A high-precision localization algorithm by improved SIFT key-points

Yang Yang*, Yixu Song*, Fangwen Zhai*, Zhaozhou Fan†, Yue Meng† and Jiaxin Wang*

*Tsinghua National Laboratory for Information Science and Technology,
Department of Computer Science, Tsinghua University, Beijing, China 100084
†TSTD OPTOELECTRONICS TECHNOLOGY CO., LTD, Beijing, China 100076
Email: yang-yang06@mails.tsinghua.edu.cn, syx@s1000e.cs.tsinghua.edu.cn

***Abstract*—High-precision localization is getting more and more dependent on computer vision techniques. In this paper a novel high-precision localization algorithm based on improved SIFT (Scale Invariant Feature Transform) key-points is presented. First considering the drawback of original SIFT algorithm and the special application background, the proposed algorithm improves the SIFT key-point description and discards the most time-consuming step of SIFT algorithm. Then the new matching strategy and localization strategy are investigated to ensure the stability and precision of localization. Compared with conventional localization algorithm by SIFT key-point, this algorithm increases computation efficiency for about 20% and makes the precision more stable, which reaches 0.1 pixel.**

I. Introduction

With the development of CV (computer vision), more and more industrial applications of CV techniques appeared, such as Image Recognition, Automatic control, Localization, etc. Localization, especially high-precision localization applications such as ACF Bonding (Anisotropic Conductive Film), TAB (Tape Automated Bonding), COG/COF (Chip on Glass/Flex) Mounting, etc, which is one of the most important applications on CV, is even more dependent on the technology development. Compared with traditional localization method, high-precision localization method needs higher precision and more robustness, which does not just require the improvement of CCD (Charge Coupled Device) resolution or mechanical performance. In order to avoid expensive cost in upgrading hardware, a novel high-precision localization algorithm will be investigated in this paper, which can not only give a stable sub-pixel position of the template but also locate the template under any kinds of situations, such as template rotation and scale change, illumination change, even partial occlusion of template.

A. Related work

Template matching is one of the most important localization algorithms. Conventional template matching methods focus on the gray-scale of each pixel and correlation between template image and target image. The major steps of these methods are sliding template image over target image, computing a measure at each pixel and estimating the degree of similarity [7]. The typical measures include MAD (Mean Absolute Distance), SAD (Sum of Absolute Differences), NCC (Normalized Cross Correlation) etc. Since these methods demand sliding over the whole target image, which consumes a lot of computational time, a number of techniques have been investigated with the intent of speeding up them [3], [11], [17].

However, there are other drawbacks of traditional methods, such as they are not invariant to the template image rotation and scale change, their precision just reaches pixel level, etc. Some algorithms try to use the circle projection and scale space to overcome the former problem [16], [21]. For the latter, a few estimation methods are investigated to get higher precision, such as using NCC and LSM (Least Square Method) [19] etc.

Although a lot of ideas attempt to overcome all kinds of drawbacks of traditional methods, their performance do not fulfill the requirements. Therefore, template matching based on features algorithms attract more attentions recently, which can keep localization more stable. These methods steps include extracting the features from images, matching them between template image and target image, locating the template with the feature relations [7].

The image features can be divided into two categories, global features and local features. The global feature algorithms collect the images characteristic and use one vector to represent an image. These features include seven moments of image proposed by Hu in 1962 [8], which is invariant to the image scale change and rotation of less than 45 degree, Fourier-Mellin features implemented with log-polar and Fourier Transform [4] and Gist features implemented with Gabor Transform [20]. However, because of the property of global features, it is difficult to accomplish localization with them.

Local feature algorithms locate key-points in the image and calculate descriptors for them, which are invariant in different images. In this way, a set of key-points corresponds to one image. Using location information of the same key-points in two different images, template localization can be achieved.

The common local features include Harris-Laplace and SIFT (Scale Invariant Feature Transform) features. Mikolajczyk and Schmid combined the Laplace operator and Harris corner detector [6] and acquired Harris-Laplace feature to keep Harris corner invariant in different scale [15]. SIFT feature was proposed by David G. Lowe in 1999 [5]. This method uses

978-1-4244-4131-0/09/$25.00 ©2009 IEEE

DOG (Difference of Gaussian) operator for extramas and HOG (Histogram of Oriented Gradient) for descriptor to extract distinctive features from images. SIFT feature is invariant to image rotation and scale change, illumination change and even some affine distortion. It has been widely used for tasks such as image retrieval [10], [2], image registration [13], object recognition [12], tracking [1], SLAM (Simultaneous Localization and Mapping) [18], etc. In order to reduce the computational time in finding the extramas and calculating the descriptors, integral image and PCA (Principal Component Analysis) methods have been added into SIFT to enhance the performance.

In the paper [22], a high-precision localization algorithm using the original SIFT key-points has been investigated. Although the algorithm [22] can give a correct answer and keep the localization repeatability in the complicated environment, the precision and time cost are not satisfactory in certain circumstance. In this paper, inspired by SIFT algorithm, a novel high-precision localization algorithm has been proposed, which can reduce the time cost and improve the precision.

B. Overview

The rest of the paper is organized as follows: Section II gives a brief description about SIFT and analyzes time complexity of SIFT and matching algorithm, then shows new matching strategy. Section III summarizes some reasons of localization errors in SIFT algorithm, and then discusses some proposals to make up for the errors and improve the precision. The experimental analysis is shown in Section IV; and Section V concludes the paper with some future recommendations.

II. Matching key-points

In this paper, the algorithm exploits improved SIFT key-points for localization. First it must find a way to match the features from template image and target image. In papers [13], [18], SIFT feature matching strategies include K-D Tree and RANSAC (Random Sample Consensus) methods. As a general rule, the matched pairs can be found with K-D Tree first, and then the mismatches eliminated with RANSAC. Considering the efficiency of matching algorithm, Computing SIFT feature is the bottleneck for its performance. In this section, we will use DOG extramas, discard SIFT descriptors and try to propose another new method to achieve matching.

A. Description of SIFT

SIFT algorithm was proposed by David G. Lowe in 1999 [5]. Compared with other image features, SIFT feature is more robust and reliable, which can keep invariant whenever images rotate, change scale, add noise or even have affine transform. The following are the major steps of generating SIFT features:

1) Build scale space
The first step is using Gaussian Filter and Gaussian Pyramid down algorithm to build DOG scale space which is prepared for searching the extramas.
2) Detect the extramas
The extramas are identified as the minima or maxima compared with their 8 neighbors in the same scale and 9 corresponding neighbors at each neighboring scale in this scale space. The points located on the edge or with low contrast should be eliminated. With the nearby information of location, scale and principal curvatures, the sub-pixel location and precise scale of key-point can be acquired.
3) Calculate the orientation
Using HOG of neighborhood block of key-point, one or more orientations are assigned to each key-point.
4) Calculate the descriptor
Each key-point has a sub-pixel location, a scale and an orientation. Rotate its neighborhood block to its orientation, cut the block into 4×4 slices and compute HOG of each slice. The 4×4 HOGs form its descriptor.

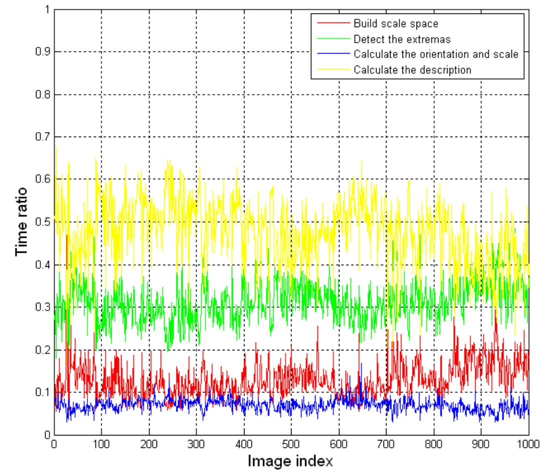


Fig. 1. The each step time-consuming of SIFT algorithm

B. Analysis of SIFT

For testing time cost in each step of SIFT algorithm, we picked 1000 images from internet randomly. Fig. 1 shows step 4 costs the most time, about 45% of total time. Since descriptor is built at every key point, this step is closely related to the feature number. Step 2 gets the second place, spending 30% of total time. This step search over the whole scale space for the extramas, so time cost is mainly dependent on the image resolution and scale space size. Compared to the former two steps, the first and third steps cost less time, just 17% and 8% of total time respectively.

Several techniques have been investigated to enhance this algorithm. [14] tried to reduce time cost of the first two step of SIFT algorithm by integral image. [9] attempted to shorten the length of descriptor using PCA method to speed up matching method. In this paper, for the special purpose, we will abandon the feature descriptor, just use its location, scale and orientation to match key-points.

C. Matching strategy

The new strategy needs the DOG key-point, including its location, scale and orientation, but not its vector descriptor.

Given two images I_{tmp} and I_{tar} representing template image and target image respectively, through the first three steps of SIFT algorithm, two sets of key-points are:

$$
\begin{aligned}
F(I_{tmp}) &= \{f_1^{tmp}, f_2^{tmp}, \ldots, f_m^{tmp}\} \\
F(I_{tar}) &= \{f_1^{tar}, f_2^{tar}, \ldots, f_n^{tar}\}.
\end{aligned}
$$

Each key-point $f_i^{tar/tmp}$ consists of four elements: x and y are its position on X-axis and Y-axis respectively, scl is its scale and ori is its orientation. The algorithm goal is getting a correct matched pair set $P_{match} = \{(f_{a_1}^{tmp}, f_{b_1}^{tar}), (f_{a_2}^{tmp}, f_{b_2}^{tar}), \ldots, (f_{a_L}^{tmp}, f_{b_L}^{tar})\}$. Our new matching strategy is as follows:

for $\forall f_i^{tmp} \in F(I_{tmp})$ **do**
  for $\forall f_j^{tar} \in F(I_{tar})$ **do**
    compute the orientation difference between f_i^{tmp} and f_j^{tar} is $ori = f_i^{tmp}.ori - f_j^{tar}.ori$
    compute the scale difference between f_i^{tmp} and f_j^{tar} is $scl = f_i^{tmp}.scl - f_j^{tar}.scl$
    transform $F(I_{tmp})$ to $F'(I_{tmp})$ by ori and scl as following:
    for $\forall f_k^{tmp} \in F(I_{tmp})$ **do**
      $f_k'^{tmp}.x = scl * (f_k^{tmp}.x * \cos(ori) - f_k^{tmp}.y * \sin(ori))$
      $f_k'^{tmp}.y = scl * (f_k^{tmp}.x * \sin(ori) + f_k^{tmp}.y * \cos(ori))$
    end
    $matchNum_{ij} = 0$
    for $\forall f_k'^{tmp} \in F'(I_{tmp})$ **do**
      if *the position of* f^{tar} *and* $f_k'^{tmp}$ *are equal or similar* **then**
        $matchNum_{ij}++$
      end
    end
  end
end
find the maximum $matchNum_{ij}$.
if $matchNum_{ij}^{max} > threshold$ **then**
  transformation from f_i^{tmp} to f_j^{tar} is the best matching, use it to acquire the rest matched pairs.
else
  There is no matched template in target image
end

Fig. 2 illustrates how to transform key-points in template image by the scale and orientation difference and how to find the correct match.

From the procedures above, time complexity of the algorithm is $O(nm^2 \log(n))$, where $O(\log(n))$ is the time complexity of binary searching. If the key-points number is large, this algorithm will take long time. Fortunately, just for the high-precision localization application, there is considerable room for optimization. For example, in real time there is always a tiny change to template image. So many pruning methods can be added to reduce computational complexity.

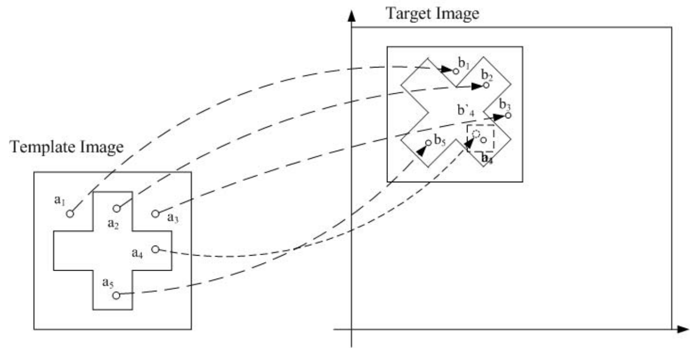
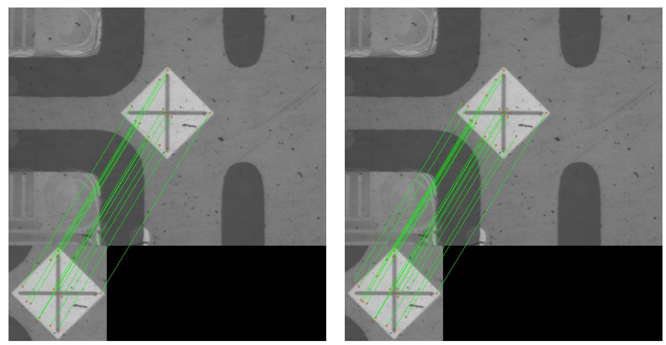


Fig. 2. Match key-points of template image to key-points of target image



(a) with 14 matched pairs (b) with 19 matched pairs

Fig. 3. Comparative result between original matching strategy and new matching strategy. (a) is the matching result of original strategy with 14 matched pairs and (b) is the result of new one with 19 matched pairs.

Compared with original SIFT matching method, new matching strategy has another advantage besides high efficiency. Fig. 3 shows new matching strategy can give more matched pairs. For the linear transformation and localization algorithm with LSM [22], these extra matched pairs may bring errors because of their imprecision on their locations. Otherwise, for the new localization algorithm below, more matched pairs means higher precision.

III. Locate Template Image

After acquiring matched pair set P_{match}, we use it to locate template image. Existent method builds a projection matrix by rotation angle, scale change and coordinates offset, and finds the sub-pixel position with LSM [22]. However, owing to the drawbacks of original SIFT algorithm, localization result is instable in some cases. In order to solve these problems, a new method will be investigated in this section.

A. Localization with LSM

Given matched pair set P_{match}, a projective matrix can be built with the key-point coordinates. It reflects how template image transform into the part of target image by rotation, changing scale or shift. The linear equation is

$$s\begin{bmatrix}\cos\theta & -\sin\theta & a\\ \sin\theta & \cos\theta & b\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}x_i^{tmp}\\ y_i^{tmp}\\ 1\end{bmatrix}=\begin{bmatrix}x_i^{tar}\\ y_i^{tar}\\ 1\end{bmatrix}\tag{1}$$

where

$$\begin{aligned}x_i^{tar}=f_i^{tar}.x,\ y_i^{tar}=f_i^{tar}.y\\ x_i^{tmp}=f_i^{tmp}.x,\ y_i^{tmp}=f_i^{tmp}.y\end{aligned}$$

Changing (1) into

$$\begin{bmatrix}x_1^{tmp} & -y_1^{tmp} & 1 & 0\\ \vdots & \vdots & \vdots & \vdots\\ x_L^{tmp} & -y_L^{tmp} & 1 & 0\\ y_1^{tmp} & x_1^{tmp} & 0 & 1\\ \vdots & \vdots & \vdots & \vdots\\ y_L^{tmp} & x_L^{tmp} & 0 & 1\end{bmatrix}\begin{bmatrix}sc\\ ss\\ sa\\ sb\end{bmatrix}=\begin{bmatrix}f_1^{tar}.x\\ \vdots\\ f_L^{tar}.x\\ f_1^{tar}.y\\ \vdots\\ f_L^{tar}.y\end{bmatrix}\tag{2}$$

where

$$\begin{aligned}sc=s\cdot\cos\theta, ss=s\cdot\sin\theta\\ sa=s\cdot a, sb=s\cdot b\end{aligned}$$

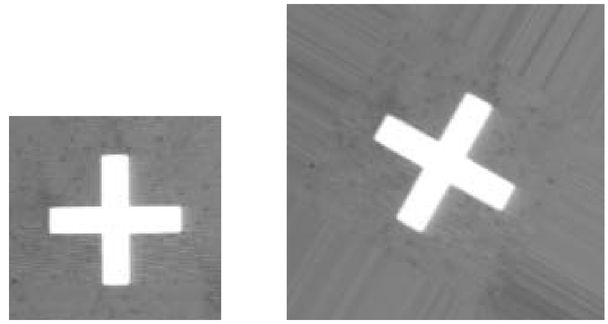
and solving (2) by LSM, values of $\{a,b,\theta,s\}$ are obtained. The coordinate offset a and b can be regarded as sub-pixel position of template image.

B. Error analysis

A lot of experiments show applying algorithm in Section III-A into SIFT key-point pairs brings a bad repeatability in some cases. Since the matching and localization strategies have been proved correct, our attention focuses on original SIFT algorithm and image itself. Through analysis, we summarize the following two potential reasons:

- For the repetitive image acquisition, each image may look the same overall. However, after comparing, there always exist some regions in these images, where there are a few differences in the gray-scale or gradient. The SIFT key-point location depends on its neighborhood information. So more or less, the difference of those regions will bring some changes on the locations.
- In the second step of SIFT algorithm, in order to get the sub-pixel position of key-point, SIFT fits a 3D quadratic function to determine the interpolated location of the maximum as accurate key-point location. This approach uses the Taylor expansion (up to the quadratic terms) of scale space function which may lead to imprecision. Because the final position should be zoomed in according to the key-point scale, the imprecision will be further magnified.

The two reasons above may make key-point location imprecise. Therefore, conventional SIFT localization algorithm just depends on each key-point location, which will influence the eventual localization precision. The first problem above can be solved just by improving industrial environment. For the latter, we will try to relocate the key-points to reduce the imprecision.



(a) (b)

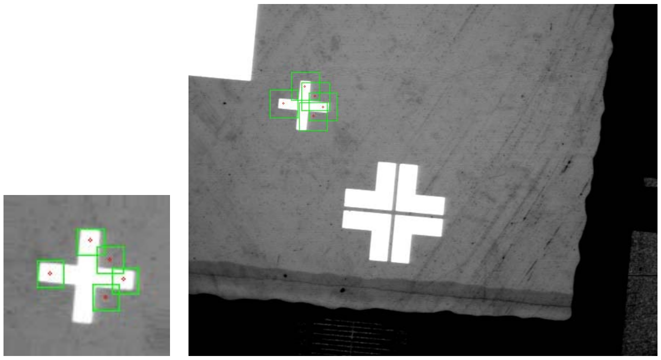
Fig. 4. Original template image (a) and new template image (b)

C. Localization strategy

With the given matched pair set P_{match}, we attempt to obtain another matched pair set P'_{match} which will be more precise. Considering imprecision caused by SIFT algorithm, traditional template matching algorithm based on gray-scale will be used to relocate the SIFT key-point. Because traditional algorithm can not locate the templates which have some transformations, images should be preprocessed before localization. Detailed procedures are as follows:

1) Use localization algorithm in Section III-A to P_{match}, compute θ and s which are the orientation difference and scale difference between template image and target image respectively.
2) Construct new template image I'_{tmp} through the transformation of original template image I_{tmp} to make I'_{tmp} have the same orientation and scale with I_{tar}. See Fig. 4.
3) For each key-point f_i^{tmp} in I_{tmp}, calculate $f_i'^{tmp}$ which is mapped point of f_i^{tmp}. Take a neighborhood region of $f_i'^{tmp}$ as its description in I'_{tmp}, Fig. 5(a) and take another region in I_{tar} as searching region according to f_i^{tar}, Fig. 5(b). Use the neighborhood regions of $f_i'^{tmp}$ as templates and locate them in I_{tar} with regional template matching algorithm, marked the locations as k_i^{tar}. Here the size of description region is always 1/36 of template image for higher precision and searching region is double size of description region.
4) Compute original positions of each regional template in I_{tmp}, mark the positions as k_i^{tmp}. So new matched pair set $P'_{match}=\{(k_1^{tmp},k_1^{tar}),\ldots,(k_L^{tmp},k_L^{tar})\}$ is constructed. Use the method in Section III-A to P'_{match} and calculate the final result.

In the new localization strategy, the description of key-point location has been changed. In original SIFT algorithm, location of key-point is described by its coordinates. Here square region is proposed as its description. Compared to coordinate description, description by square region has less effect on single pixel change and it is more stable and precise.



(a) (b)

Fig. 5. Region description for each key-point within green box (a) and searching region for each key-point in target image (b)

IV. Experimental analysis

A. Experiments configuration

The key-points extraction algorithm uses Lowe's implementation and all the algorithms are implemented in C++ on the computer with Inter Pentium 4 CPU 3.06GHz, 1G memory and OS Windows XP SP3. In order to ensure the authenticity, these experimental images are all from engineering samples.

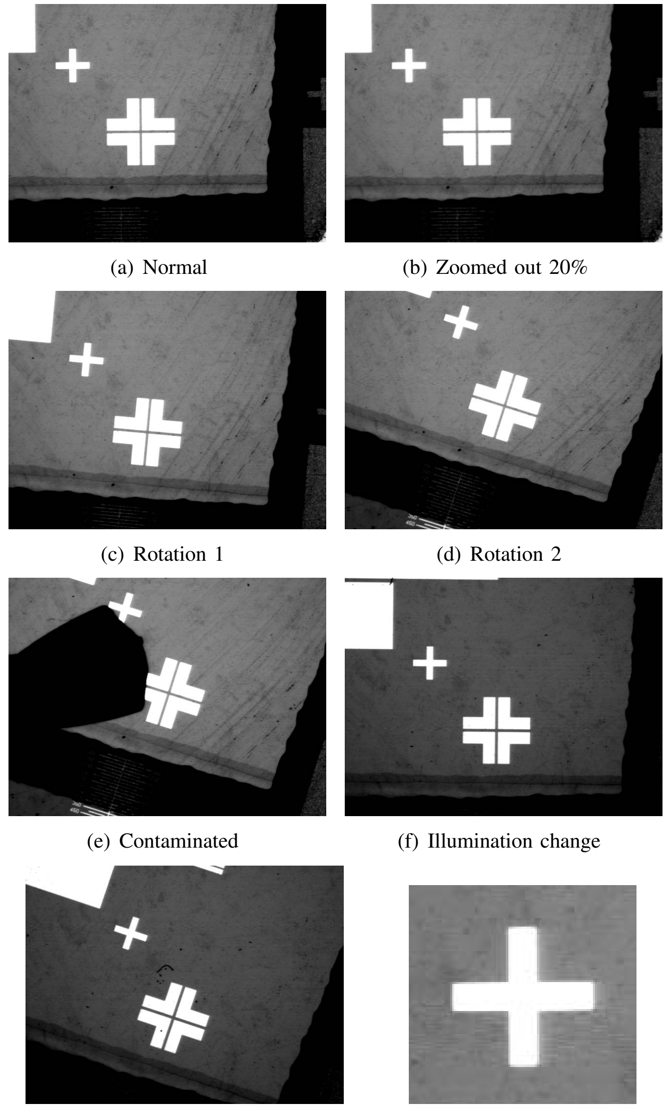
To test the localization repeatability, we perform the experiments as follows: keep the platform and camera fixed and take 10 images continuously as a test group at the same position. For each group, the template image can be located in each image. Here we use standard deviation and average time cost to evaluate the performance. The standard deviation is

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \overline{x})^2}$$

Where $\overline{x}$ is the arithmetic mean of the values x_i, N is the sample number. In statistics, σ is a measure of the variability or dispersion of a data set or a probability distribution. A low standard deviation indicates that the data points tend to be very close to the same value (the mean), while high standard deviation indicates that the data are "spread out" over a large range of values. It is fit for measuring the dispersion of the locations.

In order to compare the algorithms performance, traditional localization method, SIFT localization method and improved SIFT localization method are implemented. TLM (Traditional Localization Method) is based on NCC. Because its precision just reaches to pixel-level, we attempt to obtain higher precision through the linear fitting of the neighborhood NCC value. SLM (SIFT Localization Method) is based on SIFT features, K-D Tree and RANSAC methods. ISLM (Improved SIFT Localization Method) is introduced above.

In this experiment, we got target images under seven situations, see Fig. 6. Fig. 6(a) is normal image. Fig. 6(b) is zoomed out by Fig. 6(a). Fig. 6(c) and 6(d) are rotated with different angles. Fig. 6(e) shows that the target image is covered. Fig. 6(f) reflects the illumination changes and Fig. 6(g) shows target image is rotated in such illumination environment. We use the three algorithms mentioned above to locate template image Fig. 6(h) in these target images and compared their performance.



(a) Normal (b) Zoomed out 20%

(c) Rotation 1 (d) Rotation 2

(e) Contaminated (f) Illumination change

(g) Illumination change and rotation (h) Template image

Fig. 6. The comparison of the target images in different situations

B. Results and Analysis

In Table I, T and σ are the average time cost and location standard deviation of the three methods including traditional localization method, SIFT localization method and improved SIFT localization method. $N_{keypoint}$ is average key-point number in a group target image. $*$ means wrong localization.

From the data in Table I, the performance of TLM is perfect in normal and illumination change situations. Multi-resolution technique shortens the running time and the precision reaches to 0.01 pixel. That is why it can be used in ISLM to relocate the key-point. However, it does not overcome its fatal drawback. It fails to deal with transformations of test images.

TABLE I
PERFORMANCE COMPARISON (FIG. 6)

Image	Image Resolution	$N_{keypoint}$	T_{TLM}(sec)	σ_{TLM}(pixel)	T_{SLM}(sec)	σ_{SLM}(pixel)	$T_{ISLM}(sec)$	σ_{ISLM}(pixel)
Fig. 6a	640×480	94.8889	0.2346	0.0121	0.6476	0.1720	0.5993	0.0491
Fig. 6b	512×384	75.0000	*	*	0.5262	0.1437	0.3907	0.0484
Fig. 6c	640×480	94.2222	*	*	0.6858	0.0773	0.5469	0.1042
Fig. 6d	640×480	94.4444	*	*	0.6997	0.2048	0.5660	0.1096
Fig. 6e	640×480	97.0000	*	*	0.7500	0.0714	0.6139	0.0989
Fig. 6f	640×480	76.2222	0.2314	0.0092	0.5782	0.0598	0.5624	0.0714
Fig. 6g	640×480	57.8889	*	*	0.5650	0.0600	0.4866	0.0562

Compared with SLM, ISLM reduces about 20% computational time. Because ISLM is closely bound up with the key-point number, the time cost difference between SLM and ISLM is not stable. Considering the precision, although the average precision of ISLM and SLM has reached 0.1 pixel, the precision of SLM fluctuates around the mean more greatly. It reached 0.2 pixel to the group Fig. 6(d). Obviously ISLM works better in this respect.

However, we can see the precision of ISLM is a little bad for the rotated image. Maybe the errors are caused by localization strategy in Section III-C. In the second step, we need to construct a new template image from the old one. The transformation algorithm by NNI (Nearest Neighbor Interpolation) algorithm is a little defective, especially for image rotation, which may influence the later localization.

V. CONCLUSION

This paper investigated a novel high-precision localization method by improving SIFT key-points. For this special application background, the new algorithm improves SIFT feature, discards the most time-consuming step and develops the new matching method. The new matching strategy gives more matched pairs with lower time cost, compared to conventional SIFT matching algorithm. We analyze the reasons for errors in conventional SIFT localization and propose new localization strategy using the region description, which is more precise. Through the experiment comparison, new algorithm is more stable and its computational time decreases about 20% towards original SIFT localization algorithm.

However, there are still some shortcomings need to overcome. In Section III-C, the square region description is proposed, which is more stable to the repetitive images and it is used to relocate the key-point. But in the last step, it still uses the key-point coordinate description rather than the region description to compute the location. The future work will focus on how to describe key-points, abandon coordinate description and acquire more precise location.

REFERENCES

[1] S. Battiato, G. Gallo, G. Puglisi, and S. Scellato. Sift features tracking for video stabilization. *In Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference*, pages 825–830, 2007.

[2] J. J. Foo and R. Sinha. Using redundant bit vectors for near-duplicate image detection. *12th International Conference on Database Systems for Advanced Applications*, 4443:472–484, 2008.

[3] M. Gharavi-Alkhansari. A fast globally optimal algorithm for template matching using low-resolution pruning. *IEEE Trans. Image Process.*, 10:526C533, 2001.

[4] P. Ghosh, E. D. Gelasca, K. Ramakrisnan, and B. Manjunath. *Duplicate Image Detection in Large Scale Databases*. Book Chapter in Platinum Jubilee Volume, Oct 2007.

[5] D. G.Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[6] C. Harris and M. Stephens. A combined corner and edge detector. *In Proc. 4th Alvey Vision Conference, Manchester*, pages 189–192, 1988.

[7] C. Heipke. Overview of image matching techniques. *In OEEPE Workshop on the Application of Digital Photogrammetric Workstations*, pages 173–189, 1996.

[8] M. K. Hu. Visual pattern recognition by moment invariants. *Information Theory, IEEE Transactions on*, 8:179–187, 1962.

[9] Y. Ke and R. Sukthankar. Pca-sift: a more distinctive representation for local image descriptors. volume 2, pages II–506–II–513 Vol.2, 2004.

[10] Y. Ke, R. Sukthankar, L. Huston, Y. Ke, and R. Sukthankar. Efficient near-duplicate detection and sub-image retrieval. pages 869–876, 2004.

[11] J. P. Lewis. Fast template matching. *in Proc. Vision Interface*, pages 120–123, 1995.

[12] D. G. Lowe. Object recognition from local scale-invariant features. *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, 2:150–1157, 1999.

[13] B. Matthew and D. G.Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74:59–73, 2007.

[14] G. Michael, G. Helmut, and B. Horst. Fast approximated sift. *Asian Conference on Computer Vision, Hyderabad, India*, pages 918–927, 2006.

[15] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615–1630, 2005.

[16] A. Rosenfeld and G. J. Vanderburg. Coarse-fine template matching. *Systems, Man and Cybernetics, IEEE Transactions on*, 7:104–107, 1977.

[17] H. Schweitzer, J. W. Bell, and F. Wu. Very fast template matching. *in Proc. 7th Eur. Conf. Computer Vision IV*, page 358C372, 2002.

[18] S. Se, D. Lowe, and J. Little. Global localization using distinctive visual features. pages 226–231, 2002.

[19] M. Sun, D. Li, Z. Yu, and G. Zong. Locating cross-shaped image feature with subpixel accuracy based on ncco. *Journal of Beijing University of Aeronautics and Astronautics*, C-26:780–784, 2005.

[20] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large image databases for recognition. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[21] G. J. Vanderburg and A. Rosenfeld. Two-stage template matching. *Computers, IEEE Transactions on*, C-26:384–393, 1977.

[22] Y. Yang, Y. Song, M. A. Shaikh, and J. Wang. A high-precision template localization algorithm using sift keypoints. *Computer and Information Sciences, 2008. ISCIS '08. 23rd International Symposium on*, pages 1–6, 2008.