Inverse Compositional Method

He Zhao

1 Introduction

The inverse compositional algorithm is an efficient algorithm for image alignment and image registration. Rather than updating the *additive* estimate of warp parameters $\Delta\boldsymbol{p}$ (as in the Lucas-Kanade algorithm [5]), the inverse compositional algorithm iteratively solves for an inversed incremental warp $\boldsymbol{W}(\boldsymbol{x};\Delta\boldsymbol{p})^{-1}$ (an approach referred as *inverse compositional* method). The inverse compositional approach supports groupwise geometric transformations, and it improves efficiency by performing most computationally expensive calculations (i.e. the Gauss-Newton approximation to the Hessian matrix) at the pre-computation phase.

2 The Inverse Compositional Algorithm

Image alignment consists of moving, and possibly deforming, a template to minimize the difference between the template and an image. Suppose we are trying to align template image $T(\boldsymbol{x})$ to an input image $I(\boldsymbol{x})$, where $\boldsymbol{x} = (x, y)^T$ is a column vector containing the pixel coordinates. Let $\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p})$ denote the warp and $\boldsymbol{p} = (p_1, ...p_n)^T$ as a vector of parameters. The goal of image alignment is to minimize the sum of squared error between template image $T(\boldsymbol{x})$ and the input image $I(\boldsymbol{x})$ warped back onto the coordinate frame of the template,

$$\sum_x [I(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p})) - T(\boldsymbol{x})]^2 \tag{1}$$

with respect to $\boldsymbol{p}$, where the sum is performed over the pixel $\boldsymbol{x}$ in the template image $T(\boldsymbol{x})$. Since in general, the pixel values $I(\boldsymbol{x})$ are non-linear in $\boldsymbol{x}$, minimizing the expression in Equation (1) is a non-linear task. The Lucas-Kanade algorithm solves this problem by updating an estimate of warp parameters $\boldsymbol{p}$ and iteratively solving for increments to $\Delta\boldsymbol{p}$, i.e. to minimize the following expression,

$$\sum_x [I(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}+\Delta\boldsymbol{p})) - T(\boldsymbol{x})]^2 \tag{2}$$

with respect to $\Delta\boldsymbol{p}$, and then updating parameters $\boldsymbol{p}$ until it converges (i.e. $\|\Delta\boldsymbol{p}\| < \epsilon$)

$$\boldsymbol{p} \leftarrow \boldsymbol{p} + \Delta\boldsymbol{p} \tag{3}$$

As the name implies, the inverse compositional algorithm solves the minimization problem of Equation (1) by updating current estimated warp $\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p})$ with an *inverted* incremental warp $\boldsymbol{W}(\boldsymbol{x};\Delta\boldsymbol{p})^{-1}$,

$$\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}) \leftarrow \boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}) \circ \boldsymbol{W}(\boldsymbol{x};\Delta\boldsymbol{p})^{-1} \tag{4}$$

while minimizing the following expression,

$$\sum_{x}[T(\boldsymbol{W}(\boldsymbol{x};\Delta\boldsymbol{p})) - I(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}))]^2 \tag{5}$$

with respect to $\Delta\boldsymbol{p}$. Here the expression $\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}) \circ \boldsymbol{W}(\boldsymbol{x};\Delta\boldsymbol{p})^{-1}$ is a simple bilinear combination of the parameters of $\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p})$ and $\boldsymbol{W}(\boldsymbol{x};\Delta\boldsymbol{p})^{-1}$, and it can be rewritten as the *composition* warp $\boldsymbol{W}(\boldsymbol{W}(\boldsymbol{x};\Delta\boldsymbol{p});\boldsymbol{p})$. The Lucas-Kanade algorithm is therefore referred as the *forwards additive* algorithm [3]. It is essentially equivalent to the inverse compositional algorithm and they are both equivalent to minimizing the expression in Equation (1) [2]. However, updating $\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p})$ instead $\boldsymbol{p}$ makes the inverse compositional algorithm eligible to any set of warps which from a group. Performing a first order Taylor expansion on Equation (5) gives,

$$\sum_{x}[T(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{0})) + \nabla T\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}\Delta\boldsymbol{p} - I(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}))]^2 \tag{6}$$

Assuming $\boldsymbol{W}(\boldsymbol{x};\boldsymbol{0})$ is the identity warp, i.e. $\boldsymbol{W}(\boldsymbol{x};\boldsymbol{0}) = \boldsymbol{x}$, the solution to this least-squares problem is,

$$\Delta\boldsymbol{p} = H^{-1}\sum_{x}[\nabla T\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}]^T[I(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p})) - T(\boldsymbol{x})] \tag{7}$$

where H is the Hessian matrix,

$$H = \sum_{x}[\nabla T\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}]^T[\nabla T\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}] \tag{8}$$

and the Jacobian $\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$ is evaluated at $(\boldsymbol{x};\boldsymbol{0})$. Since the Hessian matrix is independent on the warp parameters $\boldsymbol{p}$, it is constant across iterations. Rather than computing the Hessian matrix in each iteration as in the forwards algorithms (e.g. the Lucas-Kanade algorithm), we can now pre-compute the Hessian matrix before iterations, which greatly improves efficiency. The inverse compositional algorithm can be described as follows [4],

1. **Pre-computation.** Pre-compute the Hessian matrix H using Equation (8), where $\nabla T\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$ is the steepest descent image of template $T(\boldsymbol{x})$.

2. **Image warping.** Warp the input image $I(\boldsymbol{x})$ with $\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p})$ to compute $I(\boldsymbol{W}(\boldsymbol{x};\boldsymbol{0}))$.

3. **Local registration.** Compute the local warp parameters $\Delta\boldsymbol{p}$ using Equation (7).

4. **Warp updating.** Update the current warp $\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}) \leftarrow \boldsymbol{W}(\boldsymbol{x};\boldsymbol{p}) \circ \boldsymbol{W}(\boldsymbol{x};\Delta\boldsymbol{p})^{-1}$.

Step 1 is done only once, Step 2 to 4 are iterated until warp converges, i.e. $\|\Delta\boldsymbol{p}\| < \epsilon$.

Assuming the number of warp parameters is n and the number of pixels in T is N, the computational complexity of the inverse compositional algorithm is $O(nN+n^3)$ per iteration and $O(n^2N)$ for pre-computation (performed only once), which is a substantial saving from the $O(n^2N+n^3)$-per-iteration Lucas-Kanade algorithm [3].

3 Example

To illustrate how the inverse compositional algorithm works, Baker *et al.* [1] demonstrated an example of image alignment using this algorithm. Figure 1 is the input image to be warped and Figure 2 is the template image.

Figure 1: Input image $I(\boldsymbol{x})$ [1].

Figure 2: Template image $T(\boldsymbol{x})$ [1].

Now we can follow Step 1 of the inverse compositional algorithm to pre-compute the steepest descent image (Figure 3) and the Hessian matrix H (Figure 4).

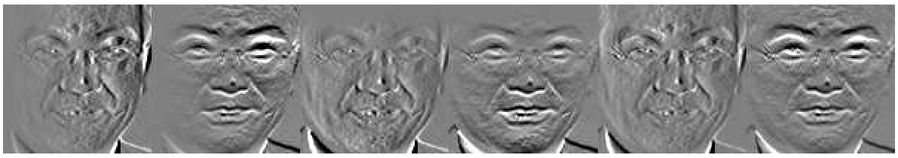

Figure 3: Steepest descent image $\nabla T \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$ [1].

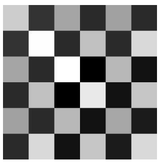

Figure 4: Hessian matrix H for $T(\boldsymbol{x})$ [1].

Then we enter the inner loop of Step 2 to 4, iteratively computing local warp parameters $\Delta\boldsymbol{p}$ (with the pre-computed H and $\nabla T \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$) and updating current warp $\boldsymbol{W}(\boldsymbol{x};\boldsymbol{p})$, until $\Delta\boldsymbol{p}$ is smaller than some threshold (Figure 5). Figure 6 shows the resulting warp of an extracted sub-region of the input image.

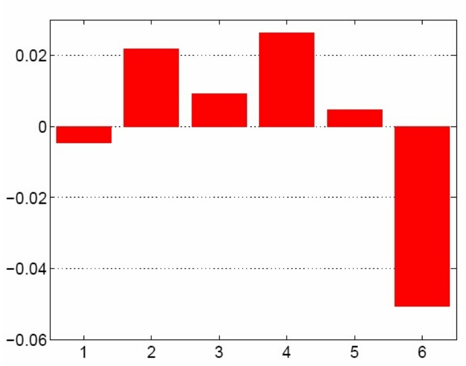

Figure 5: Parameter updates $\Delta \boldsymbol{p}$ [1].

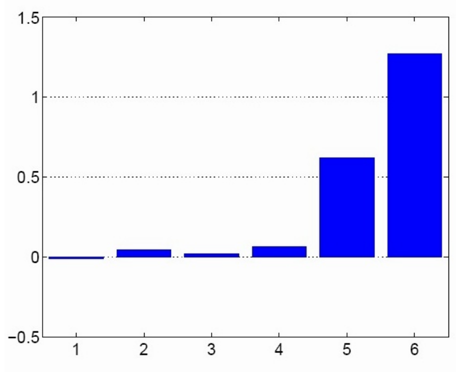

Figure 6: Resulting Warp $\boldsymbol{W}(\boldsymbol{x}; \boldsymbol{p})$ [1].

References

[1] S. Baker, R. Gross, I. Matthews, and T. Ishikawa. Lucas-kanade 20 years on: A unifying framework: Part 2. Technical Report CMU-RI-TR-03-01, Robotics Institute, Pittsburgh, PA, February 2003.

[2] S. Baker and I. Matthews. Equivalence and efficiency of image alignment algorithms. In *Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1090 – 1097, December 2001.

[3] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework: Part 1. Technical Report CMU-RI-TR-02-16, Robotics Institute, Pittsburgh, PA, July 2002.

[4] A. Bartoli. Direct image registration with gain and bias. In *Topics in Automatic 3D Modelling and Processing Workshop*, Verona, Italy, March 2006.

[5] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, pages 674–679, April 1981.