Parameter Decision for View Simulation in ASIFT Concerning the Affine Distortion Error

Joohyuk Yum[1] and Hyuk-Jae Lee[2]
Electrical and Computer Engineering, Inter-University Semiconductor Research Center,
Seoul National University, Sillim-dong, Gwanak-gu, Seoul 151-744, Korea.
E-mail: [1]joohyuk86@capp.snu.ac.kr, [2]hyuk_jae_lee@capp.snu.ac.kr

Abstract: ASIFT (affine invariant extension of SIFT) is fully affine invariant by simulating all image views obtained by varying viewpoints of the camera. This paper proposes an algorithm, which searches proper viewpoints for ASIFT algorithm in order to improve the robustness for the affine distortion. In this searching process, the affine distortion error proposed in this paper is used as the index of the distortion error caused by the viewpoint change. The ASIFT algorithm with the proposed viewpoints selection establishes correct matches by 1.18 times more than the conventional ASIFT algorithm, even though the computational complexity of the proposed method and the conventional method is same.

Keywords-- **ASIFT, Affine transform, Error modeling**

1. Introduction

In many computer vision applications, feature matching is a basic step for object recognition and tracking. Among various research works for feature matching, scale-invariant feature transform (SIFT) algorithm, proposed by Lowe in [1], is one of the most widely used local features, because it is scale, and rotation invariant. However, it cannot establish correct correspondences if an axis of the camera is changed. In order to guarantee the matching accuracy against the viewpoint change, an affine invariant extension of SIFT (ASIFT) is proposed by Morel *et al.* [2]. ASIFT simulates many images obtained by varying viewpoints of the camera. In this process, the deformation caused by the slanted viewpoint is simulated by the affine transform, and it enforces the robustness of ASIFT features for the viewpoint change.

This paper proposes the viewpoint search algorithm, which searches the proper viewpoints for ASIFT in order to improve the endurance for affine distortion. The framework proposed by Morel *et al.* is efficient to deal with the affine distortions, but the method to select viewpoints needs additional improvement. This paper presents the affine distortion error which estimates the distance of pixel movement by the affine distortion. The affine distortion error is used by the viewpoint search algorithm, which selects proper viewpoints. In the experimental result, the proposed method finds correct matches 1.18 times more than the conventional method with same computational load.

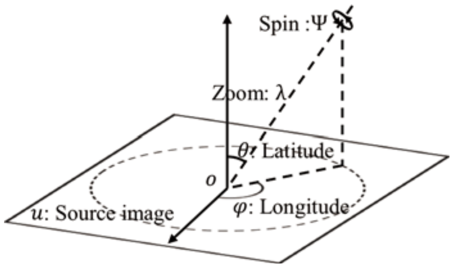


Figure 1 A camera pose interpretation of the affine decomposition

2. Previous works

2.1 Simulation of Affine distortion in ASIFT

ASIFT algorithm is an extension of SIFT, which is fully affine invariant [2]. In order to achieve this property, at first, ASIFT simulates all possible affine distortions caused by the change of camera optical axis orientation from a frontal position. At second, all simulated images are processed by SIFT algorithm. Then extracted ASIFT features can find correct correspondences between two images taken with different viewpoints.

Figure 1 shows a camera pose interpretation of the affine decomposition. The camera pose is represented on the hemispherical coordinates, of which the center o meets the center position of a source image u. An affine distortion caused by the change of camera pose from a frontal position is represented by latitude (θ), longitude (φ) as driven by (1). R_φ is a rotation matrix and $T_{1,1/t}$ is a scaling matrix represented by a diagonal matrix of which first eigenvalue is 1 and second eigenvalue is $1/t$.

$$A = T_{1,1/t}R_\varphi = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{t} \end{bmatrix} \begin{bmatrix} cos\varphi & -sin\varphi \\ sin\varphi & cos\varphi \end{bmatrix}, t = 1/cos\theta \quad (1)$$

The scene image observed from the camera pose is given by (2). p is a pixel position in the affine transformed image, and the corresponding pixel position is $q = (q_x, q_y)$. W and H mean width and height of u respectively.

$$p = Aq = T_{1,1/t}R_\varphi q, 0 \le q_x \le W \; and \; 0 \le q_y \le H \quad (2)$$

Figure 2 shows the procedure of image transform by (2) step by step. R_φ rotates a source image shown in Figure 2 (a) by φ. The rotated image is represented in Figure 2 (b). Then $T_{1,1/t}$ down-samples the rotated image vertically. The result of the affine transform is shown in Figure 2 (c).

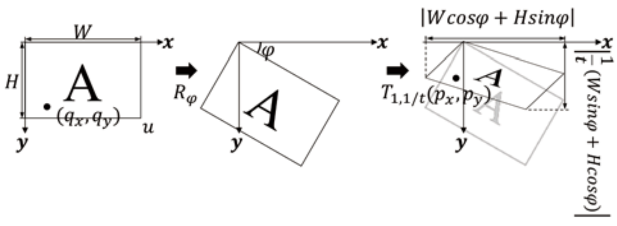


Figure 2 Procedure of the operation with original affine transform

TABLE I

PRE-DEFINED AFFINE TRANSFORM PARAMETERS OF CONVENTIONAL ASIFT

Latitude (Degree)	**Longitude (Degree)**														
0	(0) **0**														
45	(1) **0**	(2) **51**	(3) **102**	(4) **153**											
60	(5) **0**	(6) **36**	(7) **72**	(8) **108**	(9) **144**										
69	(10) **0**	(11) **25**	(12) **51**	(13) **76**	(14) **102**	(15) **127**	(16) **153**								
76	(17) **0**	(18) **18**	(19) **36**	(20) **54**	(21) **72**	(22) **90**	(23) **108**	(24) **126**	(25) **144**	(26) **162**					
80	(27) **0**	(28) **13**	(29) **25**	(30) **38**	(31) **51**	(32) **64**	(33) **76**	(34) **89**	(35) **102**	(36) **115**	(37) **127**	(38) **140**	(39) **153**	(40) **165**	(41) **178**

In order to obtain fully affine-invariance, ASIFT algorithm simulates affine deformations densely using pre-defined θ and φ. The tilt range of t=1/cos θ is [t_{min}, t_{max}] = [1, $4\sqrt{2}$] and the angle range of φ is [φ_{min}, φ_{max}] = [0º, 180º]. The sampling step are $\Delta t = \sqrt{2}$ and $\Delta\varphi$ = 72º/t. The affine transform matrices satisfying this conditions are 42 as shown in Table I. A number in parentheses is viewpoint index that is assigned to each simulation.

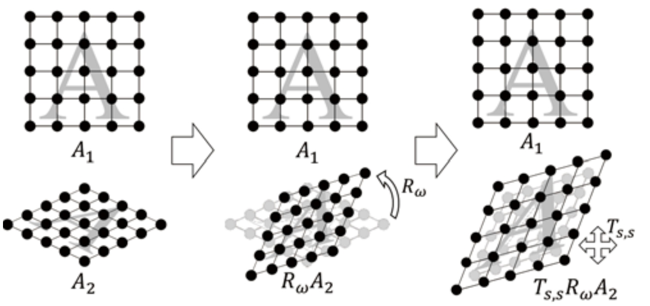
2.2 SIFT feature generation

The second procedure of ASIFT is SIFT feature generation, which is composed of key-point detection, and descriptor generation. In the first step, a Gaussian pyramid is generated by Gaussian filtering a source image with different sigma values. Then SIFT key-points are detected by finding the extremes of Gaussians (DoG). For a detected key-point, a descriptor vector is calculated and then assigned to the key-point. The gradient magnitude and orientation (GMO) of a local-patch, which consists of pixels around a key-point, are calculated. Then a descriptor that is a set of histograms containing the calculated gradient orientation is generated. Further details of the SIFT algorithm are available in the previous literature [1].

3. Determination of the proper position of viewpoints for ASIFT algorithm

3.1 Definition of affine distortion error

There is an image taken in frontal view, and another image is taken in another view. If the difference of angle between two viewpoints is small, the movement of each pixel by the affine distorion is also small, then SIFT can find corespondences between them. On the otehr hand, when the difference of angle is large, the movements of pixels are large, and SIFT cannot match two images. In summary, the reason that SIFT cannot match images in difference views is the movement of pixels cause by the viewpoint change. In order to measure it, this paper uses the sum of distances of pixel movements, which is called distance error $e(A_1, A_2)$, and it is calculated by (3). In (3), N means the length of an local image patch, and A_1 and A_2 are affine transform matrices corresponding two viewpoints. $dist(a, b)$ returns the Euclidean distance between point a and b.



(a) (b) (c)

Figure 3 Procedure of finding minimum distance error between $A_1 = I$ and A_2 in terms of rotation parameter ω and scaling parameter s.

$$e(A_1, A_2) = \sum_{j=-\frac{N-1}{2}}^{(N-1)/2} \sum_{i=-\frac{N-1}{2}}^{(N-1)/2} dist(A_1 q, A_2 q)\,, q = (i, j) \quad (3)$$

SIFT algorithm is scale, and rotation invariant, so it finds correspondence when the source image is rotated or scaled with same horizontal and vertical ratio. The equation (3) does not consider this properties of SIFT algorithm, so the distance error cannot be used for estimating the endurance of SIFT for the viewpoint change. Therefore, this paper proposes affine distortion error $Err(A_1, A_2)$, which is given by (4). In (4), scaling parameter s and rotation parameter ω is used for finding the minimum distance error for the matrices A_1 and A_2 in order to consider the properties of SIFT.

$$Err(A_1, A_2) = \min_{s,\omega} e\big(A_1, T_{s,s} R_\omega A_2\big)\,, 0 < s \le 2, 0 \le \omega \le 2\pi \quad (4)$$

Figure 3 shows an example of the procedure of (4). In figure 3 (a), there are two transformed images of an NxN image by A_1 and A_2 respectively. In this figure, A_1 is indentity matrix. The image transformed by A_2 in figure 3 (a) is rotated by R_ω as shown in figure 3(b). In this case, ω minimizes $e(A_1, R_\omega A_2)$. SIFT calculates dominant orientation of SIFT features, and local-patches used for generating SIFT descriptors are normalized by the dominant orientations [1]. By this procedure, SIFT algorithm could find the angle ω minimizing the distance error. Equation (4) also finds ω minimizing the distance error in order to consider the rotational invariance of SIFT algorithm. Figure 3 (c) shows a case that the distance error between A_1 and A_2 is minimized in terms of ω and s. SIFT algorithm generates Gaussian pyramid and extracts features among various scales [2]. In this process, SIFT algorithm achieves scale invariance.

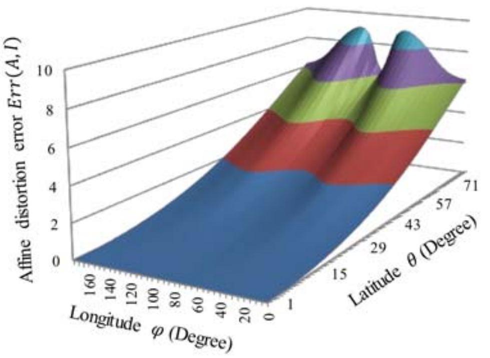


Figure 4 Affine distortion error model between frontal view and various viewpoints.

Equation (4) finds s minimizing the distance error in order to consider scale invariance of SIFT. As a result, the affine distortion error calculated by (4) considers the two properties of SIFT algorithm, and it can be used as an index of distortion error for SIFT caused by the viewpoint change.

Figure 4 shows an affine disrotion error caused by the change of camera pose from a frontal position. It is calculated using (4). A_1 is set by indentity matirx and A_2 is represented by various latitude and longitude indicated by each axis.

3.2 Strategy for searching viewpoints concerning affine distortion error and computational load

This paper proposes an algorithm, which selects viewpoints going to be simulated by ASIFT algorithm concerning the proposed affine distortion error and the computational load of ASIFT algorithm. Figure 5 presents an interpretation about the search algorithm. The black circles in figure 5 are selected viewpoints denoted as v_k. k indicates selection order. The white circle is a viewpoint candidate is going to be tested whether it satisfies some constraints or not. At first this algorithm selects v_0 as an initial viewpoint. Then, this algorithm searches a viewpoint satisfying three constraints explained below.

At first, the algorithm searches from low latitude to high latitude. Latitude of a viewpoint is in inverse proportion to the number of pixels of the simulated image for the viewpoint. The data size of the simulated image is related to the robustness of SIFT algorithm. Because SIFT algorithm generates enough features when the number of pixels of the image also is enough. Therefore, if there are two viewpoint candidates, the candidate with low latitude is selected preferentially.

Secondly, if there is a value of affine distortion error between selected viewpoint and a viewpoint candidate, which is equal or larger than the pre-defined error threshold (Err_{thrs}), the candidate is selected. In figure 5, the notation $t(v_k)$ returns the affine transform matrix corresponding the v_k. In order to verify whether v_c satisfies this contraint or not, the values of affine transform error between v_c and viewpoints from v_0 to v_i are calculated. If all error values are smaller than Err_{thrs}, the candidate is not selected. On the other hand, if there is one or some error values, which are larger than Err_{thrs}, this candidate is selected.

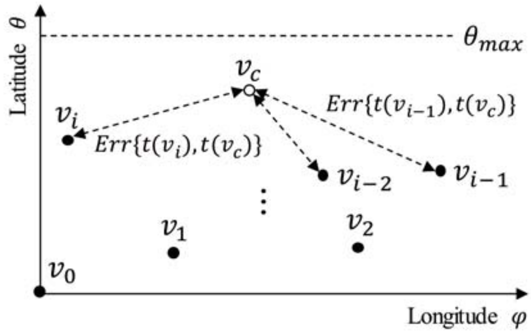


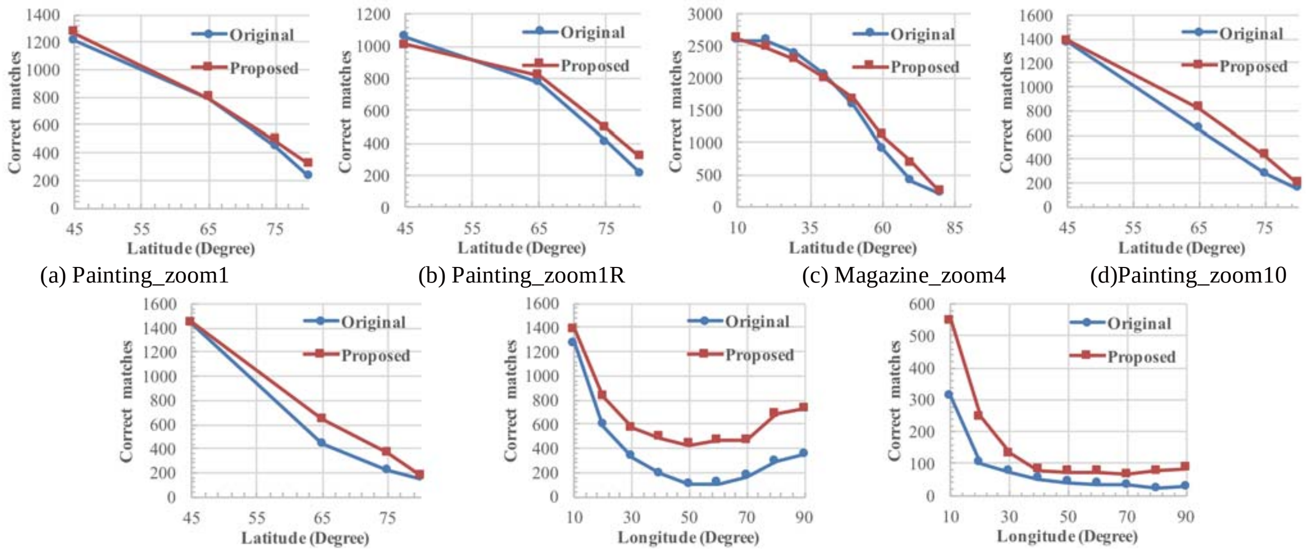
Figure 5 An interpretation about the second contraint of the viewpoint search algorithm

Lastly, computational load of ASIFT algorithm with the selected viewpoints is checked whether it is reasonable or not. If Err_{thrs} is extremely low, too many viewpoints are selected by the search algorithm. It enhances the robustness of ASIFT algorihtm for affine distortion, but too much computing power is required. Therefore, the proposed search algorithm checkes total computational load of ASIFT, which increases in proportion to the number of pixels of simulated images. The search algoritm accumuates the number of the pixels for all selected viewpoints in order to compare with the total number of pixels of conventional ASIFT algorithm. There is an viewpoint candidate satisfying the first and second contraints and latitude of it is larger than the pre-defined maximum latitude (θ_{max}). If the total number of simulated pixels is already larger than that of conventional ASIFT, the search algorithm does not select it and returns the selected viewpoints. On the other hand, if the total number of simulated pixels is smaller than that of conventional ASIFT, the algorithm returns false, and re-starts with new Err_{thrs} decreased compared with previous one. If the total number of pixels already larger than that of conventional ASIFT and the latitude of the last selected viewpoint is smaller than θ_{max}, the search algorithm also return false, and re-starts with increased Err_{thrs}.

4. Experimental results

In order to evaluate the matching accuracy of ASIFT with the selected viewpoints by the proposed search algorithm, experiments are performed. The information about the selected viewpoints by the viewpoint search algorithm is shown in Table II, with N=5, Err_{thrs}=2.9, and θ_{max}=85º. In order to evaluate accuracy, test images proposed by Morel *et al.* in [2] are used and the number of correct matches are counted by the software proposed by Mikolajczyk. *et al.* in [3]. The numbers of correct matches are presented in figure 6. As shown in this figure, the proposed method finds more correct matches than the original ASIFT for almost cases. As a result, the average number of correct matches of the proposed is 1.18 times larger than that of conventional ASIFT, although the computational load of two methods are same.

The ASIFT with the selected viewpoints by proposed search algorithm finds more correct matches than the original ASIFT. Because the selected viewpoints are well distributed than the previous algorithm in terms of affine distortion error. For example, when the first image (φ=0º)



(a) Painting_zoom1 (b) Painting_zoom1R (c) Magazine_zoom4 (d)Painting_zoom10

(e) Painting_zoom10R (f) Magazine_tilt2 (g) Magazine_tilt4

Figure 6 The comparison of the correct matches between the conventioanl ASIFT algorithm (Original) and the ASIFT with the viewpoints selected by the proposed algorithm (Proposed)

and last image (φ =90º) of Magazine_tilt4 test set are matched, there are 31 combinations of viewpoints between the first image and the last image in the proposed algorithm, which find correct matches. On the other hand, in the original ASIFT, there are only 5 combinations of viewpoints establishing correct matches. This property is observed for almost matching operations.

5. Conclusion

This paper proposes an algorithm, which searches proper viewpoints for ASIFT algorithm in order to enforce the robustness for the viewpoint change. To this end, the affine distortion error is proposed. It estimates the distance of pixel movement by the affine distortion . The viewpoint search algorithm selects the proper viewpoints in order to enhance the endurance of ASIFT using the affine distortion error. In the experimental result, the proposed method finds the correct matches 1.18 times more than the conventional method with same computational load.

Acknowledgment

This research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2016-H8601-16-1008) supervised by the IITP(Institute for Information & communications Technology Promotion)

References

[1] D. Lowe, “Distinctive image features from scale- invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Jan. 2004.

[2] J.-M. Morel and G. Yu, "ASIFT: A new framework for fully affine invariant image comparison," SIAM Journal on Imaging Sciences (SIIMS), vol. 2, no. 2, pp.438-469, 2009.

[3] K. Mikolajczyk, T. Tuytelaars, and C. Schmid, “A Comparison of Affine Region Detectors,” Int. J. Comput. Vis., vol. 65, no. 1-2, pp. 43-72, Nov. 2005.

TABLE II

THE SELECTED VIEWPOINTS BY THE PROPOSED VIEWPOINT SEARCH ALGORITHM FOR ASIFT

Viewpoint index	Latitude (Degree)	Longitude (Degree)	Viewpoint index	Latitude (Degree)	Longitude (Degree)	Viewpoint index	Latitude (Degree)	Longitude (Degree)
0	0	0	14	70	87	28	78	94
1	48	45	15	70	153	29	79	82
2	48	123	16	71	81	30	79	144
3	52	159	17	71	122	31	79	163
4	53	77	18	72	63	32	81	44
5	54	8	19	73	8	33	81	99
6	54	94	20	73	92	34	82	87
7	62	137	21	74	134	35	82	178
8	63	67	22	75	35	36	83	82
9	64	27	23	75	52	37	83	94
10	64	102	24	75	74	38	83	153
11	65	47	25	75	101	39	84	28
12	65	114	26	75	109	40	84	88
13	69	173	27	78	20	41	84	90