Research Article

Extracting Corresponding Point Based on Texture Synthesis for Nearly Flat Textureless Object Surface

Min Mao,¹ Kuang-Rong Hao,^{1,2} and Yong-Sheng Ding^{1,2}

¹College of Information Sciences and Technology, Donghua University, Shanghai 201620, China

²Engineering Research Center of Digitized Textile & Fashion Technology, Ministry of Education, Donghua University, Shanghai 201620, China

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

Received 5 June 2014; Accepted 28 August 2014

Academic Editor: Minrui Fei

Copyright © 2015 Min Mao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Since the image feature points are always gathered at the range with significant intensity change, such as textured portions or edges of an image, which can be detected by the state-of-the-art intensity based point-detectors, there is nearly no point in the areas of low textured detected by classical interest-point detectors. In this paper we describe a novel algorithm based on affine transform and graph cut for interest point detecting and matching from wide baseline image pairs with weakly textured object. The detection and matching mechanism can be separated into three steps: firstly, the information on the large textureless areas will be enhanced by adding textures through the proposed texture synthesis algorithm TSIQ. Secondly, the initial interest-point set is detected by classical interest-point detectors. Finally, graph cuts are used to find the globally optimal set of matching points on stereo pairs. The efficacy of the proposed algorithm is verified by three kinds of experiments, that is, the influence of point detecting from synthetic texture with different texture sample, the stability under the different geometric transformations, and the performance to improve the quasi-dense matching algorithm, respectively.

1. Introduction

In the last decades, more and more methods for 3D modeling have been proposed. Techniques which only use images or video as input have been developed to reconstruct 3D scenes. For examples, techniques of shape-from-video used flexibility of the recording to reconstruct the wide variety of scenes. However, these methods require large overlap between subsequent frames, which might not always be possible to record video of the interest object due to time pressure or obstacles. Furthermore a reconstruction is desirable from the images which are not taken for the purpose of 3D modeling. Since the stereo matching has been focusing on the small baseline stereo or frontoparallel planes, many algorithms such as graph cuts [1–3], minimal path search [4], and belief propagation [5] are based on the diversity of concepts.

One of the most studied problems in stereo matching is finding corresponding points between stereo image pairs, and this is sometimes very hard. The system should set

aside smoothness assumptions to detect the occlusions and depth discontinuities, and it also strongly depends on the capacity of handling weakly textured regions. The algorithms mentioned above can handle short-baseline stereo matching well; however in contrast, there are much more challenging in the wide-baseline situation due to increased occluded areas and large perspective distortions. But the wide-baseline only requires fewer images to reconstruct a scene completely, and it is worth addressing.

On the other hand, local, viewpoint invariant features can be used for wide-baseline matching, and hence the viewpoints can be further apart. Amintoosi et al. [6] used SIFT key-points for image registration, and Jian et al. [7] proposed a key-point detector based on wavelet transform for image retrieval. In general, the process for matching the discrete image points can be divided into three main steps: firstly, extracting the interest point from each image, such as T-junctions, corners. The point detector should have the property of repeatability, which guarantees finding

the same physical interest point under different viewing conditions. Secondly, each interest point can be represented by a distinctive feature vector through the descriptor. Finally, the descriptor vectors are matched between different images.

Almost all of the detectors are based on the gradient map of image. For example, the Harris corner detector [8] 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 [9] proposed two scale-invariant methods, that is, Harris-Laplace and Hessian-Laplace, which are based on the concept of automatic scale selection [10]. And the location is selected by Harris measure or the determinant of the Hessian matrix; scale is selected by Laplacian. Lowe [11] speeds up the above methods by using the difference of Gaussians (DoG) to approximate the Laplacian of Gaussians (LoG). Stansk and Hellwich [12] introduce a new operator to extract the salient points from image, which used interest points as the anchor points to match points between different perspectives. There are lots of different detectors that have been proposed in the literature [13–16]. However, image blurring, magnification, and illumination are still problems of the methods based on interest point; one of the most serious weaknesses is that these methods could not get a point on the weakly texture areas by classical interest-point detectors. For solving this, Dragon et al. [17] proposed an NF-features method to complementary regular feature detection, which is similar to our method. The method uses regular detectors, such as SIFT, Hessian-Affine, and SURF to get the interest points surround the nontextured regions and get the NF-features by using these points as anchor features according to the Euclidean distance between every location in the textureless areas and regular point.

In this paper, we will extend the capability of classical interest-point detectors by transfer strongly texture to extracting the corresponding points in the weakly texture areas. The motivation is the fact that large nearly flat textureless areas in the real world are always assumed as a smoothness surface [18], which means the depth of these areas are continuous. Hence, if the textureless area is nearly flat, it can also be approximated by a plane and the transformation of this plane between two image pairs can be described by a simple affine transform. Different from the texture synthesis for 3D shape algorithm, we will use a 2D texture synthesis for the weakly texture regions in two image pairs and then use classical interest-point detectors to extract the correspondences from these regions. Hence, we can take advantage of these powerful detectors to obtain corresponding points from 3D textureless objects in wide-baseline stereo matching. We focus only on the points corresponding problem in large nearly flat textureless areas and propose an algorithm to get the corresponding points on those areas by using both salient features surround large weakly texture objects and the information on these objects. The proposed algorithm can be divided into two steps. First, the feature points detected on the contour of the object, which has a large textureless region, will be used to estimate the

affine transform and to locate the center points for adding synthesizing textures on the object surface between two image pairs. Second, since the real surfaces of these objects are not planar completely, the correspondences extract from the first step are not accurate. We therefore use the graph cut algorithm to adjust the corresponding results from first step by incorporating the features on the object, such as the intensity and the self-similarity.

The proposed approach can be used for enhancing the performance of dense wide-baseline matching by increasing the seed points obtained from classical interest-point detectors [16]. The results obtained on image sets under different types of photometric and geometric transformations show the high robustness of the method. The proposed approach gives a rich set of corresponding points in the large weakly-textured region which has no distinctive feature points detected by classical detectors.

The paper is organized as follows. Section 2 describes the method for adding synthetic texture to image pairs. We will discuss why the texture can be added in two image pairs by an affine transform and how this transform will be estimated. In Section 3, we propose an approach to adjust the position of the corresponding point in the second image by graph cut algorithm and comparing the adjusting result to the one which is directly obtained from Section 2. In Section 4, we verify the proposed method with extensive experiments of human model images under different circumstances. Section 5 provides concluding remarks and possible extensions of the proposed approach.

2. Texture Synthesis for Point Detecting and Matching

In this section, we will present a texture transfer mechanism used for interest point detecting and matching. We first explain the reason why the relationship of a large weak texture region between two image pairs can be approximated by an affine transform and then can present the estimation methods of this affine transform and structuring synthetic texture for adding texture on the textureless object surface.

2.1. Relationship of Large Textureless Area between Image Pairs. Let P_0 be a 3D point and p, p' are the corresponding projection points in image plane I, I' from two cameras with different orientations, respectively. According to this relationship, a mapping $H : I \rightarrow I'$ exists, when $(p_i, p'_i) \in H$, $p_i \in I, p'_i \in I'$ are the projection points from a same scene point P_i .

In wide-baseline situations, the feature of point in two images would be changed frequently by lighting or perspective changes. Thus, most researchers favor using classical detectors, such as GLOH and SURF [16, 19], which are both based on the gradient map of image and have proven successful for sparse matching in wide-baseline situation. And the mapping $H : I \rightarrow I'$ can be got by using the sparse matching result as seed points [20].

The general sparse matching algorithm is as follows:

- (1) interesting point detecting;
- (2) point describing;
- (3) matching points between two images.

Let P and P' denote the point sets got by interest point detector, x is a point in P , and then its corresponding point x' in P' can be got as follows:

$$x' = \arg \min_{x' \in P'} D(F(x), F(x')), \quad (1)$$

where $F(\cdot)$ denotes the descriptor operator, $D(\cdot)$ is the match operator and the pair of points $(x, x') \in H$.

From the above analysis, we see that the result from sparse matching algorithm depend largely on its detector and descriptor, so these algorithms cannot give the corresponding points in textureless areas, since points in these areas will not be detected by those detectors. On the other hand, texture synthesis can replace the textureless areas with strong texture image. The idea is as follows.

let Ω_{tx_1} and Ω_{tx_2} be the regions which the textureless areas are replaced by texture image I_{TX} in image I, I' , respectively; then according to sparse matching algorithm, it can get a new mapping $G : \Omega_{tx_1} \rightarrow \Omega_{tx_2}$. On the other hand, considering the relationship between two images, we should estimate the transformation of the texture image I_{TX} from I to image I' . Let $T(\cdot)$ denote the transformation operator, which transforms the point location in Ω_{tx_1} to Ω_{tx_2} . Then $T(\cdot)$ should satisfy the following two conditions.

Condition 1. If p, p' are corresponding points; that is, $(p, p') \in H$; then $T(p) = p'$.

Condition 2. Let $F(\cdot)$ denote the descriptor operator, since after the transformation we should still use classical detectors to detect interest point; then $T(\cdot)$ should satisfy $F(T(p)) \approx F * (p)$ and Condition 1, $p \in \Omega_{tx_1}, T(p) \in \Omega_{tx_2}$.

Obviously, if transformation operator $T(\cdot)$ satisfies the above conditions, then $G = H$ in $\Omega_{tx_1}, \Omega_{tx_2}$; that is, if $(p_1, p'_1) \in G$, then $(p_1, p'_1) \in H$, where (p_1, p'_1) are the point locations in image I, I' , respectively.

In general situation, it is more difficult to get transformation operator $T(\cdot)$ than to get mapping H directly. However, if $T(\cdot)$ is an affine transform, then descriptors such as SIFT and SURF are both satisfying Condition 2 with this transformation.

As discussed above, it is difficult to get a transformation which satisfies Condition 1, but fortunately, textureless region can be assumed as a plane, and in this paper we assume that these regions are almost planar, and according to the property of affine transform, it is possible to use affine transform as $T(\cdot)$ under this hypothesis. The following theorem explains the reason why an affine transform can be used as an approximation of mapping between 3D points in the same plane which are projected on different viewpoint images.

A 3D point coordinate vector in world coordinate system is denoted by $\mathbf{M} = [X_W, Y_W, Z_W, 1]^T$, and its coordinate

vector in camera coordinate system is $\mathbf{m} = [x, y, z, 1]^T$; according to the simple pinhole model, the relationship between $\mathbf{M}$ and $\mathbf{m}$ is given by

$$\mathbf{m} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{M}, \quad (2)$$

where $[\mathbf{R}, \mathbf{t}]$ is the extrinsic parameters which describes the rotation and translation between two coordinate systems. Moreover, the relationship between $\mathbf{m}$ and its image projection $\mathbf{m}' = [\mu, \nu, 1]^T$ is given by

$$\mathbf{m}' = \mathbf{C}\mathbf{m}, \quad \text{with } \mathbf{C} = \begin{bmatrix} a_x & 0 & \mu_0 & 0 \\ 0 & a_y & \nu_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad (3)$$

where $\mathbf{C}$ is the camera intrinsic matrix, a_x and a_y are scale factors in image μ and ν axes, respectively, and (μ_0, ν_0) are the coordinates of the principal point. According to formulas (2)-(3), the relationship between $\mathbf{M}$ and $\mathbf{m}'$ is given by

$$\mathbf{m}' = \mathbf{C} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{M} = \mathbf{B}\mathbf{M}, \quad (4)$$

where

$$\mathbf{B} = \mathbf{C} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \quad (5)$$

which denotes the Pinhole camera model. Let p_i ($i = 1, 2, 3$) be the 3D points in the same plane as shown in Figure 1; then each 3D point p_j on this plane can be represented as follows:

$$p_j = \begin{bmatrix} X_1 & X_2 & X_3 & 0 \\ Y_1 & Y_2 & Y_3 & 0 \\ Z_1 & Z_2 & Z_3 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ 1 \end{bmatrix} = [p_1, p_2, p_3] \cdot \mathbf{W} = \mathbf{P}\mathbf{W}, \quad (6)$$

where $\mathbf{W}$ is the parameter for p_i ($i = 1, 2, 3$) to represent point p_j in 3D coordinate. According to above formulas, the 2D coordinate of p_j in each image can be represented as

$$p'_j = \mathbf{B}' p_j = \mathbf{B}' \mathbf{P} \mathbf{W}, \quad p''_j = \mathbf{B}'' p_j = \mathbf{B}'' \mathbf{P} \mathbf{W}. \quad (7)$$

Equation (7) gives the 2D coordinates of a 3D point in two images, respectively, since the relationship of points p_i ($i = 1, 2, 3$) in image pairs can be represented by affine transform $\mathbf{A}$ as follows:

$$\mathbf{B}'' \mathbf{P} = \mathbf{A} \mathbf{B}' \mathbf{P}. \quad (8)$$

According to formulas (7) and (8), the relationship between p'_j and p''_j is represented as follows:

$$p''_j = \mathbf{B}'' \mathbf{P} \mathbf{W} = \mathbf{A} \mathbf{B}' \mathbf{P} \mathbf{W} = \mathbf{A} p'_j. \quad (9)$$

Formula (9) proves that the relationship between this two 2D coordinates in two images is an affine transform, which means if we find three noncollinear corresponding points between two image pairs, and the affine transform according to these points can be estimated, the corresponding relationship of the points which are on the same planar can be obtained from the affine transform. Figure 2(b) shows that the image got from the affine transform is almost seem to the one get from camera in different perspectives.

FIGURE 1: Pinhole camera model.

2.2. Snake Based Textureless Object Segmentation. As described in Section 2.1. the corresponding point on the textureless surface can be obtained by affine transformation. We use the snake-based scheme to segment the textureless region from image. In this paper, we use human model surface as the example of textureless object.

According to [21], the snake can be defined as a controlled continuity contour which is attracted by salient features of an image and whose motion should satisfy the minimize of the following energy function:

$$E_{\text{snake}} = \int_0^1 [E_{\text{int}}(r(s)) + E_{\text{ext}}(r(s))] ds, \quad (10)$$

where $r(s) = (x(s), y(s))$ represents the position of a snake. E_{int} and E_{ext} are the internal energy and the external energy, respectively. They are defined as

$$E_{\text{int}} = \frac{1}{2} (\alpha |r_s(s)|^2 + \beta |r_{ss}(s)|^2), \quad (11)$$

$$E_{\text{ext}}(r(s)) = -|\nabla G_\sigma(x, y) * I(x, y)|, \quad (12)$$

where the internal energy is used to keep the active contour smooth, and the external energy is depended on the application. $G_\sigma(x, y)$ is a 2D Gaussian kernel with the standard deviation σ , $I(x, y)$ denotes the image intensity at position (x, y) , and ∇ denotes the gradient operator.

The minimization of (10) can be solved by variational calculus techniques and the force balance equation as follows [19]:

$$\begin{aligned} -\frac{\partial E_{\text{ext}}}{\partial x} &= -\alpha x_{ss} + \beta x_{ssss}, \\ -\frac{\partial E_{\text{ext}}}{\partial y} &= -\alpha y_{ss} + \beta y_{ssss}, \end{aligned} \quad (13)$$

where $(-\partial E_{\text{ext}}/\partial x, -\partial E_{\text{ext}}/\partial y)$ are the external forces which can be used for attracting the snake to the object boundary. Figure 3 shows the segmentation results.

2.3. Embed Texture Image Based on Feature Point. After the objects have been segmented, the next step is to estimate the affine transform between them and select the centre position for adding the texture image in each object. To develop this method, we first have to extract the image content. Since the centre positions in two images should be the same point on the real object, the representation of the image content has to be robust to geometric transformations. We choose classical feature points detectors to extract the image content.

The embed texture method is summarized as follows:

- (1) detecting robust feature points on the object (include the boundary of the object);
- (2) using different sizes of slide windows to detect the largest region inside the textureless object, and this region will seem to be the centre region of the object;
- (3) computing the characteristic scale of each detected feature point by searching for a local extremum over scales of LoG as the way in [22], and the detection scale is represented by a circle around the point as show in Figure 3;
- (4) selecting point whose characteristic scale region and the centre region are intersect;
- (5) using those points' positions to estimate the affine transform by least squares, computing the centroid coordinate of those points and this coordinate will seem to be the centre position of the textureless object.

Note that they might have no feature points on the object; therefore we will add the feature points along the contour of the object by a marked cross line. Figure 3 is the texture adding presentation from above mechanism.

2.4. Texture Synthesis Based on Image Quilting (TSIQ). In this section we will develop a texture synthesis algorithm based on the quilting technique described in [23]. The goal of texture synthesis algorithms TSIQ is that using images from samples of the real world to synthesize novel views instead of recreating the entire physical world from scratch directly. In the situation of weakly texture areas, we need to use this algorithm to transfer texture from the samples with strong texture to these areas. However, the traditional texture synthesis algorithm need not consider the correspondence

FIGURE 2: Plane transformation by an affine transform. (a) Target images. (b) The transformation results from the original image according to the affine transform. (c) Three corresponding points between image pairs which are used to estimate the affine transform, where the first column in (c) is the original image, and the second column is the target images.

FIGURE 3: The texture adding mechanism using human model. First row, from left to right: (a) detecting the largest region inside the textureless object with different sizes of slide windows. (b) Getting points overlap the slide window. (c) Getting the corresponding point. The second row, from left to right: (a), (b) computing centre position of the textureless object on each image and estimating the affine transform between two images. (c) Adding synthetic texture to the object surface in two images.

problems between two image pairs; hence, it should not be used in weak texture areas directly.

The texture synthesis algorithm should follow the following criteria.

Criteria 1. The results from this algorithm could be used for point matching, specifically; the classical detectors can find corresponding points as much as possible; that is, each patch of quilting should be distinctive under an affine transformation.

Criteria 2. The distribution of the detected correspondences inside the texture image should be uniform against the circumstance that the points gathered on the edge which were quilting together.

According to the above two criteria, our proposed TSIQ algorithm is as follows.

- (1) Go through the image to be synthesized in raster scan order in steps of one block.
- (2) For every location, search the input texture for a set of new blocks and use some transformation to transform each block (e.g., use rotation or Gaussian transformation), and pick the transformed block with the biggest distinction compared to old blocks.
- (3) Compute the error surface between the old blocks and the newly chosen transformed block at overlap region. Find the minimum cost path along the surface and make the boundary of the new block. Paste the block onto the texture. Repeat.

FIGURE 4: The results from traditional texture synthesis algorithm and TSIQ. (a) Texture sample. (b) The corresponding result on the traditional synthetic texture image. (c) The synthesis texture given by TSIQ. (d) The corresponding result of TSIQ.

FIGURE 5: The assumed possible region of corresponding point in the right picture.

The difference between our image quilting algorithm and the algorithm in [23] is that we transform each new block and pick the one with the biggest distinction to the old blocks in step 2; the reason of doing this is directly from Criteria 1.

Figure 4 shows the results from our texture synthesis algorithm, and the performance is quite good for extracting correspondences.

3. Adjust the Corresponding Points Based on Graph Cut

Since the surface of human model is not planar completely (see Figure 5), the matching result obtained from classic

detectors is not accurate exactly. The second picture of Figure 5 shows the possible position of corresponding point. We assume that the possible position in the region which surround the initial corresponding point detected by classical interest point detectors. On the other hand, the human model of surface can be assumed to be composed of local regions which are self-similar, and these self-similar regions are also distinguishable with their neighborhoods. Hence, this structure can also be used for point matching.

In this section, we propose a method of incorporation self-similarity into graph cut to improve the point matching accuracy. In Section 3.1, we focus on measuring the self-similarity of region in human model surface. In Section 3.2,

we explain the mechanism for incorporating self-similarity within the formulation of graph cut. In Section 3.3, we adjust the performance of point matching process.

3.1. Self-Similarity Region Quantifying. Let $I(x)$ be the intensity value at location x , P denotes points set in a fragment Ω , and $\bar{I}$ denotes the average intensity value of fragment Ω . Then the similarity of the fragment Ω by transformation T can be measured by the normalized correlation coefficient as follows:

$$\text{ncc}(P, T) = \frac{\sum_i (I(x_i) - \bar{I})(I(T(x_i)) - \bar{I})}{\sqrt{\left(\sum_i (I(x_i) - \bar{I})^2\right) \left(\sum_i (I(T(x_i)) - \bar{I})^2\right)}}. \quad (14)$$

Since transformation T is reflection and rotation, formula (14) also can be written as follows:

$$\text{ncc}(P, T) = \frac{\sum_i (I(x_i) - \bar{I})(I(T(x_i)) - \bar{I})}{\sum_i (I(x_i) - \bar{I})^2}, \quad (15)$$

where the purpose of formula $I(x) - \bar{I}$ is to reduce the influence caused by intensity change.

It is clear that fragment on the human model surface has strong mirror symmetry, let point $x = (r, \theta)$ be in polar coordinates, let $\varphi \in [0, \pi)$ denote the mirror line orientation, and then the symmetric point of x about the mirror line can be represented as $x' = (r, 2\varphi - \theta)$. For measuring the mirror symmetry about the region Ω_x , it should fulfill all mirror line orientations $\varphi \in [0, \pi)$, the symmetry of region Ω_x can be obtained as follows [24]:

$$S_{w\text{-texture}}(\Omega_x) = \frac{1}{N} \sum_{i=0}^{N-1} \text{ncc}(\Omega_x, T(\varphi_i)), \quad (16)$$

where $T(\cdot)$ denotes the symmetry transformation function, and it transforms 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 $\max(S_{w\text{-texture}}(\Omega_x)) = 1$ and $\min(S_{w\text{-texture}}(\Omega_x)) = 0$, $\Omega_x \in I$.

3.2. Graph Cut Based Point Matching. In this section we incorporate self-similarity into graph cut for improving the point matching accuracy.

The idea is as follows: let $\vec{l}$ be the motion vector from prior correspondence to the accurate correspondence position. If pixels p and q correspond, they are assumed to have similar intensity, and since the surface of object is piecewise smooth, same value of self-similarity S_p and S_q should have same motion vector $\vec{l}$. However, there are special circumstance when corresponding pixels have very different intensity due to the effects of image sampling. We use the technique of [25] to avoid the sensitivity to image sampling.

On the other hand, while the matching results given by classical interest point detector are not accurate exactly, they have given the approximation locations of point correspondences in two images. Hence, these results can be used as

constraint conditions, and we assume the accurate location of point correspondence in the region with the center of its prior correspondence.

As mentioned above, the goal is to find a labeling f that assigns each pixel x a label $f_x \in L$, the mean of labeling f is the location of accurate correspondence against its prior correspondence, and since the distribution of the points on the human model surface is both piecewise smooth and consistent with observed data, f should also satisfy these two conditions.

It can be formulated in terms of energy minimization to find the best labeling f , and the energy can be represented as follows:

$$E(f) = E_{\text{smooth}}(f) + E_{\text{data}}(f). \quad (17)$$

$E_{\text{smooth}}(f)$ measures the extent to which f is not piecewise smooth, and $E_{\text{data}}(f)$ measures the disagreement between f and the observed data. The form of the energies can also be represented as

$$E(f) = \sum_{\{x,y\} \in N} V_{x,y}(f_x, f_y) + \sum_{x \in p} D_x(f_x). \quad (18)$$

In this paper we use the Potts model for its distinct penalty V ; that is,

$$E(f) = \sum_{\{x,y\} \in N} u_{\{x,y\}} \cdot T(f_x \neq f_y) + \sum_{x \in p} D_x(f_x). \quad (19)$$

We turn to improving the point correspondence problem, and the improving process is scored according to the following criteria.

Criterion 1. The pixel's intensity $I(x)$ is used for measuring how well the pixel matches its corresponding point; that is, comparing intensities between corresponding points and low values represents better matches.

Criterion 2. Each pixel x on the human model surface is given a value of self-similarity with region Ω_x . According to the property of piecewise smooth, the adjacent pixels with the same value of self-similarity should also have the same label; hence, the adjacent pixels which have the same label correspond to low E_{smooth} . The goal of the above criteria is to devise an algorithm improving the point correspondence results in Section 2.

For data support energy $E_{\text{data}}(f)$, how well the point position x fits to its corresponding point position y in the range of $|y' - (y + f)| \leq 1/2$ will be measured by

$$C_{fwd}(x, y, f) = \min_{|\Delta l - f| \leq 1/2} |I_x - I'_{y+\Delta l}|. \quad (20)$$

The fractional values $I'_{y+\Delta l}$ can be gotten by linear interpolation between pixel values, and we also measure the following parameter as symmetry:

$$C_{rev}(x, y, f) = \min_{|\Delta l| \leq 1/2} |I_{x+\Delta l} - I'_{y+f}|. \quad (21)$$

FIGURE 6: Results of corresponding-points adjusting by graph cut algorithm. The second row is the local enlarged image of the first row; the dot points in the right image are the modulated points adjusted by graph cut algorithm. The square points in the right image are the original points directly got from classical point detector.

The final dissimilarity C between two points is defined as follows:

$$C(x, y, f) = \left(\min \{C_{fwd}(x, y, f), C_{rev}(x, y, f), \text{const}\} \right)^2. \quad (22)$$

In this paper we set $\text{const} = 20$ for all experiments.

For smooth energy $E_{\text{smooth}}(f)$, as mentioned above, we consider the Potts model as the distinct penalty, and it can be represented as $V_{x,y} = u_{\{x,y\}} \cdot T(f_x \neq f_y)$. The information of self-similarity of pixels in the first image can significantly influence the assessment of motion vector label f without even considering the second image. Specifically, the neighboring pixels are much more likely to have same motion vector label if we know that $S_x \approx S_y$, that is, Criterion 2.

According to Criterion 2, it is easy to incorporate the information of the first image into the framework by varying $u_{\{x,y\}}$ depending on the self-similarities S_x and S_y . Let

$$u_{\{x,y\}} = U(|S_x - S_y|). \quad (23)$$

Each $u_{\{x,y\}}$ denotes a penalty for assigning different motion vector label to neighboring pixels x and y . As Criterion 2 mentioned above, the value of penalty $u_{\{x,y\}}$ should be larger for pairs $\{x, y\}$ with smaller self-similarity

difference $|S_x - S_y|$. In practice, $U(|S_x - S_y|)$ can be defined as follows:

$$U(|S_x - S_y|) = \begin{cases} 2K & \text{if } |S_x - S_y| \leq 0.02 \\ K & \text{if } |S_x - S_y| > 0.02, \end{cases} \quad (24)$$

where K is the Potts model parameter.

3.3. Improved Results from Graph Cut. The size of images we used here is 512×512 , and the region of the motion vector label for graph cut is 81×81 pixels, so the position of the corresponding points in the second image can be adjusted in this size of region with the center of their prior correspondences positions. The self-similarity radius for each point is set to 10 pixels. To determine the neighboring of each point, we set the distance up to 30 pixels; that is, two points in the first image are neighbours, if their distance is less than 30 pixels.

Figure 6 shows the correspondences from classical detectors and the adjusted correspondences (e.g., in each right image of the corresponding pairs, the points represented by square are got from classical detectors directly, and the points represented by circle are results from the graph cut algorithm.), respectively. Moreover, for showing the efficiency of the adjustment in more detail, we also marked four improved correspondence results.

4. Experiment Results

In this section, we present experiments for evaluation of the proposed approach. The experiments can be divided into three parts. In the first part, the performance of extracting feature points from textureless region will be examined in a texture database of 36 samples of textures with three classical interest point detectors. In the second part, we will demonstrate the stability of TSIQ method by comparing the performance across different images in various situations with classical detectors. And in the last part, TSIQ will be used for improving the performance of dense wide-baseline matching [20].

4.1. Points Extraction Testing with Different Textured Surfaces.

Since the number of detected points varies with different textured surfaces, we measure our algorithm performance for points detection under the publicly available textured database at <http://sipi.usc.edu/database/database.php>. And the texture with the best performance is selected under three classical point detectors, that is, SIFT, SURF, and Harris-Affine. The database consists of 12 samples, each of them has three different textures (the resolution of each sample is 512×512), and the textures include additional sources of variability wherever possible: inhomogeneous texture patterns (bark and wood), nonplanarity surface (bark), and nonrigid deformations in the same classes (water and leather). For measuring the performance, we use the proportion between the number of detected points on the objects (OP) and total number of detected points (TP). Table 1 gives the average performance at each texture with three detectors under test images.

The first observation of the results from Table 1 is that SIFT detector which performs better than SURF, and Harris-Affine, but it is more sensitive to the change of texture than others; the standard deviations of the points' numbers on the object detected by SIFT, SURF and Harris-Affine are 123.6, 72.6, and 27.9, respectively. Next, the performance of individual textures for our system can be analyzed. To this end, the texture is arranged in order to increase the detection rate of points on the objects which are detected by SIFT detector (second column). Roughly speaking, the number of detected points is positively correlated with the resolution of the texture. Some of the lowest numbers belong to coarse-scale like Brick wall (T12) and weave (T04), while some of the highest belong to fine-scale textures like plastic bubbles (T13), grass (T01), and woolen cloth (T05). However, the relationship is not universal. For example, wood grain (T09), which is fine-scale, has a relatively low rate 60.62%, while the large-scale, bark, (T02) has a relatively high rate 80.18%. On the other hand, the order of detection rate is different with different detectors. Overall, the intrinsic characteristics of the various textures are not related directly to point detection performance. According to Table 1, Raffia (T10) has the best average performance under three detectors. Figure 7 shows the performance of TSIQ under T10.

4.2. The TSIQ Performance under Varying Blur, Lighting Change, Rotating, and Viewpoint Change. The aim of these experiments is to test the proposed TSIQ algorithm with three detectors, that is, SIFT, SURF, and Harris-Affine. Four measures are reported: the number of matched points on the object with and without adding texture (i.e., TOP and OP) and the number of matched points on the total pairs of test images with and without adding texture (i.e., TTP and TP). Since the Harris-Affine detector has no descriptor, the SIFT descriptor will be used to represent the detected region. To display the performance of baseline under different photometric and geometric transformations, all image pairs used in this experiment are shown in Figures 8, 9, 10, and 11; the curves show the proportion of the number of matched points on the object compared with the original image pairs under different transformation levels.

Moreover, since the purpose is to detect the stability of TSIQ in various situations, the number of points detected from the initial image pairs become important for measuring this property, so all experiments reported will use Raffia (T10) as the texture sample because of its best performance for point detection that has been demonstrated by the previous experiment. On the other hand, the interest points were detected on images of size 512×512 , and the resolution of texture image is 256×256 .

4.2.1. Lighting Changes. Figure 8 shows the results obtained from the reference image and lighting change images which are undergoing decreasing amounts of light intensity. The initial numbers of matched points on the object got from SIFT, SURF, and Harris-affine are 114, 208, and 128, respectively; when the intensity decreases to 50%, the decrease of the numbers of detected points on the weak texture region is 23.7%, 22.6%, and 34.4%, respectively. However the ratio between the number of points detected on the object and total image is increased with the decrease of intensity level. There are two reasons for those results: the first one is that the lighting change has little influence on HSV space, so the contour of human detected by snake algorithm is almost the same in different levels; moreover the synthetic texture added on the region is independent from lighting changes. More generally, the second reason is that the feature in the background becomes nonsignificant to detectors.

4.2.2. Scale Changes. Results presented in Figure 9 clearly show the performance on the image pairs with increasing amounts of object scale. The initial numbers of matched points on the object got from SIFT, SURF, and Harris-affine are 66, 68, and 62, respectively. And the reason why it gets the low numbers of initial matched points on the object here is that the object on the reference image is too small (as Figure 9 shows). The decrease of the numbers of detected points on the total pair of test images is 38.1%, 53.9%, and 61.1%, respectively. The decrease of the numbers of detected points on the object is 57.6%, 23.5%, and 62.9%, respectively. The decrease speed of the number of detected points on the object is slower than on the total pair of test images, as show in Figure 9(c). According to the performance with

FIGURE 7: Performance of TSIQ under T10.

FIGURE 8: Lighting changes for image. First row, from left to right: the first image is the left image in each image pairs; images from second to the end are the right images in each image pairs. The second row from (a) to (c): the Number of TOP, OP, TTP, and TP under lighting changes with SIFT, SURF, and Harris-affine detectors, respectively. The third row: (a) the proportion of number of corresponding points on the objects compared to the original image pairs. (b) The proportion of number of corresponding points on entire image pairs compared to the original image pairs. (c) The ratio between the number of points detected on the object and entire image pairs.

TABLE 1: Performance of points' extraction on different texture surface.

Texture images	SIFT		SURF		Harris-Affine	
	OP/TP	OP	OP/TP	OP	OP/TP	OP
T09 (Wood grain)	0.6062	137	0.5465	376	0.3436	167
T12 (Brick wall)	0.6293	146	0.5355	347	0.3319	152
T04 (weave)	0.6654	179	0.3573	174	0.4011	223
T03 (Straw)	0.6813	171	0.5081	314	0.3458	175
T08 (Water)	0.6957	208	0.5620	390	0.3851	201
T06 (calf leather)	0.7833	318	0.5756	411	0.4177	241
T02 (Bark)	0.8018	356	0.5758	418	0.3897	205
T01 (Grass)	0.8022	369	0.5615	402	0.3860	210
T13 (Plastic bubbles)	0.8083	371	0.5658	404	0.4019	207
T07 (Beach sand)	0.8164	378	0.5522	397	0.4026	219
T05 (Woolen cloth)	0.8242	422	0.5893	452	0.4044	220
T11 (Pigskin)	0.8257	431	0.5961	459	0.4063	221

TABLE 1: Continued.

Texture images	SIFT		SURF		Harris-Affine	
	OP/TP	OP	OP/TP	OP	OP/TP	OP
0.8447	506	0.5543	388	0.4323	249
T10 (Raffia)						
Avg.	0.7527	307.07	0.54	379.38	0.39	206.92

FIGURE 9: Scale changes for image. First row, from left to right: the first image is the left image in each image pairs; images from second to the end are the right images in each image pairs. The second row from (a) to (c): the number of TOP, OP, TTP, and TP under scale changes with SIFT, SURF, and Harris-affine detectors, respectively. The third row: (a) the proportion of number of corresponding points on the objects compared to the original image pairs. (b) The proportion of number of corresponding points on entire image pairs compared to the original image pairs. (c) The ratio between the number of points detected on the object and entire image pairs.

FIGURE 10: Changes in vertical camera angle. First row, from left to right: the first image is the left image in each image pairs; images from second to the end are the right images in each image pairs. The second row from (a) to (c): the number of TOP, OP, TTP, and TP under vertical camera angle changes with SIFT, SURF, and Harris-affine detectors, respectively. The third row: (a) the proportion of number of corresponding points on the objects compared to the original image pairs. (b) The proportion of number of corresponding points on entire image pairs compared to the original image pairs. (c) The ratio between the number of points detected on the object and entire image pairs.

three detectors, SURF gives the best stability under the scale changes situation (Figure 9(a)).

4.2.3. Changes in Vertical Camera Angle. Figure 10 shows the results changing amounts of vertical camera angle. The initial numbers of matched points on the object got from SIFT, SURF, and Harris-affine are 136, 193, and 144, respectively. The decrease of the numbers of detected points on the total pair of test images is 74.7%, 66.2%, and 71.6%, respectively, and the decrease of the numbers of detected points on the object is 78.7%, 69.4%, and 74.3%, respectively, the decrease speed of the number of detected points on the object is almost

close with the one on the total pair of test images as show in Figure 10(c), and it also demonstrates that the stability of added texture by this scheme is the same as the nature texture on the background in this situation; furthermore, the performance depends mostly on detectors. Here the performance of three detectors is nearly as the same as the one shown in Figures 10(a) and 10(b).

4.2.4. Changes in Horizontal Camera Angle. The results for changing amounts of horizontal camera angles are shown in Figure 11. The initial numbers of matched points on the object got from SIFT, SURF, and Harris-affine are 97, 147, and 94,

FIGURE 11: Changes in horizontal camera angle. First row, from left to right: the first image is the left image in each image pairs; images from second to the end are the right images in each image pairs. The second row from (a) to (c): the number of TOP, OP, TTP, and TP under horizontal camera angle changes with SIFT, SURF, and Harris-affine detectors, respectively. The third row: (a) the proportion of number of corresponding points on the objects compared to the original image pairs. (b) The proportion of number of corresponding points on entire image pairs compared to the original image pairs. (c) The ratio between the number of points detected on the object and entire image pairs.

respectively. The decrease of the detected points on the total pair of test images is 61.2%, 52.8%, and 48.0%, respectively. And the decrease of the detected points on the object is 49.5%, 38.1%, and 31.9%, respectively. Harris-Affine detector gives the most stability under these perspective distortions. The comparative evaluation of detected points with the ones on the total image pairs has confirmed that the TSIQ method is more stable than nature texture on the background in this situation as shown in Figure 11(c).

4.3. Improved Quasidense Matching. In this section, we use the TSIQ mechanism to promote the quasidense matching algorithm which was proposed by Maxime Lhuillier and Long Quan (2002). The quasidense 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 quasidense disparity map. Since most initial sparse seed matches are distributed in the strong texture regions, they nearly have

FIGURE 12: Results from improved dense matching and original one under different circumstance. Top row: corresponding points matched with horizontal camera angle changes. Middle row: corresponding points matched with vertical camera angle changes. Bottom row: corresponding points matched with scale changes. (a) Corresponding results from improved quasidense matching algorithm. (b) Corresponding results from original quasidense matching algorithm.

no seed matches in the textureless areas; furthermore if the matches got from propagated step in this area are wrong, gross reconstruction errors will occur, so we use the proposed approach to get the initial sparse seed matches in the large textureless areas.

The quasidense matching algorithm in [20] is independently implemented for comparison on the same image pairs. The TSIQ based algorithm is evaluated on the changes of horizontal camera angle, vertical camera angle, and image scale, respectively, and the corresponding point sets obtained by the proposed TSIQ based method and traditional quasidense matching algorithm are shown in Figure 12; according to this result our method can get more points on the textureless region between wide baseline images due to more sparse seeds gotten by TSIQ algorithm. Table 2 also lists the numbers of corresponding points taken by improved quasi and original quasi, respectively, under different circumstance. The improved quasidense matching algorithm is very efficient; in particular, the number of corresponding points gotten from this algorithm is nearly three times more than the original one with the vertical camera angle situation.

5. Conclusions

We propose a TSIQ based method to add texture to large weak texture areas between two wide-baseline image pairs; it extends the capability of classical detectors to extract the correspondences in the weak texture. While adding corresponding texture to image pairs is sometimes harder

TABLE 2: Comparison between original dense matching and TSIQ under different circumstance.

Change	Algorithm	#Corresponding point on untext
Horizontal camera angle	Quasi + TSIQ	813
	Quasi	637
Vertical camera angle	Quasi + TSIQ	876
	Quasi	222
Scale	Quasi + TSIQ	270
	Quasi	130

even than extracting the corresponding points directly, we developed an incorporate affine transform and graph cut algorithm to add corresponding texture to large weak texture areas based on the assumption described in Section 2. The performance of this algorithm was demonstrated on a wide variety of geometric transformations, and empirically the TSIQ based algorithm performs well for improving the dense wide-baseline matching. The proposed approach gives rich information in the textureless object under wide-baseline that can be used for 3D reconstruction.

A deficiency of the proposed method is that the added texture will become unreliable, when there are no enough corresponding points surrounding the object, and our future research goal is to find a way to use the information inside the object such as edges to estimate the affine transform which

can extend this approach to larger perspective distortion circumstance.

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 and 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] V. Kolmogorov and R. Zabih, "Multi-camera scene reconstruction via graph cuts," in *Computer Vision—ECCV 2002*, vol. 2352 of *Lecture Notes in Computer Science*, pp. 82–96, Springer, Berlin, Germany, 2002.
- [3] S. Roy and I. J. Cox, "Maximum-flow formulation of the N-camera stereo correspondence problem," in *Proceedings of the 6th International Conference on Computer Vision*, pp. 492–499, January 1998.
- [4] 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.
- [5] J. Sun, N.-N. Zheng, and H.-Y. Shum, "Stereo matching using belief propagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 7, pp. 787–800, 2003.
- [6] M. Amintoosi, M. Fathy, and N. Mozayani, "A fast image registration approach based on SIFT key-points applied to super-resolution," *Imaging Science Journal*, vol. 60, no. 4, pp. 185–201, 2012.
- [7] M. W. Jian, J. Y. Dong, and J. Ma, "Image retrieval using wavelet-based salient regions," *Imaging Science Journal*, vol. 59, no. 4, pp. 219–231, 2011.
- [8] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proceedings of the Alvey Vision Conference*, p. 50, 1988.
- [9] 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, July 2001.
- [10] T. Lindeberg, "Feature detection with automatic scale selection," *International Journal of Computer Vision*, vol. 30, no. 2, pp. 79–116, 1998.
- [11] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [12] A. Stansk and O. Hellwich, "Spiders as robust point descriptors," in *Pattern Recognition*, W. G. Kropatsch, R. Sablatnig, and A. Hanbury, Eds., vol. 3663 of *Lecture Notes in Computer Science*, pp. 262–268, Springer, Berlin, Germany, 2005.
- [13] K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Computer Vision—ECCV 2002*, vol. 2350, pp. 128–142, Springer, 2002.
- [14] F. Jurie and C. Schmid, "Scale-invariant shape features for recognition of object categories," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 2, pp. II-90–II-96, Washington, DC, USA, July 2004.
- [15] T. Tuytelaars and L. van Gool, "Wide baseline stereo matching based on local, affinely invariant regions," in *Proceedings of the 11th British Machine Vision Conference (BMV '00)*, vol. 2, pp. 412–425, 2000.
- [16] 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.
- [17] R. Dragon, M. Shoaib, B. Rosenhahn, and J. Ostermann, "NF-featuresno-feature-features for representing non-textured regions," in *Computer Vision—ECCV 2010*, pp. 128–141, Springer, 2010.
- [18] H. Tao, H. S. Sawhney, and R. Kumar, "A global matching framework for stereo computation," in *Proceedings of the 8th International Conference on Computer Vision (ICCV '01)*, pp. 532–539, July 2001.
- [19] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.
- [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] A. F. Stalder, G. Kulik, D. Sage, L. Barbieri, and P. Hoffmann, "A snake-based approach to accurate determination of both contact points and contact angles," *Colloids and Surfaces A: Physicochemical and Engineering Aspects*, vol. 286, no. 1–3, pp. 92–103, 2006.
- [22] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.
- [23] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 341–346, 2001.
- [24] J. Maver, "Self-similarity and points of interest," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1211–1226, 2010.
- [25] M. Smithson, *Statistics with Confidence: An Introduction for Psychologists*, Sage, 2000.