Expansion of Hager Belhumeur Inverse Additive Algorithm to Homographies

Hyun Seok Hong and Myung Jin Chung
Department of Electrical Engineering and Computer Science
Korea Advanced Institute of Science and Technology
373-1 Guseong-dong, Yuseong-gu, Daejeon, Republic of Korea
Email: wiser@kaist.ac.kr, mjchung@ee.kaist.ac.kr

Abstract—Image alignment is a widely used technique in computer vision, and it can be applied to many areas such as image registration and region tracking. Hager and Belhumeur proposed an efficient image alignment algorithm which can find out translation and affine deformations quickly. However, it is known that their algorithm cannot be applied to homographies. Image alignment with a homography is important because general motions of object in 3D space are observed as perspective motions which are another name of homographies. In this paper, we propose an expanded inverse additive algorithm for homographies, and apply the algorithm to region tracking. We conduct real-time tracking experiments to show the performance of the proposed algorithm.

Keywords—Hager-Belhumeur algorithm, inverse additive algorithm, region tracking, image alignment, homography

I. INTRODUCTION

In image processing and computer vision areas, image alignment is one of the most widely used techniques. It can be used as image registration, region tracking, etc. This image alignment is achieved by finding motion parameters which minimize the sum of squared differences between template and input images. It is known that the most accurate alignment is based on gradient descent methods for the minimization. In 1981, Lucas and Kanade derived an image alignment algorithm based on optical flow [1]. It works accurately for all kinds of motion warps. In 1998, Hager and Belhumeur developed an efficient version of the Lucas-Kanade algorithm, and made it to cope with illumination variation and partial occlusions [2]. In 2000, Shum and Szeliski developed a panoramic image alignment technique based on a compositional warp [3]. In 2001, Baker and Matthews proposed an efficient image alignment algorithm [4] based on a compositional motion warp. And they have developed a unifying framework including previously developed image alignment algorithms [5]. The framework included previous four algorithms: forward additive (Lucas and Kanade’s algorithm), forward compositional (Shum and Szeliski’s algorithm), inverse additive (Hager and Belhumeur’s algorithm) and inverse compositional (Baker and Matthews’ algorithm), and dealt with affine motions and homographies (except for the inverse additive algorithm for homographies). It is well known that the Hager-Belhumeur inverse additive algorithm can be applied to very limited sets of warps such as translational, affine, and some simple non-linear motion warps. Furthermore, there have been few studies that tried to apply the inverse additive algorithm to homographies. Consequently, Baker and Matthews also assumed that it cannot be applied to homographies.

A homography warp is frequently observed phenomenon related with a projective geometry. Suppose a planar object is rotating and translating relatively to a camera in 3D space, then a homography relation exists between any two of the projected images. When the motions of the object are generated near the camera, there will be the large amount of the perspective deformation. Therefore, an image alignment technique for a homography warp is very important especially in manipulating objects close to the camera, such as humanoid robots.

In Section II, we review the Hager-Belhumeur inverse additive algorithm in brief, and propose the directly expanded inverse additive algorithm for homographies. And in Section III, we apply the derived inverse algorithm for homographies to region tracking with synthetic sequential images and real-time video.

II. HAGER-BELHUMEUR INVERSE ADDITIVE ALGORITHM FOR HOMOGRAPHIES

A. Hager-Belhumeur Algorithm

The goal of the Hager-Belhumeur algorithm is to find an incremental parameters vector minimizing the sum of squared differences between a warped image of an image I and a reference template image T:

$$\sum_x |I(w(x; p + \Delta p), t + \Delta t) - T(x)|^2,$$  \hspace{1cm} (1)

where $x = [x, y]^T$ is a reference image pixel position vector, $p = [p_1, \cdots, p_k]^T$ is a warp parameters vector, $t$ is time, $\Delta p$ is an incremental parameters vector of $p$ during incremental time $\Delta t$, and a warp function $w$ moves $x$ to a new position using $p$. The warped image intensity $I$ in (1) can be linearized w.r.t. $\Delta p$ and $\Delta t$ by performing the first order Taylor expansion on $I(w(x; p + \Delta p), t + \Delta t)$ at $p$ and $t$:

$$I(w(x; p + \Delta p), t + \Delta t) \approx I(w(x; p), t) + \frac{\partial I(w(x; p), t)}{\partial p} \Delta p + \frac{\partial I(w(x; p), t)}{\partial t} \Delta t.$$  \hspace{1cm} (2)

The image intensity derivative w.r.t. time $t$ can be approximated as

$$\frac{\partial I(w(x; p), t)}{\partial t} \approx I(w(x; p), t + \Delta t) - I(w(x; p), t) / \Delta t.$$  \hspace{1cm} (3)
Using (2) and (3), the objective function (1) can be rewritten as
\[ \sum_x \left[ I(\mathbf{w}(\mathbf{x}; \mathbf{p}), t + \Delta t) + \frac{\partial I(\mathbf{w}(\mathbf{x}; \mathbf{p}), t)}{\partial \mathbf{p}} \Delta \mathbf{p} - T(\mathbf{x}) \right]^2. \] (4)

In general, the Jacobian in (4) is calculated by multiplication of image gradients and warp derivatives:
\[ \frac{\partial I(\mathbf{w}(\mathbf{x}; \mathbf{p}), t)}{\partial \mathbf{p}} = \frac{\partial I(\mathbf{w}(\mathbf{x}; \mathbf{p}), t)}{\partial \mathbf{w}(\mathbf{x}; \mathbf{p})} \cdot \frac{\partial \mathbf{w}(\mathbf{x}; \mathbf{p})}{\partial \mathbf{p}}. \] (5)

The Gaussian-Newton minimization solution \( \Delta \mathbf{p} \) of (4) is
\[ \Delta \mathbf{p} = H(\mathbf{p})^{-1} \sum_x J(\mathbf{x}, \mathbf{p})^T [T(\mathbf{x}) - I], \] (6)
where \( I \) is an abbreviation of \( I(\mathbf{w}(\mathbf{x}; \mathbf{p}), t + \Delta t) \), and
\[ J(\mathbf{x}, \mathbf{p}) = \frac{\partial I(\mathbf{w}(\mathbf{x}; \mathbf{p}), t)}{\partial \mathbf{p}}, \] (7)
\[ H(\mathbf{p}) = \sum_x J(\mathbf{x}, \mathbf{p})^T J(\mathbf{x}, \mathbf{p}). \] (8)

The parameters update rule is
\[ \mathbf{p} \leftarrow \mathbf{p} + \Delta \mathbf{p}. \] (9)

Equations (6) and (9) is actually the Lucas-Kanade algorithm, and (7) and (8), which depend on \( \mathbf{p} \), require much time for the calculations of image gradients and warp derivatives at the currently estimated parameters vector \( \mathbf{p} \).

The idea of the efficient Hager-Belhumeur algorithm is based on assuming image constancy:
\[ I(\mathbf{w}(\mathbf{x}; \mathbf{p}), t) = T(\mathbf{x}). \] (10)

Using (10), the Jacobian in (5) can be represented in terms of the template image:
\[ \frac{\partial I(\mathbf{w}(\mathbf{x}; \mathbf{p}), t)}{\partial \mathbf{p}} = \frac{\partial T(\mathbf{x})}{\partial \mathbf{x}} \left( \frac{\partial \mathbf{w}(\mathbf{x}; \mathbf{p})}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathbf{w}(\mathbf{x}; \mathbf{p})}{\partial \mathbf{p}}. \] (11)

If the two derivatives of \( \mathbf{w} \) in (11) can be factored into the product of a \( 2 \times k \) matrix \( \Gamma \) which depends only on image coordinates \( \mathbf{x} \), and a \( k \times k \) matrix \( \Sigma \) which depends only on parameters vector \( \mathbf{p} \) as
\[ \left( \frac{\partial \mathbf{w}(\mathbf{x}; \mathbf{p})}{\partial \mathbf{x}} \right)^{-1} \frac{\partial \mathbf{w}(\mathbf{x}; \mathbf{p})}{\partial \mathbf{p}} = \Gamma(\mathbf{x}) \Sigma(\mathbf{p}), \] then (11) can be rewritten as
\[ \frac{\partial I(\mathbf{w}(\mathbf{x}; \mathbf{p}), t)}{\partial \mathbf{p}} = \frac{\partial T(\mathbf{x})}{\partial \mathbf{x}} \cdot \Gamma(\mathbf{x}) \Sigma(\mathbf{p}). \] (13)

Finally, applying (13) to (6)-(8), the efficient solution \( \Delta \mathbf{p} \) is acquired:
\[ \Delta \mathbf{p} = \Sigma(\mathbf{p})^{-1} H_s^{-1} \sum_x J^T_s [T(\mathbf{x}) - I], \] (14)
where
\[ J_s(\mathbf{x}) = \frac{\partial T(\mathbf{x})}{\partial \mathbf{x}} \cdot \Gamma(\mathbf{x}), \] (15)
\[ H_s = \sum_x J_s(\mathbf{x})^T J_s(\mathbf{x}). \] (16)

Equations (15) and (16) can be calculated off-line, and only \( \Sigma(\mathbf{p})^{-1} \) and some multiplications with image differences are required on-line. This makes the Hager-Belhumeur algorithm to be efficient. Furthermore, the function \( \Sigma(\mathbf{p})^{-1} \) is given in the form of a simple pre-defined inverse function in most cases, therefore it does not require the matrix inversion.

The applicability of the Hager-Belhumeur inverse additive algorithm depends on the decomposition described in (12). In other words, the Hager-Belhumeur algorithm can be applied, whenever \( \Gamma \) and \( \Sigma \) satisfying (12) exist for a given warp function.

B. Derivation of Hager-Belhumeur Algorithm for Homographies

A homography warp has eight parameters and described as a nonlinear warp
\[ \mathbf{w}(\mathbf{x}; \mathbf{p}) = \frac{1}{p_{3x} p_{4y} + 1} \begin{bmatrix} p_{1x} + p_{4y} + p_7 & p_{2x} + p_{5y} + p_8 & p_{3x} + p_{4y} + p_1 \end{bmatrix}, \] (17)
where \( \mathbf{p} = [p_{1x}, \ldots, p_8]^T \) is a projective motion parameters vector. As described in Section II.A, to derive the inverse additive algorithm for a given motion warp, the decomposition (12) should be held.

The two derivatives for the homography warp are written as
\[ \frac{\partial \mathbf{w}(\mathbf{x}; \mathbf{p})}{\partial \mathbf{x}} = \frac{1}{z'} \begin{bmatrix} p_{1x} & p_{4y} & p_7 \\ p_{2x} & p_{5y} & p_8 \end{bmatrix} - \frac{1}{(z')^2} \begin{bmatrix} x'p_3 & y'p_3 & y'p_6 \\ y'p_3 & y'p_6 & y'p_6 \end{bmatrix}, \] (18)
\[ \frac{\partial \mathbf{w}(\mathbf{x}; \mathbf{p})}{\partial \mathbf{p}} = \frac{1}{z'} \begin{bmatrix} x & 0 & -x' & 0 \\ 0 & x & -y' & 0 \end{bmatrix} \begin{bmatrix} 0 & -y' & 0 & 1 \\ y' & 0 & 0 & 1 \end{bmatrix}. \] (19)

where \( x' = p_{1x} + p_{4y} + p_7, y' = p_{2x} + p_{5y} + p_8 \), and \( z' = p_{3x} + p_{4y} + 1 \).

The two derivatives (18) and (19) are non-linear equations, and have second order denominators coupled with \( \mathbf{x} \) and \( \mathbf{p} \), and therefore, it seems to be impossible to decompose the multiplication of (18) and (19) into \( \Gamma \) and \( \Sigma \) of (12). In spite of the complexity, we found that each element of the multiplication of (18) and (19) contains quadratic equations of \( x \) and \( y \). Using this element representation, the decomposition (12) can be found.

In calculating the efficient solution in (14), \( \Sigma^{-1} \) and \( \Gamma \) are required, and those are listed in the followings:
\[ \Gamma(\mathbf{x}) = \begin{bmatrix} x & 0 & -x^2 & y & 0 & -xy & 1 & 0 \\ 0 & x & -xy & 0 & y & -y^2 & 0 & 1 \end{bmatrix}, \] (20)
\[ \Sigma(\mathbf{p})^{-1} = \begin{bmatrix} p_1 p_4 p_7 & 0 & 0 & -p_1 p_3 & -p_1 p_6 \\ p_2 p_4 p_7 & 0 & 0 & -p_2 p_3 & -p_2 p_6 \\ p_4 p_6 & 0 & 0 & -p_3 p_5 & -p_3 p_6 \\ 0 & 0 & 0 & p_1 p_4 p_7 & -p_3 p_4 & -p_4 p_6 \\ 0 & 0 & 0 & p_2 p_5 p_8 & -p_3 p_5 & -p_5 p_6 \\ 0 & 0 & 0 & -p_5 p_6 & 1 & -p_5 p_6 & -p_5 p_6 \\ 0 & 0 & 0 & 0 & p_1 & -p_3 p_7 & -p_4 p_7 & -p_6 p_7 \\ 0 & 0 & 0 & 0 & p_2 & -p_3 p_8 & -p_4 p_8 & -p_6 p_8 \end{bmatrix}. \] (21)

Because (21) is provided as an inverted matrix, calculating the inverse of \( \Sigma(\mathbf{p}) \) at each calculation is not necessary. Once
the incremental parameters vector $\Delta \mathbf{p}$ is calculated using (14), (20), and (21), the new warp parameters vector can be updated using (9).

Let $\mathbf{p}_{IC} = [\tilde{p}_1, \cdots, \tilde{p}_8]^T$, $\Delta \mathbf{p}_{IC} = [\Delta \tilde{p}_1, \cdots, \Delta \tilde{p}_8]^T$ and $\mathbf{w}_{IC}$ be a parameters vector, an incremental parameters vector and a homography warp function of inverse compositional image alignment algorithm [4] respectively. In the inverse compositional algorithm, to make a vector with all zeros to be an identity warp parameters vector, a warp function is slightly different from (17):

$$
\mathbf{w}_{IC}(\mathbf{x}; \mathbf{p}_{IC}) = \frac{1}{\tilde{p}_3 x + \tilde{p}_6 y + 1} \begin{bmatrix}
(1 + \tilde{p}_1)x + \tilde{p}_4 y + \tilde{p}_7 \\
\tilde{p}_2 x + (1 + \tilde{p}_5)y + \tilde{p}_8
\end{bmatrix}.
$$

There are some similarities between the expanded Hager-Belhumeur inverse additive algorithm and the inverse compositional algorithm for homographies:

1) $-\Delta \mathbf{p}$ corresponds to the parameters of $\mathbf{w}_{IC}(\mathbf{x}; \Delta \mathbf{p}_{IC})^{-1}$.

2) $\Sigma(\mathbf{p})^{-1}$ corresponds to $\mathbf{w}_{IC}(\mathbf{x}; \mathbf{p}_{IC})$.

3) Instead of calculating the gradient of the current image, both of the two algorithms calculate the gradient of the template image off-line.

4) Online calculations are the computations of the difference image between the current image and the template image, some multiplications with the pre-computed Jacobian and inverse Hessian.

III. EXPERIMENTAL RESULTS

The derived image alignment algorithm for homographies is applied to a planar region tracking. The first tracking experiment uses synthetic sequential images, and the second tracking experiment uses real-time video.

The first sequential images are generated by translating and rotating a book cover image in 3D space with a camera with varying focal length.

The outlines of the tracked regions are represented by white dotted lines and plotted in Fig. 1. The template regions are tracked exactly under perspective deformation of the region. Furthermore, at frame 50, although there is partially occluded region in the image, it succeeds in tracking. This means the proposed algorithm is robust to small amount of occlusions by itself.

For the second tracking experiment, a camera captures images while a person manipulates a box and moves the camera in 3D space. In real-time tracking, there can be frequent partial occlusions. To be robust to the possibly large amount of occlusion, we adopt IRLS (Iteratively Reweighted Least-squares) algorithm, which Dutter and Huber [6] used. The real-time tracking was performed during moving boxes with human hand while the camera is looking at them. The real-time tracking results are shown in Fig. 2. The real-time tracking experiment showed that our tracker using inverse additive algorithm extended to homographies is working precisely even in the partial occlusion and illumination changes.

In case of tracking a 100 × 100 template image on a 3.4GHz Pentium-IV, the off-line calculations of the proposed algorithm took 2.02 milliseconds in C. The off-line calculation is actually not important because that portion is not calculated in the mean time of tracking stage. The online calculation per iteration took 0.53 milliseconds (that means the calculation speed is more than 1886 iterations/second) in C. In the implementation, the image warping took most of the elapsed time. If a user allocates 15 iterations per frame to cope with large amount

Fig. 1. Tracking results on synthetic images using the proposed image alignment algorithm which expanded the Hager-Belhumeur algorithm to homographies: white dotted lines represent tracked template regions. Perspective deformed regions have intensity errors in warped images into the reference template coordinates. Large deformation means that the warped images possibly have more intensity noise, and as a results, it makes the algorithm to converge slowly.

Fig. 2. Real-time tracking results: moving and rotating a box while looking at it with a moving camera. White outer lines and the center of the region are marked. Tracking is successful in arbitrary rotation and translation in the visual field of a camera. Furthermore, it is successful under illumination changes and partial occlusion.
of incremental deformation, the C implementation of the proposed algorithm goes on tracking at high speed of 125.7 frames per second.

IV. CONCLUSION

Trackers using homographies can cope with any kinds of 3D motions of a planar object, whereas trackers using translational or affine motion model can cope only limited 3D motions such as in-plane rotation and translations in distance. In this paper, we derived the inverse additive algorithm equations for homographies based on the Hager-Belhumeur algorithm, and compared the similarities to the another efficient algorithm, inverse compositional algorithm, for homographies. The proposed algorithm enabled the efficient Hager-Belhumeur image alignment algorithm to work with homographies. Furthermore, we conducted tracking with synthetic images, and real-time tracking using the derived inverse additive algorithm for homographies. Our proposed algorithm can be used in coping with occlusion, illumination changes, etc. We have shown that the proposed algorithm is efficient and fast enough to track image region with the speed of 1886 iterations per second, which means 125.7 frames per second provided that 15 iterations are assigned for an alignment between a template and a given frame.

ACKNOWLEDGMENT

This work was supported in part by the Ministry of Information & Communications, Korea, under the Information Technology Research Center (ITRC) Support Program.

REFERENCES