Lucas-Kanade Image Registration Using Camera Parameters

Sunghyun Cho[a], Hojin Cho[a], Yu-Wing Tai[b],
Young Su Moon[c], Junguk Cho[c], Shihwa Lee[c], and Seungyong Lee[a]

[a]POSTECH, Pohang, Korea
[b]KAIST, Daejeon, Korea
[c]Samsung Electronics, Suwon, Korea

ABSTRACT

The Lucas-Kanade algorithm and its variants have been successfully used for numerous works in computer vision, which include image registration as a component in the process. In this paper, we propose a Lucas-Kanade based image registration method using camera parameters. We decompose a homography into camera intrinsic and extrinsic parameters, and assume that the intrinsic parameters are given, e.g., from the EXIF information of a photograph. We then estimate only the extrinsic parameters for image registration, considering two types of camera motions, 3D rotations and full 3D motions with translations and rotations. As the known information about the camera is fully utilized, the proposed method can perform image registration more reliably. In addition, as the number of extrinsic parameters is smaller than the number of homography elements, our method runs faster than the Lucas-Kanade based registration method that estimates a homography itself.

Keywords: Image alignment, registration, Lucas-Kanade, homography, intrinsic parameters

1. INTRODUCTION

Image registration is the problem of aligning different images of the same scene, i.e., finding pixel-wise correspondence between images. This problem has been important in many fields, including computer vision and robotics. Many computer vision and robotics algorithms use multiple images, e.g., video frames or images captured by multiple cameras, and they often need pixel-wise correspondence between the images.

The Lucas-Kanade algorithm[1] would be the most well-known method for image registration. The algorithm aligns an input image to a template image so that the transformed input image has the same pixel values as the template image. For aligning images, the algorithm iteratively estimates an additive increment for the alignment parameters to update the current parameter values. Due to the simplicity and the effectiveness of the algorithm, it is still one of the most widely used image registration methods, even though it was proposed in 1981.

Besides the original Lucas-Kanade algorithm, a number of variants have been introduced. Szeliski and Shum[2] proposed a compositional approach, which iteratively estimates an incremental warp, instead of an additive increment of parameters. In their approach, an incremental warp is computed around the identity warp to reduce the computational overhead with pre-computation of the Hessian matrix. Baker and Matthews[3] proposed the inverse compositional method, which warps the template image instead of the input image. This enables pre-computation of a significant amount of values, resulting in a faster registration process. For a comprehensive literature review for the Lucas-Kanade algorithm and its variants, the reader is referred to Baker and Matthews.[3]

To align images, the Lucas-Kanade algorithm assumes a motion model, which describes the motion of the target scene or the motion of a camera between images. The most commonly used motion model in image registration is a 2D planar transformation, which includes translation, Euclidean, similarity, affine, and projective transforms. These 2D planar transforms can effectively describe the motions between images using a small number of parameters. For example, a projective transform or a homography is determined by only eight parameters. Due to the simplicity of this motion model, it has been widely used for many applications of image registration.

Further author information (emails): Sunghyun Cho, sodomau@postech.ac.kr; Hojin Cho, nucl23@postech.ac.kr; Yu-Wing Tai, yuwing@cs.kaist.ac.kr; Young Su Moon, mys66@samsung.com; Junguk Cho, junguk.cho@samsung.com; Shihwa Lee, hwa.lee@samsung.com; Seungyong Lee, leesy@postech.ac.kr

For aligning or tracking deformable objects, more complicated motion models have also been used. Cootes *et al.*[4] introduced Active Appearance Model, and Sclaroff and Isidoro[5] introduced Active Blobs. However, due to the high complexity, these motion models are usually used in limited cases, such as face tracking.

Although 2D planar transforms have been widely used, they do not explicitly represent the camera properties and motions, and the accuracy of image registration can be reduced. For example, projective transforms can express arbitrary planar motions, some of which are implausible under specific camera configurations. A projective transform represented by a homography can be decomposed into two sets of parameters: intrinsic and extrinsic parameters of a camera. Intrinsic parameters describe the properties of a camera, such as focal length, aspect ratio, and sensor resolution. Extrinsic parameters represent the relative position and direction of a camera to the target scene. By separating the intrinsic and extrinsic parameters, we can explicitly incorporate the camera properties and motion into the image registration process to improve the performance.

In this paper, we present a Lucas-Kanade based image registration algorithm, which separately handles the intrinsic and extrinsic parameters of a camera. We assume that the target scene is planar and the intrinsic parameters are given, e.g., from the EXIF information of a photograph. We estimate only the extrinsic parameters that represent camera motions needed for image registration. For camera motions, we consider full six-parameter motions, containing translations and rotations, and three-parameter rotations. With this approach, we fully utilize the known information of a camera, and image registration can be performed more reliably. The number of extrinsic parameters is smaller than the number of elements in a homography, and our method runs faster than estimating a homography directly. We also experimentally analyze how much gain we can obtain in terms of accuracy and speed of image registration, compared to homography estimation. In the experiments, we use the inverse compositional method, as it is one of the fastest variant of the Lucas-Kanade algorithm.[3]

The idea of using camera parameters for image registration is not completely novel. Szeliski and Shum[2] presented an image registration method that estimates three-parameter rotational motions of a camera using known intrinsic parameters. However, they used simplified parameters for camera intrinsics and did not provide an analysis of performance gains obtained using camera parameters. In this paper, we use a more complete set of camera intrinsic parameters and show detailed derivations for image registration using six- and three-parameter camera motions. We also apply the detailed derivations to the inverse compositional method and analyze the accuracy and speed using experimental results.

The remaining of this paper is organized as follows. In Sec. 2, we review the Lucas-Kanade algorithm and the inverse compositional method, which is a fast variant of the Lucas-Kanade algorithm. In Sec. 3, we derive the Lucas-Kanade algorithm and the inverse compositional method using camera parameters. In Sec. 4, we analyze the performance gain of the proposed approach using experimental results. In Sec. 5, we conclude the paper and discuss future work.

2. REVIEW OF THE LUCAS-KANADE ALGORITHM

2.1 Lucas-Kanade Algorithm

The objective of the Lucas-Kanade algorithm is to align a template image $T(\mathbf{x})$ to an input image $I(\mathbf{x})$, where $\mathbf{x} = (x, y)^T$ denotes a 2D vector representing a pixel position. Let $\mathbf{M}(\mathbf{x}; \mathbf{p})$ be a spatially-varying motion field that gives a new position for each pixel position $\mathbf{x}$, where $\mathbf{p}$ is a vector of motion parameters. If we assume a translational motion, then we may define a parameter vector $\mathbf{p}$ as $\mathbf{p} = (p_1, p_2)^T$, and a motion field $\mathbf{M}(\mathbf{x}; \mathbf{p})$ can be represented by $\mathbf{M}(\mathbf{x}; \mathbf{p}) = (x + p_1, y + p_2)^T$. In general, $\mathbf{M}(\mathbf{x}; \mathbf{p})$ can be an arbitrary transform, such as affine and projective transforms.

The Lucas-Kanade algorithm aligns a template image $T(\mathbf{x})$ to an input image $I(\mathbf{x})$ by minimizing

$$\sum_{\mathbf{x}} \left[I(\mathbf{M}(\mathbf{x}; \mathbf{p})) - T(\mathbf{x}) \right]^2 \tag{1}$$

with respect to $\mathbf{p}$. Unfortunately, minimizing Eq. (1) is not easy, because I is not a smooth function with respect to $\mathbf{M}(\mathbf{x}; \mathbf{p})$. To resolve this problem, the Lucas-Kanade algorithm assumes that the current estimate $\mathbf{p}$ of the

parameters is known, and iteratively estimates an incremental update $\Delta\mathbf{p}$, which minimizes

$$\sum_{\mathbf{x}} \left[I(\mathbf{M}(\mathbf{x};\mathbf{p}+\Delta\mathbf{p})) - T(\mathbf{x})\right]^2, \tag{2}$$

and updates $\mathbf{p}$ by $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$.

Since Eq. (2) is still difficult to optimize, the Lucas-Kanade algorithm instead optimizes a linearly approximated function

$$\begin{aligned} E(\mathbf{p}+\Delta\mathbf{p}) &= \sum_{\mathbf{x}} \left[I(\mathbf{M}(\mathbf{x};\mathbf{p}+\Delta\mathbf{p})) - T(\mathbf{x})\right]^2 & (3)\\ &\approx \sum_{\mathbf{x}} \left[I(\mathbf{M}(\mathbf{x};\mathbf{p})) + \nabla I \frac{\partial \mathbf{M}}{\partial \mathbf{p}} \Delta\mathbf{p} - T(\mathbf{x})\right]^2, & (4) \end{aligned}$$

where $\nabla I = (\partial I/\partial x, \partial I/\partial y)$ is the gradient of image I evaluated at the transformed position $\mathbf{M}(\mathbf{x};\mathbf{p})$. For a translational motion, $\partial\mathbf{M}/\partial\mathbf{p}$ is defined by

$$\frac{\partial \mathbf{M}}{\partial \mathbf{p}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{5}$$

Eq. (4) can be effectively solved by the least square method.

For a homography, we can derive the Lucas-Kanade algorithm in the same way. We first define a parameter vector $\mathbf{p}$ and its corresponding motion field $\mathbf{M}(\mathbf{x};\mathbf{p})$, and then derive $\partial\mathbf{M}/\partial\mathbf{p}$. Finally, the Lucas-Kanade algorithm for a homography is derived by substituting the motion field and its derivative in Eq. (4). First, we define a parameter vector $\mathbf{p}$ as

$$\mathbf{p} = (p_{00}, p_{01}, p_{02}, p_{10}, p_{11}, p_{12}, p_{20}, p_{21})^T. \tag{6}$$

Using the parameters, a homography $\mathbf{P}$ can be defined by

$$\mathbf{P} = \begin{bmatrix} 1+p_{00} & p_{01} & p_{02} \\ p_{10} & 1+p_{11} & p_{12} \\ p_{20} & p_{21} & 1 \end{bmatrix}. \tag{7}$$

The motion field $\mathbf{M}(\mathbf{x};\mathbf{p})$ can be defined by

$$\mathbf{M}(\mathbf{x};\mathbf{p}) = \frac{\mathbf{D}\mathbf{M}_H(\mathbf{x};\mathbf{p})}{\mathbf{n}^T\mathbf{M}_H(\mathbf{x};\mathbf{p})}, \tag{8}$$

where $\mathbf{D}$ is a projection matrix that maps a 3D vector $\mathbf{y}$ in homogenous coordinates to a 2D vector $\mathbf{x}$ in inhomogeneous (Euclidean) coordinates:

$$\mathbf{D} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{x} = \mathbf{D}\mathbf{y}. \tag{9}$$

$\mathbf{n} = (0,0,1)^T$ is a vector for extracting the z-component of homogeneous coordinates. $\mathbf{M}_H(\mathbf{x};\mathbf{p})$ is a function that maps a 2D inhomogeneous vector to a transformed 3D homogeneous vector:

$$\mathbf{M}_H(\mathbf{x};\mathbf{p}) = \mathbf{P}\left(\mathbf{D}^T\mathbf{x} + \mathbf{n}\right). \tag{10}$$

Finally, through some algebraic manipulation, we get

$$\frac{\partial \mathbf{M}}{\partial \mathbf{p}} = \frac{1}{D}\begin{bmatrix} x & y & 1 & 0 & 0 & 0 & -x'x & -x'y \\ 0 & 0 & 0 & x & y & 1 & -y'x & -y'y \end{bmatrix}, \tag{11}$$

where

$$x' = \frac{(1+p_{00})x + p_{01}y + p_{02}}{D}, \tag{12}$$

$$y' = \frac{p_{10}x + (1+p_{11})y + p_{12}}{D} \tag{13}$$

and

$$D = p_{20}x + p_{21}y + 1. \tag{14}$$

2.2 Inverse Compositional Method

The inverse compositional method proposed by Baker and Matthews[3] iteratively minimizes:

$$\sum_{\mathbf{x}} \left[T\left(\mathbf{M}(\mathbf{x};\Delta\mathbf{p})\right) - I(\mathbf{M}\left(\mathbf{x};\mathbf{p}\right))\right]^2 \tag{15}$$

with respect to $\Delta\mathbf{p}$ and then updates the motion field $\mathbf{M}(\mathbf{x};\mathbf{p})$ using

$$\mathbf{M}(\mathbf{x};\mathbf{p}) \leftarrow \mathbf{M}(\mathbf{x};\mathbf{p}) \circ \mathbf{M}(\mathbf{x};\Delta\mathbf{p})^{-1}, \tag{16}$$

where $\circ$ is a composition operator such that $\mathbf{M}(\mathbf{x};\mathbf{p}) \circ \mathbf{M}(\mathbf{x};\Delta\mathbf{p})^{-1} \equiv \mathbf{M}(\mathbf{M}(\mathbf{x};\Delta\mathbf{p})^{-1};\mathbf{p})$. A significant difference between the inverse compositional method and the original Lucas-Kanade algorithm is that the inverse compositional method warps T instead of I in Eq. (15). This enables pre-computation of a significant amount of values, and consequently, each iteration needs much less amount of computation.

As Eq. (15) is not easy to optimize, the inverse compositional method minimizes its linearized approximation:

$$\sum_{\mathbf{x}} \left[T\left(\mathbf{M}(\mathbf{x};\mathbf{0})\right) + \nabla T \frac{\partial \mathbf{M}}{\partial \mathbf{p}} \Delta\mathbf{p} - I(\mathbf{M}\left(\mathbf{x};\mathbf{p}\right))\right]^2, \tag{17}$$

where $\partial\mathbf{M}/\partial\mathbf{p}$ is computed at $(\mathbf{x};\mathbf{0})$. If we assume a projective motion or a homography, $\partial\mathbf{M}/\partial\mathbf{p}$ is defined by

$$\frac{\partial \mathbf{M}}{\partial \mathbf{p}} = \begin{bmatrix} x & y & 1 & 0 & 0 & 0 & -x^2 & -xy \\ 0 & 0 & 0 & x & y & 1 & -xy & -y^2 \end{bmatrix}. \tag{18}$$

It is worth noting that Eq. (18) is much simpler than Eq. (11), since Eq. (18) is computed at $\mathbf{p} = \mathbf{0}$.

3. LUCAS-KANADE ALGORITHM USING CAMERA PARAMETERS

If the information about the camera is available, we can decompose the elements of a homography $\mathbf{P}$ by rewriting $\mathbf{P}$ as follows:

$$\mathbf{P} = \mathbf{K}(\mathbf{R} + \mathbf{T})\mathbf{K}^{-1}, \tag{19}$$

where $\mathbf{K}$ is the camera intrinsic matrix, $\mathbf{R}$ is the rotation matrix, and $\mathbf{T}$ is the translation matrix.[6] Assuming there is no shearing effect in the camera sensor, we can obtain the camera intrinsic matrix $\mathbf{K}$ from the information in an EXIF tag of a photograph. As a result, six parameters (three for $\mathbf{R}$ and three for $\mathbf{T}$) are sufficient to describe each homography $\mathbf{P}$. In the following, we will describe how to estimate a homography $\mathbf{P}$ with these six parameters.

To incorporate the camera information into the Lucas-Kanade algorithm or the inverse compositional method, we first need to define a motion parameter vector $\mathbf{p}$ and its corresponding motion field $\mathbf{M}(\mathbf{x};\mathbf{p})$, and then derive $\partial\mathbf{M}/\partial\mathbf{p}$ as done for a general homography in Sec. 2. First, we define a parameter vector $\mathbf{p}$ as

$$\mathbf{p} = \left(\theta_x, \theta_y, \theta_z, t_x, t_y, t_z\right)^T, \tag{20}$$

where θ_x, θ_y, and θ_z are rotational angles along the x, y, and z axes, respectively, and t_x, t_y and t_z are translation parameters. We can then define a motion field as in Eq. (8), and $\mathbf{M}_H(\mathbf{x};\mathbf{p})$ becomes

$$\mathbf{M}_H(\mathbf{x};\mathbf{p}) \quad = \quad \mathbf{K}(\mathbf{R}+\mathbf{T})\mathbf{K}^{-1}\left(\mathbf{D}^T\mathbf{x}+\mathbf{n}\right). \tag{21}$$

Note that matrices $\mathbf{R}$ and $\mathbf{T}$ are functions of the parameter vector $\mathbf{p}$.

By differentiating $\mathbf{M}(\mathbf{x};\mathbf{p})$ in Eq. (8) using $\mathbf{M}_H$ in Eq. (21) with respect to each element $q \in \mathbf{p}$, we can derive $\partial\mathbf{M}/\partial\mathbf{p}$ as

$$\frac{\partial\mathbf{M}}{\partial\mathbf{p}} \quad = \quad \left[\begin{array}{cccccc} \frac{\partial\mathbf{M}}{\partial\theta_x} & \frac{\partial\mathbf{M}}{\partial\theta_y} & \frac{\partial\mathbf{M}}{\partial\theta_z} & \frac{\partial\mathbf{M}}{\partial t_x} & \frac{\partial\mathbf{M}}{\partial t_y} & \frac{\partial\mathbf{M}}{\partial t_z} \end{array}\right], \tag{22}$$

where

$$\frac{\partial\mathbf{M}}{\partial q} \quad = \quad \frac{\left(\mathbf{D}\frac{\partial\mathbf{M}_H}{\partial q}\right)\left(\mathbf{n}^T\mathbf{M}_H\right) - \left(\mathbf{D}\mathbf{M}_H\right)\left(\mathbf{n}^T\frac{\partial\mathbf{M}_H}{\partial q}\right)}{\left(\mathbf{n}^T\mathbf{M}_H\right)^2} \tag{23}$$

and

$$\frac{\partial\mathbf{M}_H}{\partial q} \quad = \quad \mathbf{K}\frac{\partial(\mathbf{R}+\mathbf{T})}{\partial q}\mathbf{K}^{-1}\left(\mathbf{D}^T\mathbf{x}+\mathbf{n}\right). \tag{24}$$

It is straightforward to derive $\partial(\mathbf{R}+\mathbf{T})/\partial q$, and we omit the details.

The inverse compositional method using camera parameters can be derived similarly. In this case, $\partial\mathbf{M}_H/\partial q$ is significantly simplified because $\partial\mathbf{R}/\partial q$ is computed at $\mathbf{p}=\mathbf{0}$. Assuming that the camera coordinate system does not have any skewness, the camera intrinsic matrix $\mathbf{K}$ can be represented as:

$$\mathbf{K} \quad = \quad \left[\begin{array}{ccc} \alpha & 0 & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{array}\right], \tag{25}$$

where α and β are quantities determined by sensor resolutions, and (u_0, v_0) is a vector for matching the origin of the camera coordinate system and the center of the CCD sensor matrix. By putting Eq. (25) into Eq. (24), we can derive

$$\frac{\partial\mathbf{M}}{\partial\mathbf{p}} \quad = \quad \left[\begin{array}{cccccc} -y'x'/\beta & x'^2/\alpha+\alpha & -\alpha y'/\beta & \alpha & 0 & -x' \\ -y'^2/\beta-\beta & x'y'/\alpha & \beta x'/\alpha & 0 & \beta & -y' \end{array}\right] \tag{26}$$

for $\mathbf{p}=\mathbf{0}$, where $x'=x-u_0$ and $y'=y-v_0$.

We can further restrict the Lucas-Kanade algorithm and reduce the number of parameters by making additional assumptions on the camera configuration. For example, we may assume that the relative position of the camera to the (planar) scene is fixed, making $t_x=t_y=t_z=0$. Then, the parameter vector $\mathbf{p}$ becomes a 3D vector $\mathbf{p}=(\theta_x,\theta_y,\theta_z)^T$ with only rotation angles, and $\partial\mathbf{M}/\partial\mathbf{p}$ is simplified to

$$\frac{\partial\mathbf{M}}{\partial\mathbf{p}} \quad = \quad \left[\begin{array}{ccc} \frac{\partial\mathbf{M}}{\partial\theta_x} & \frac{\partial\mathbf{M}}{\partial\theta_y} & \frac{\partial\mathbf{M}}{\partial\theta_z} \end{array}\right]. \tag{27}$$

Again, for the inverse compositional method, we can compute Eq. (27) at $\mathbf{p}=\mathbf{0}$ and derive $\partial\mathbf{M}/\partial\mathbf{p}$ as:

$$\frac{\partial\mathbf{M}}{\partial\mathbf{p}} \quad = \quad \left[\begin{array}{ccc} -y'x'/\beta & x'^2/\alpha+\alpha & -\alpha y'/\beta \\ -y'^2/\beta-\beta & x'y'/\alpha & \beta x'/\alpha \end{array}\right]. \tag{28}$$

Note that Eq. (28) becomes the same Jacobian matrix described by Szeliski and Shum[2] if $\alpha=\beta$ and $u_0=v_0=0$.

Figure 1. Some pairs of template and input images in our test set. In each pair, the left is template and the right is input.

Table 1. Peak signal-to-noise ratio (PSNR) values of the input images warped by estimated homographies.

Test case	1	2	3	4	5	6	7	8	9	10
LK-Homography	88.02	104.65	97.72	114.20	89.87	76.99	81.18	76.60	101.32	108.35
LK-Camera	88.03	105.09	98.06	114.94	90.25	77.31	81.29	77.38	101.06	108.33
LK-CameraRot	88.14	105.08	98.11	114.54	90.64	77.12	81.27	77.31	101.31	108.52

4. EXPERIMENTS

To analyze the performance gains of the proposed approach for image registration, we experimented with the inverse compositional method using the approach. For algorithm implementation, we used C++ language. The testing environment was a PC running MS Windows 7 64bit version with Intel Core i7 CPU and 12GB RAM. To perform image registration for large scale pixel displacements, we adopted a multi-scale approach.[7]

We generated a test set consisting of 10 pairs of template and input images. For each template image, we generated a random synthetic warp, then applied the warp to the template image in order to produce the corresponding input image. For generating each synthetic warp, we randomly picked three values for rotational angles along the three axes, and then composed a homography using the three random angles and pre-defined intrinsic parameters. To obtain a test set good for rotation only motions, as well as full 3D motions with six degrees of freedom, we did not introduce translational motions for the homographies. Fig. 1 shows some pairs in our test set. To each pair, we applied three different algorithms: the previous inverse compositional method that estimates a homography, our algorithm that estimates a homography with six degrees of freedom (three translation and three rotation parameters), and our algorithm that estimates a homography with three degrees of freedom (three rotation parameters). For clarity, we name the algorithms LK-Homography, LK-Camera, and LK-CameraRot, respectively.

We first analyzed how much LK-Camera and LK-CameraRot gained in terms of accuracy. To measure the accuracy of each algorithm, we warped the input images using the homographies estimated by each algorithm, and measured the differences between the template images and the corresponding warped input images with PSNR. Table 1 shows the PSNR values of the test cases. The PSNR values of LK-CameraRot are the highest among the three algorithms on average, and the PSNR values of LK-Homography are the lowest. The average PSNR values of LK-Homography, LK-Camera, and LK-CameraRot are 93.89, 94.17, and 94.20 dB, respectively. This shows the benefits of using additional information of camera intrinsic parameters in image registration.

As well as the accuracy improvement, the computation time for image registration can be reduced, as the proposed camera motion estimation methods have smaller degrees of freedom than direct estimation of a ho-

Table 2. Computation times (in seconds) for the three different image registration algorithms on our test set.

Test case	1	2	3	4	5	6	7	8	9	10
LK-Homography	12.35	12.27	12.22	12.15	12.13	12.18	12.30	12.09	12.77	12.72
LK-Camera	11.95	11.87	11.82	11.73	11.73	11.82	11.85	11.75	12.34	12.00
LK-CameraRot	11.25	11.17	11.13	11.15	11.23	11.19	7.60	11.28	11.70	11.73

mography. To compare computation times, we measured the running time of each algorithm for each test case. Each image in the test set is of 1.4 mega pixels. Table 2 shows the computation times of the three algorithms on the test cases. As a homography has eight degrees of freedom, the computation times of LK-Homography are the highest among the three algorithms on average. On the other hand, LK-CameraRot shows the lowest computation times. The average computation times of LK-Homography, LK-Camera, and LK-CameraRot are 12.32, 11.89, and 10.94 seconds, respectively.

5. CONCLUSION AND FUTURE WORK

In this paper, we proposed a Lucas-Kanade algorithm using camera parameters. We first reviewed the Lucas-Kanade algorithm and the inverse compositional method. We then modified the Lucas-Kanade algorithm and the inverse compositional method to utilize the known information of a camera while only estimating camera motions. Finally, we showed by experimental results that our approach can provide more accurate and faster image registration.

Although the performance gains were demonstrated in this paper, the experiments were limited to the inverse compositional method and the gains were not huge. Testing the idea of using camera parameters for image registration with other algorithms, such as Levenberg-Marquardt,[3] will be interesting future work, as variants of the Lucas-Kanade algorithm show different characteristics in terms of accuracy and speed.

ACKNOWLEDGEMENTS

This work was supported in part by the Brain Korea 21 Project, the Industrial Strategic Technology Development Program of MKE/MCST/KEIT (KI001820, Development of Computational Photography Technologies for Image and Video Contents), the Basic Science Research Program of MEST/NRF (2010-0019523), and a grant from Samsung Electronics.

REFERENCES

[1] Lucas, B. and Kanade, T., "An iterative image registration technique with an application to stereo vision," in [*Proceedings of the International Joint Conference on Artifical Intelligence*], 674–679 (1981).

[2] Szeliski, R. and Shum, H.-Y., "Creating full view panoramic image mosaics and texture-mapped models," *SIGGRAPH 95* , 251–258 (1997).

[3] Baker, S. and Matthews, I., "Lucas-kanade 20 years on: a unifying framework," *International Journal of Computer Vision* **56**(3), 221–255 (2004).

[4] Cootes, T., Edwards, G., and Taylor, C., "Active appearance models," in [*Proceedings of the European Conference on Computer Vision*], **2**, 484–498 (1998).

[5] Sclaroff, S. and Isidoro, J., "Active blobs," in [*Proceedings of the 6th IEEE International Conference and Computer Vision*], 1146–1153 (1998).

[6] Forsyth, D. and Ponce, J., [*Computer Vision: A Modern Approach*], Prentice Hall (2002).

[7] Szeliski, R., "Image alignment and stitching: A tutorial," Tech. Rep. MSR-TR-2004-92, Microsoft Research (2004).