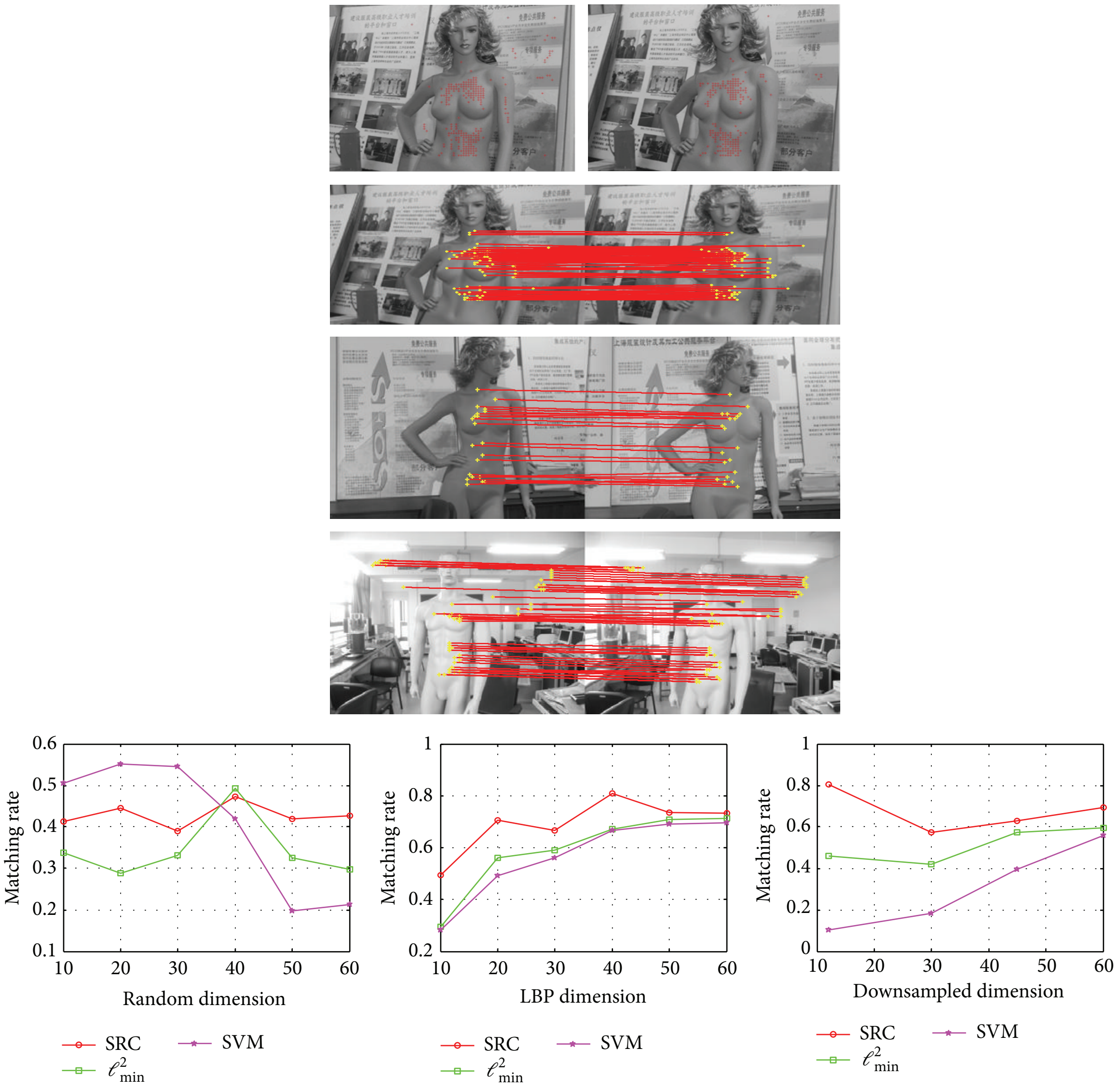


Figure 18: Matching rate for various feature transformations.

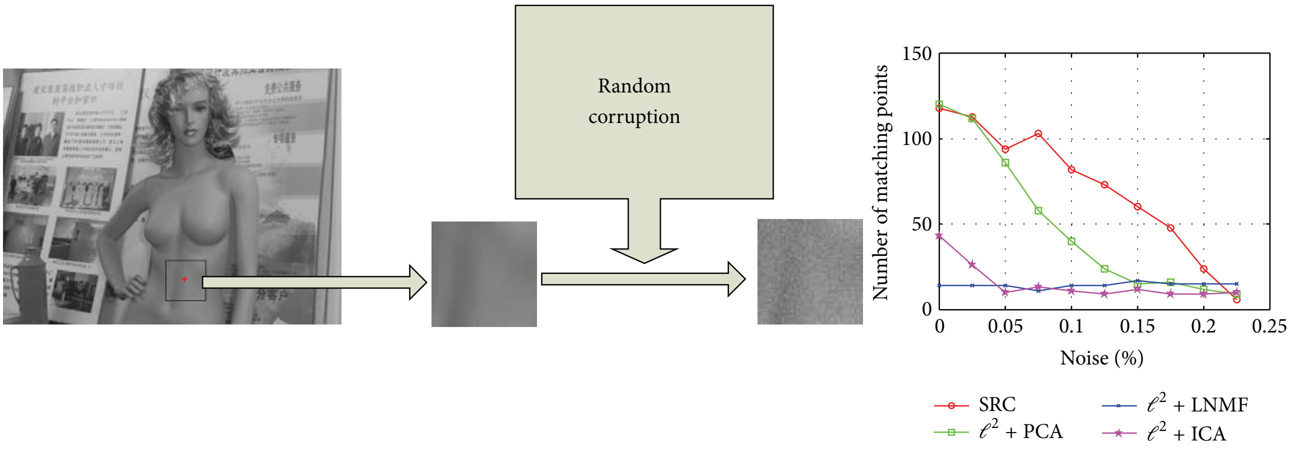


Figure 19: Recognition under random corruption.

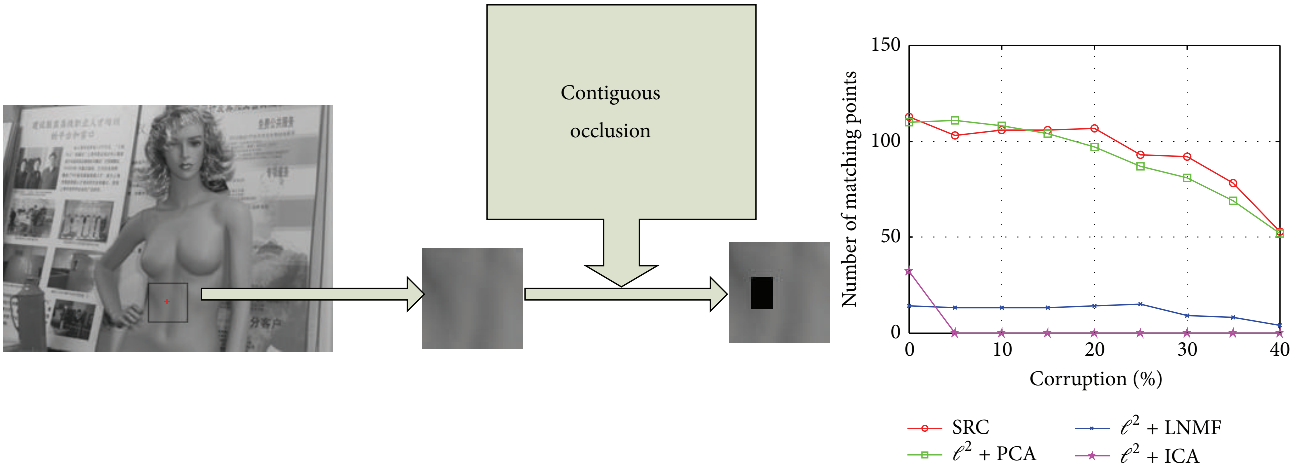


Figure 20: Recognition under varying level of contiguous occlusion.

SRM and its three competitors, we see that SRM outperforms others; namely, from 5 percent up to 20 percent occlusion, the number of matching weakly textured points are higher than others. At 10 percent corruption, the highest matching number of others is 40 points, while the one gotten by SRM 80 points. Even at 15 percent occlusion, the matching number is still 60 points. Clearly, the SRM can ensure to tolerate under the corruption less than 25 percent for matching the weakly textured points. On the other hand, the matching points numbers gotten by $\ell^2_{\min}$ + LNMF and $\ell^2_{\min}$ + ICA are obviously less than SRM and $\ell^2_{\min}$ + PCA; it has been suggested that both LNMF and ICA are not suitable for the preprocessing of weakly textured feature.

5.3.3. Matching despite Random Block Occlusion. The final part of the expe riment here is to simulate various levels of contiguous occlusion, from 0 percent to 40 percent, by replacing a randomly located square block of each feature region with the gray-lever at zero, and the reason why we choose zero gray-lever block is that the matching points are weakly textured point, so if we choose the texture block here, the characteristic of these points will be replaced by the features in this texture block. And the region here is the same as the discussion in the above experiment (see Figure 20). The location of occlusion is randomly chosen for each image and is unknown to the computer. The results got from SRM and $\ell^2_{\min}$ + PCA are still better than others. Again, the above implication is also true for this case. Despite the fact that the performances of SRM and $\ell^2_{\min}$ + PCA are closed, it is clear that SRM outperforms $\ell^2_{\min}$ + PCA. For example, at 20 percent occlusion, SRM achieves 109 points; it is higher than $\ell^2_{\min}$ + PCA's 98 points. On the other hand, the reason why the number of matching points is not decreased to zero when occlusion increases to 40 percent is that the located square block we used here is the zero gray-lever images. And the matching points also have weakly textured characteristic, so it will not be much affected by this square block.

5.4. Improved Quasi-Dense Matching. In this section, we use WTD to promote the quasi-dense matching algorithm [20]. The quasi-dense matching algorithm starts from a set of sparse seed matches which were usually obtained by classical detectors, then propagates to the neighboring pixels by the best-first strategy, and finally produces a quasi-dense disparity map. Since most initial sparse seed matches distribute in the strong texture regions, it nearly has no seed matches in the textureless areas; furthermore if the matches got from propagating step in these areas are wrong, gross reconstruction errors will occur, so here we use the proposed approach to get the initial sparse seed matches in the large textureless areas.

Table 4: Comparison between original dense matching and improved one under different circumstances.

Change	Algorithm	#corresponding-point on untex
Lighting	Quasi + ours	845
	Quasi	414
Image scale	Quasi + ours	490
	Quasi	210
Camera angle	Quasi + ours	676
	Quasi	351

The original algorithm is independently implemented for comparison on the same image pairs. The corresponding-point sets obtained by WTD and traditional quasi-dense matching algorithm are shown in Figure 21. One can verify the effectiveness of our implementation by corresponding quality measures which are given in Table 4.

Our algorithm is evaluated on the changes of lighting, scale, and camera angle, respectively, and the results demonstrate that the algorithm performs particularly better on the large textureless object surface than the original one (as shown in Figure 21).

6. Conclusion

In this paper, we proposed an efficient detector for weakly textured points and used sparse representation for matching the detected weakly textured points in two different images.

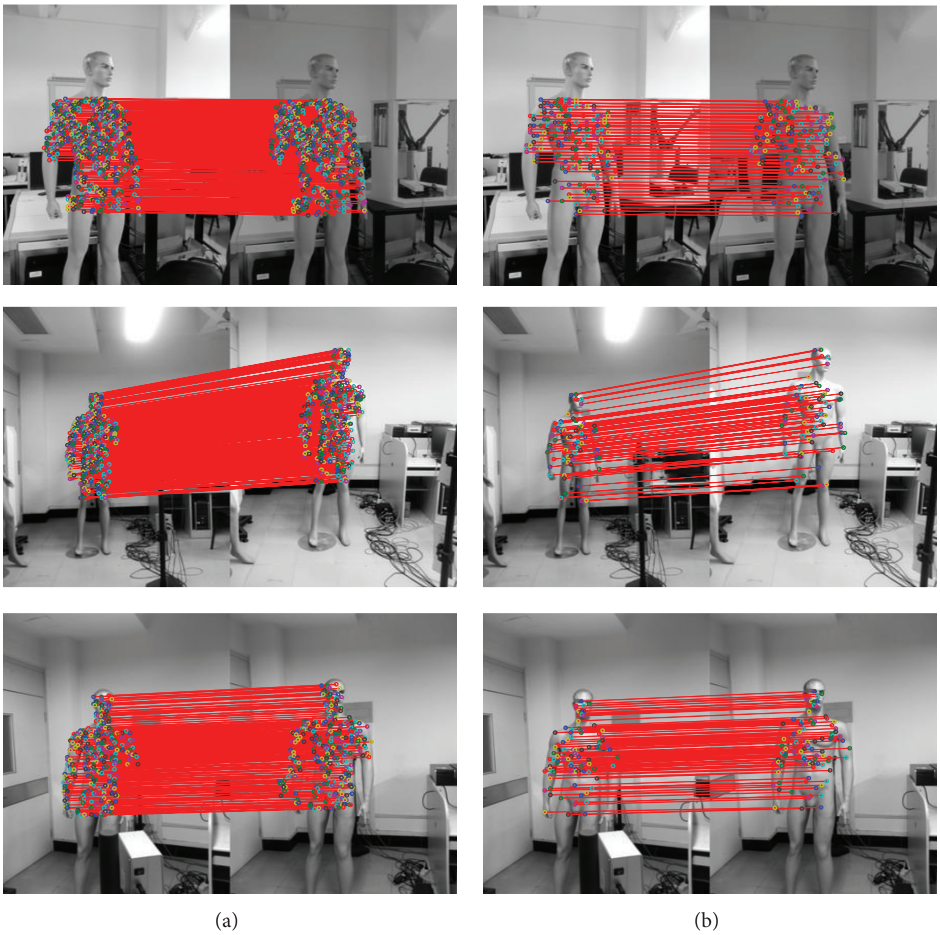


Figure 21: Results from improved dense matching and original one under different circumstances. Top row: corresponding-points matched with lighting changes. Middle row: corresponding-points matched with scale changes. Bottom row: corresponding-points matched with camera angle changes. (a) Corresponding results from improved quasi-dense matching algorithm. (b) Corresponding results from original quasi-dense matching algorithm.

We have contended both theoretically and experimentally that the proposed algorithms outperform others in weakly textured points detecting and matching. The performance of the new detector was demonstrated on a wide variety of image sets in which the weakly textured objects were included. The proposed detector gives a way of detecting the weakly textured points which would be useful for 3D reconstruction of the object with weak texture at the dense matching step. On the other hand, the proposed matching algorithm SRM has the higher performance than others on the LBP feature space; moreover, to a certain extent, SRM can handle occlusion and corruption on the feature of weakly textured points.

Intriguing questions for future work are whether this method is appropriate for wide-based stereo and what description can be used for these weakly textured points. The first problem can be solved by replacing circumferences with ellipses; the second one might be solved by using the information of the surround points which are detected by classical detectors, such as SIFT, GOLH, and SURF, and it is a most challenging one as well. These are two directions for our future work.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by the Key Project of the National Nature Science Foundation of China (no. 61134009), the National Nature Science Foundation of China (no. 61473078), Cooperative research funds of the National Natural Science Funds Overseas and Hong Kong and Macao scholars (no. 61428302), Specialized Research Fund for Shanghai Leading Talents, Project of the Shanghai Committee of Science and Technology (nos. 13JC1407500, 11JC1400200), and Innovation Program of Shanghai Municipal Education Commission (no. 14ZZ067).

References

[1] 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.

[2] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124–1137, 2004.

[3] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the Alvey Vision Conference*, pp. 147–151, 1988.

[4] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," in *Proceedings of the 8th International Conference on Computer Vision (ICCV '01)*, vol. 1, pp. 525–531, Vancouver, Canada, July 2001.

[5] T. Lindeberg, "Feature detection with automatic scale selection," *International Journal of Computer Vision*, vol. 30, no. 2, pp. 79–116, 1998.

[6] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[7] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Computer Vision—ECCV 2002*, vol. 2350 of *Lecture Notes in Computer Science*, pp. 128–142, Springer, Berlin, Germany, 2002.

[8] F. Jurie and C. Schmid, "Scale-invariant shape features for recognition of object categories," *Computer Vision and Pattern Recognition*, vol. 2, pp. 90–96, 2004.

[9] T. Tuytelaars and L. van Gool, "Wide baseline stereo matching based on local, affinely invariant regions," in *Proceedings of the British Machine Vision Conference (BMVC '00)*, pp. 412–422, 2000.

[10] H. Bay, T. Tuytelaars, and L. van Gool, "SURF: speeded up robust features," in *Computer Vision—ECCV 2006*, vol. 3951 of *Lecture Notes in Computer Science*, pp. 404–417, Springer, Berlin, Germany, 2006.

[11] M. J. Atallah, "On symmetry detection," *IEEE Transactions on Computers*, vol. 34, no. 7, pp. 663–666, 1985.

[12] G. Loy and A. Zelinsky, "Fast radial symmetry for detecting points of interest," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 959–973, 2003.

[13] J. Maver, "Self-similarity and points of interest," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1211–1226, 2010.

[14] V. G. Kim, Y. Lipman, and T. Funkhouser, "Symmetry-guided texture synthesis and manipulation," *ACM Transactions on Graphics*, vol. 31, no. 3, article 22, 2012.

[15] T. Lindeberg, *Scale-Space Theory in Computer Vision*, Springer, Boston, Mass, USA, 1994.

[16] R. Deriche, Z. Zhang, Q. Luong, and O. Faugeras, "Robust recovery of the epipolar geometry for an uncalibrated stereo rig," in *Computer Vision—ECCV '94*, pp. 567–576, 1994.

[17] D. Aiger and K. Kedem, "Exact and approximate geometric pattern matching for point sets in the plane under similarity transformations," in *Proceedings of the 19th Canadian Conference on Computational Geometry (CCCG '07)*, pp. 181–184, Ottawa, Canada, August 2007.

[18] D. Donoho, H. Kakavand, and J. Mammen, "The simplest solution to an underdetermined system of linear equations," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '06)*, pp. 1924–1928, Seattle, Wash, USA, July 2006.

[19] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[20] M. Lhuillier and L. Quan, "A quasi-dense approach to surface reconstruction from uncalibrated images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 418–433, 2005.

[21] K. Mikolajczyk, T. Tuytelaars, C. Schmid et al., "A comparison of affine region detectors," *International Journal of Computer Vision*, vol. 65, no. 1-2, pp. 43–72, 2005.

[22] T. Kadir, A. Zisserman, and M. Brady, "An affine invariant salient region detector," in *Computer Vision-ECCV 2004*, pp. 345–457, 2004.

[23] C. Sun, "Fast stereo matching using rectangular subregioning and 3D maximum-surface techniques," *International Journal of Computer Vision*, vol. 47, no. 1–3, pp. 99–117, 2002.

[24] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990.

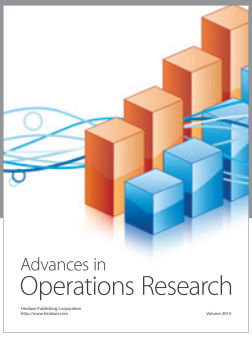
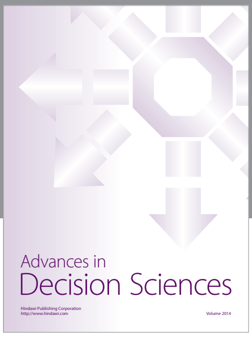
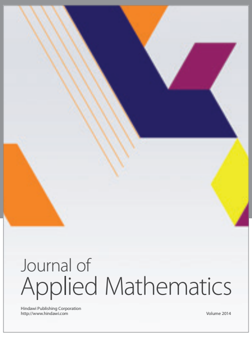
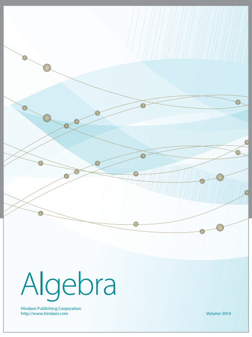
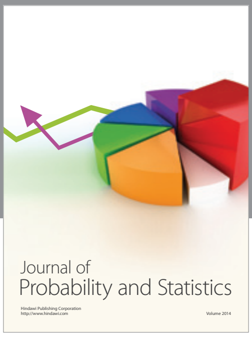
[25] C. Wallraven, B. Caputo, and A. Graf, "Recognition with local features: the kernel recipe," in *Proceedings of the 9th IEEE International Conference on Computer Vision*, pp. 257–264, Nice, France, October 2003.

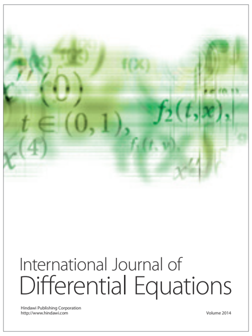
[26] S. Cost and S. Salzberg, "A weighted nearest neighbor algorithm for learning with symbolic features," *Machine Learning*, vol. 10, no. 1, pp. 57–78, 1993.

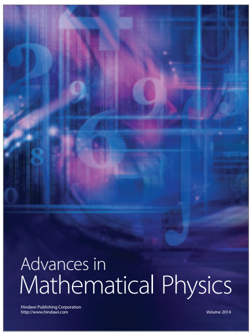
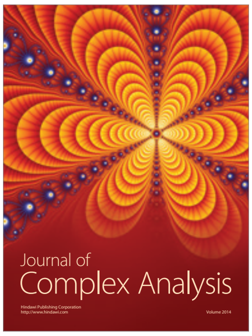
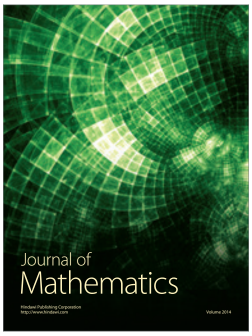
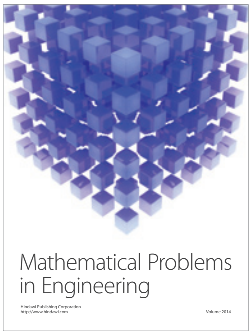
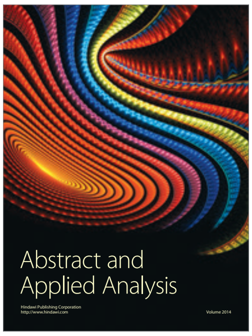
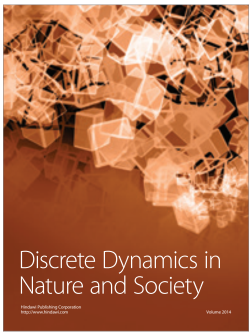
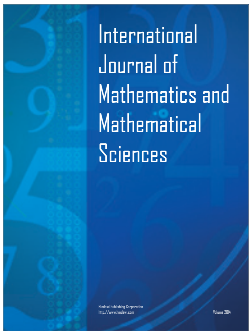
[27] E. Amaldi and V. Kann, "On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems," *Theoretical Computer Science*, vol. 209, no. 1-2, pp. 237–260, 1998.

[28] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: universal encoding strategies?" *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

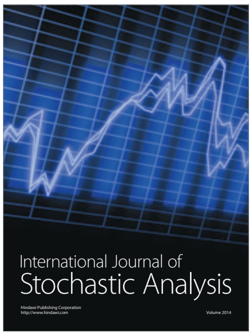
[29] C. Liu, "Capitalize on dimensionality increasing techniques for improving face recognition grand challenge performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 725–737, 2006.


Advances in
Operations Research

Advances in
Decision Sciences

Journal of
Applied Mathematics

Algebra

Journal of
Probability and Statistics

The Scientific
World Journal

International Journal of
Differential Equations

International Journal of
Combinatorics

Advances in
Mathematical Physics

Journal of
Complex Analysis

Journal of
Mathematics

Mathematical Problems
in Engineering

Abstract and
Applied Analysis

Discrete Dynamics in
Nature and Society

International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces

International Journal of
Stochastic Analysis

Journal of
Optimization