Comparing Object Alignment Algorithms with Appearance Variation: Forward-Additive vs Inverse-Composition

Patrick Lucey #1, Simon Lucey *2, Mark Cox #3, Sridha Sridharan #4 and Jeff Cohn *5

# *Speech, Audio, Image and Video Technology Laboratory*
Queensland University of Technology, Brisbane, QLD, 4000, Australia
[1] p.lucey@qut.edu.au
[3] md.cox@qut.edu.au
[4] s.sridharan@qut.edu.au

* *Robotics Institute*
Carnegie Mellon University, Pittsburgh, PA, 15213, USA
[2] slucey@cs.cmu.edu
[5] jeffcohn@cs.cmu.edu

***Abstract*—A common problem that affects object alignment algorithms is when they have to deal with objects with unseen intra-class appearance variation. Several variants based on gradient-decent algorithms, such as the Lucas-Kanade (or forward-additive) and inverse-compositional algorithms, have been proposed to deal with this issue by solving for both alignment and appearance simultaneously. In [1], Baker and Matthews showed that without appearance variation, the inverse-compositional (IC) algorithm was theoretically and empirically equivalent to the forward-additive (FA) algorithm, whilst achieving significant improvement in computational efficiency. With appearance variation, it would be intuitive that a similar benefit of the IC algorithm would be experienced over the FA counterpart. However, to date no such comparison has been performed. In this paper we remedy this situation by performing such a comparison. In this comparison we show that the two algorithms are not equivalent due to the inclusion of the appearance variation parameters. Through a number of experiments on the MultiPIE face database, we show that we can gain greater refinement using the FA algorithm due to it being a truer solution than the IC approach.**

I. Introduction

The accurate alignment of images is essential to almost all tasks in computer vision. Lucas and Kanade [2] devised an algorithm to image alignment using a gradient-descent approach, which worked on the basis of iteratively minimising the squared difference between a template and input image. Since then, the Lucas-Kanade algorithm has become one of the most widely used techniques in computer vision. However, an inherent problem with the Lucas-Kanade algorithm is the computational cost associated with the calculating the gradients of the input image at each iteration. In the seminal work of Baker and Matthews [1], they alleviated this overhead by swapping the role of template and input image, thus allowing for significant savings in computation by pre-computing the gradients from the template. In this overview paper, Baker and Matthews theoretically and empirically showed that their implementation, which they called the inverse-compositional (IC) algorithm, was equivalent to the Lucas-Kanade (or otherwise known as the forward-additive (FA)) algorithm. This result was important as the IC algorithm provided an efficient framework in which tasks like object alignment and tracking could be performed in real-time [3].

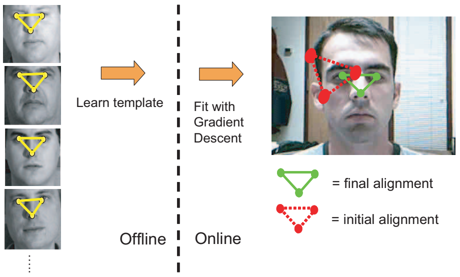


Fig. 1. Given that we have an ensemble of offline training face images, aligning the template to the unseen subject can be problematic as it is assumed that the template appears within the unseen image (which is obviously not true). Simultaneously solving for both the alignment and appearance by incorporating variation in the template has shown itself to somewhat overcome this problem. In this paper we show that by using the forward-additive (FA) algorithm we can achieve greater refinement compared to the current inverse-compositional (IC) algorithm which is in use.

A problem with gradient-descent approaches such as the FA and IC algorithms, are that they have poor generalisation properties when they have to deal with previously unseen intra-class object variation [4]. This is due to the assumption that the template appears within the input image which is problematic. For example for aligning faces (see Figure 1), if we learn a template face from an ensemble of training face images and try to align it to an unseen face, there is a very high probability that the template face and input face will be misaligned. This is because of the discrepancy between the appearance of the template face and the input face image.

In an attempt overcome this inherent mismatch, Black and Jephson [5] proposed an algorithm that incorporated the appearance variation in the template through an ensemble of known appearance variation images (which were learnt from a stack of images) and unknown appearance parameters. In this work, Black and Jephson utilised the FA framework. Baker et al. [6] employed a similar strategy, however they employed the IC algorithm. They called this the *simultaneous algorithm*, as it solved for both alignment and appearance at the same time using the IC framework.

As noted previously, it was shown in [1] that the IC algorithm was much more computationally efficient than the FA approach, whilst achieving the same alignment performance. As such, it would be intuitive that the simultaneous IC algorithm would enjoy the same performance and efficiency improvement over the simultaneous FA algorithm. However, no such comparison has been made. In an attempt to remedy this situation, in this paper we focus on providing a comparison between these two algorithms. From this comparison, we will show that these assumptions are not valid as the appearance variation parameters are included into different terms for both approaches, thus resulting in different solutions. The appearance parameters also diminish the advantage in efficiency as no pre-computation can take place[1].

In this paper, we show that the simultaneous FA algorithm provides a truer solution than the simultaneous IC algorithm as it does not include any second order terms. As a result, we show by using the simultaneous FA approach, we can get finer alignment compared to the simultaneous IC algorithm. Hence, our specific contributions stemming from this paper are:

- We implement a simultaneous FA approach to image alignment to alleviate the problem of linear appearance variation and compare it to the IC counterpart (Section 3). No such comparison has been done before.
- Mathematically, we show that the simultaneous FA and simultaneous IC algorithms are intrinsically different. From this we show that the simultaneous FA is a truer solution than the simultaneous IC approach (Section 3).
- In our experiments we show that the simultaneous FA algorithm achieves greater refinement than the simultaneous IC algorithm with an initial pixel error of 5 RMS for the affine warp (Section 4).

II. Background: Image Alignment Algorithms

A. Forward-Additive Algorithm

The standard Lucas-Kanade or forward-additive (FA) alignment algorithm attempts to find the parametric warp $\mathbf{p}$ that minimises the sum of the squared error between the template image T and the input imageY^2, such that

$$\arg\min_{\mathbf{p}} \|Y(\mathcal{W}(\mathbf{z};\mathbf{p})) - T(\mathcal{W}(\mathbf{z};\mathbf{0}))\|^2 \qquad (1)$$

where the vector $\mathbf{z} = \left[\mathbf{x}_1^T, \ldots, \mathbf{x}_N^T\right]^T$ is a concatenation of individual pixel $2D$ coordinates $\mathbf{x}$ within the template image T. One can map the image positions $\mathbf{z}$ to a set of new positions $\mathbf{z}'$,

$$\mathbf{z}' = \mathcal{W}(\mathbf{z};\mathbf{p}) = \left[\mathcal{W}(\mathbf{x}_1;\mathbf{p})^T, \ldots, \mathcal{W}(\mathbf{x}_N;\mathbf{p})^T\right]^T \qquad (2)$$

based on the warp parameters $\mathbf{p}$. The warp function is canonically translation $\mathcal{W}(\mathbf{x};\mathbf{p}) = \mathbf{x} + \mathbf{p}$, but $\mathcal{W}(\mathbf{x};\mathbf{p})$ has been extended to numerous other types of warps such as affine [5,3] and piece-wise affine [8]. In this paper we focus on the affine warp, which consists of $n = 6$ parameters such that $\mathbf{p} = [p_0, \ldots, p_5]^T$. Minimising Equation (1) is a non-linear optimisation task, so to optimise it we assume that a current estimate of $\mathbf{p}$ is known and then iteratively solve for increments to the parameters $\Delta\mathbf{p}$, where $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$. As such, we minimise the following expression

$$\arg\min_{\mathbf{\Delta p}} \|Y(\mathcal{W}(\mathbf{z};\mathbf{p}+\mathbf{\Delta p})) - T(\mathbf{z})\|^2 \qquad (3)$$

with respect to $\Delta\mathbf{p}$. It is worth noting that from here on we shall refer to $T(\mathcal{W}(\mathbf{z};\mathbf{0}))$ as $T(\mathbf{z})$ because $\mathcal{W}(\mathbf{z};0)$ is the identity warp as $\mathcal{W}(\mathbf{z};\mathbf{0}) = \mathbf{z}$. Before we can minimise Equation (3), we need to linearise $Y(\mathcal{W}(\mathbf{z};\mathbf{p}+\mathbf{\Delta p}))$. We can do this via the Taylor series expansion which yields

$$\left\| Y(\mathcal{W}(\mathbf{z};\mathbf{p})) + \nabla Y^T \frac{\partial \mathcal{W}(\mathbf{z};\mathbf{p})}{\partial \mathbf{p}}^T \Delta\mathbf{p} - T(\mathbf{z}) \right\|^2 \qquad (4)$$

From this we can denote the steepest-decent images as

$$\mathbf{SD}_{\mathrm{FA}}(\mathbf{z}) = \frac{\partial \mathcal{W}(\mathbf{z};\mathbf{p})}{\partial \mathbf{p}} \nabla Y \qquad (5)$$

and the error image as

$$\mathbf{E}_{\mathrm{FA}}(\mathbf{z}) = T(\mathbf{z}) - Y(\mathcal{W}(\mathbf{z};\mathbf{p})) \qquad (6)$$

Therefore, substituting in the steepest-decent and error images back into Equation (4), the minimisation with respect to $\Delta\mathbf{p}$ gives

$$2\mathbf{SD}_{\mathrm{FA}}(\mathbf{z}) \left[\mathbf{SD}_{\mathrm{FA}}(\mathbf{z})^T \Delta\mathbf{p} - \mathbf{E}_{\mathrm{FA}}(\mathbf{z})\right] \qquad (7)$$

Setting Equation (7) to equal zero and solving for $\Delta\mathbf{p}$ yields

$$\Delta\mathbf{p} = H^{-1}\mathbf{SD}_{\mathrm{FA}}(\mathbf{z})\mathbf{E}_{\mathrm{FA}}(\mathbf{z}) \qquad (8)$$

where the Hessian matrix

[1] It is worth noting that the simultaneous IC framework does allow for the "project-out" algorithm [6] which is a very computationally efficient algorithm. However, this algorithm suffers from poor alignment when the appearance variation is large due to simplifying assumptions. As we are interested in the performance, the "project-out" algorithm was not considered in this paper.

[2] For convenience we employ the notation that $\|\mathbf{a}-\mathbf{b}\|^2 = (\mathbf{a}-\mathbf{b})^T(\mathbf{a}-\mathbf{b})$, where $\mathbf{a}$ and $\mathbf{b}$ are column vectors

$$H = \mathbf{SD}_{\mathrm{FA}}(\mathbf{z})\mathbf{SD}_{\mathrm{FA}}(\mathbf{z})^T \tag{9}$$

We iteratively solve for $\Delta\mathbf{p}$ and then update $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$ until convergence or a maximum number of iterations is met. This non-linear optimisation process is called Gauss-Newton optimisation [7], but other least-square variants such as Newton and Levenberg-Marquardt optimisations can be used [7].

B. Inverse-Composition Algorithm

As pointed out by Baker et al. [7], there is a huge computational cost associated with the FA algorithm as the Hessian matrix in Equation (9) needs to be calculated every iteration. To overcome this problem, Baker et al. [7] found a way of switching the roles of the template and the input image, by compositionally inverting the warp parameters. By doing this, it allows the Hessian to be calculated from the constant template image, which which allows it to be precomputed, resulting in considerable computational savings. This is called the inverse-compositional (IC) algorithm and as mentioned, it is implemented by essentially inverse-composing the warp update, i.e. $\mathcal{W}(\mathbf{z};\mathbf{p}) \leftarrow \mathcal{W}(\mathcal{W}(\mathbf{z};\Delta\mathbf{p})^{-1};\mathbf{p})$. Thus, instead of minimising Equation (4), we attempt to minimise

$$\left\| [T(\mathbf{z}) - \nabla T^T \frac{\partial \mathcal{W}(\mathbf{z};\mathbf{0})}{\partial \mathbf{p}}^T \Delta\mathbf{p} + Y(\mathcal{W}(\mathbf{z};\mathbf{p}))\right\|^2 \tag{10}$$

By employing a similar process to the FA algorithm, we can obtain the $\Delta\mathbf{p}$ which minimises the expression. Updating our current estimate of the warp parameters $\mathbf{p}$ is done by evaluating $\mathcal{W}(\mathbf{z};\mathbf{p}) \leftarrow \mathcal{W}(\mathcal{W}(\mathbf{z};\Delta\mathbf{p})^{-1};\mathbf{p})$. So for the IC algorithm,

$$\Delta\mathbf{p} = -H^{-1}\mathbf{SD}_{\mathrm{IC}}(\mathbf{z})\mathbf{E}_{\mathrm{IC}}(\mathbf{z}) \tag{11}$$

where the Hessian matrix is

$$H = \mathbf{SD}_{\mathrm{IC}}(\mathbf{z})\mathbf{SD}_{\mathrm{IC}}(\mathbf{z})^T \tag{12}$$

and the steepest-decent images[3] are

$$\mathbf{SD}_{\mathrm{IC}}(\mathbf{z}) = \frac{\partial \mathcal{W}(\mathbf{z};\mathbf{0})}{\partial \mathbf{p}} \nabla T \tag{13}$$

and the error image is

$$\mathbf{E}_{\mathrm{IC}}(\mathbf{z}) = T(\mathbf{z}) - Y(\mathcal{W}(\mathbf{z};\mathbf{p})) \tag{14}$$

In [7], the FA and IC algorithms are shown to obtain the same image alignment performance. We also show this to be the case in the experiments presented in Section 5. However, this is not the case when appearance variation is included. We investigate this in the next section.

III. Simultaneous Algorithms

A. Simultaneous IC Algorithm

The image alignment algorithms described in the previous section both assume that the template appears in the input image. However, as noted back in Fig. 1, this assumption is likely to cause misalignment due to the appearance variation between the template and the input image. To counteract this, Baker et al. [6] included this linear appearance variation within the framework of the IC algorithm. They achieved this by incorporating the appearance variation within the template by letting $T(\mathbf{z}) = T(\mathbf{z}) + \sum_{i=1}^m \lambda_i A_i(\mathbf{z})$, where $A_i, i = 1,\dots,m$ is a set of known appearance variation images and $\lambda_i, i = 1,\dots,m$ is a set of unknown appearance parameters [6]. The set of known appearance variation images $A_i, i = 1,\dots,m$, were usually computed by applying principal component analysis (PCA) to a set of training images and keeping the eigenvectors which correspond to the top 95% of the energy. Therefore, by including the appearance variation we want to minimise

$$\left\| \begin{array}{l} T(\mathcal{W}(\mathbf{z};\Delta\mathbf{p})) - Y(\mathcal{W}(\mathbf{z};\mathbf{p})) \\ + \sum_{i=1}^m (\lambda_i + \Delta\lambda_i) A_i(\mathcal{W}(\mathbf{z};\Delta\mathbf{p})) \end{array} \right\|^2 \tag{15}$$

simultaneously with respect to $\Delta\mathbf{p}$ and $\boldsymbol{\Delta}\lambda = (\Delta\lambda_1,\dots,\lambda_m)^T$, and then updating the warp $\mathcal{W}(\mathbf{z};\mathbf{p}) \leftarrow \mathcal{W}(\mathcal{W}(\mathbf{z};\Delta\mathbf{p})^{-1};\mathbf{p})$ and the appearance parameters $\lambda \leftarrow \lambda + \Delta\lambda$. The appearance parameters are just additively updated as composition has no meaning for them [6]. As the expression in Equation 15 is non-linear, we need to linearise it by performing a first order Taylor expansion on $T(\mathcal{W}(\mathbf{z};\Delta\mathbf{p}))$ and $A_i(\mathcal{W}(\mathbf{z};\Delta\mathbf{p}))$. This yields

$$\left\| \begin{array}{l} T(\mathbf{z}) + \nabla T^T \frac{\partial \mathcal{W}(\mathbf{z};\mathbf{0})}{\partial \mathbf{p}}^T \Delta\mathbf{p} - Y(\mathcal{W}(\mathbf{z};\mathbf{p})) \\ + \sum_{i=1}^m \lambda_i A_i(\mathbf{z}) + \sum_{i=1}^m \Delta\lambda_i A_i(\mathbf{z}) \\ + \sum_{i=1}^m \lambda_i \nabla A_i{}^T \frac{\partial \mathcal{W}(\mathbf{z};\mathbf{0})}{\partial \mathbf{p}}^T \Delta\mathbf{p} \\ + \sum_{i=1}^m \Delta\lambda_i \nabla A_i{}^T \frac{\partial \mathcal{W}(\mathbf{z};\mathbf{0})}{\partial \mathbf{p}}^T \Delta\mathbf{p} \end{array} \right\|^2 \tag{16}$$

For this expansion, we can see that the last term $\sum_{i=1}^m \Delta\lambda_i \nabla A_i{}^T \frac{\partial \mathcal{W}(\mathbf{z};\mathbf{0})}{\partial \mathbf{p}}^T \Delta\mathbf{p}$ is a second-order term which makes solving for $\Delta\mathbf{p}$ and $\boldsymbol{\Delta}\lambda$ prohibitive (see Appendix A for details). As such this term is neglected which addressed later on in the next subsection. For the IC algorithm, neglecting this term simplifies the expression to:

$$\left\| \begin{array}{l} T(\mathbf{z}) - Y(\mathcal{W}(\mathbf{z};\mathbf{p})) + \sum_{i=1}^m \lambda_i A_i(\mathbf{z}) \\ + (\nabla T + \sum_{i=1}^m \lambda_i \nabla A_i)^T \frac{\partial \mathcal{W}(\mathbf{z};\mathbf{0})}{\partial \mathbf{p}}^T \Delta\mathbf{p} \\ + \sum_{i=1}^m A_i(\mathbf{z}) \Delta\lambda_i \end{array} \right\|^2 \tag{17}$$

From this we can determine the $(n + m) \times N$ dimensional steepest-descent images as

$$\mathbf{SD}_{\mathrm{simIC}}(\mathbf{z}) = \left[\begin{array}{c} \frac{\partial \mathcal{W}(\mathbf{z};\mathbf{0})}{\partial \mathbf{p}} (\nabla T + \sum_{i=1}^m \lambda_i \nabla A_i) \\ [A_1(\mathbf{z}),\dots,A_m(\mathbf{z})]^T \end{array} \right] \tag{18}$$

[3] It is worth noting that when we are finding the Jacobian matrix $\frac{\partial \mathcal{W}(\mathbf{z};\mathbf{0})}{\partial p} \neq \frac{\partial \mathbf{z}}{\partial \mathbf{p}}$ as the Jacobian is found as $\frac{\partial \mathcal{W}(\mathbf{z};\mathbf{0})}{\partial \mathbf{p}} = \frac{\partial \mathcal{W}(\mathbf{z};\mathbf{p})}{\partial \mathbf{p}}|_{\mathbf{p}=0}$

where $\mathbf{q} = [\mathbf{p}, \lambda]^T$ and $\Delta\mathbf{q} = [\Delta\mathbf{p}, \mathbf{\Delta}\lambda]^T$, and where $\mathbf{q}$ is the associated $n+m$ dimensional column vector. The error image can then be computed as

$$\mathbf{E}_{\text{simIC}}(\mathbf{z}) = T(\mathbf{z}) + \sum_{i=1}^{m} \lambda_i A_i(\mathbf{z}) - Y(\mathcal{W}(\mathbf{z};\mathbf{p})) \tag{19}$$

The change in the warp parameters $\mathbf{p}$ and λ can therefore be found by solving for the $\Delta\mathbf{q}$ which minimises

$$\left[\mathbf{E}_{\text{simIC}}(\mathbf{z}) + \mathbf{SD}_{\text{simIC}}(\mathbf{z})^T \Delta\mathbf{q}\right]^2 \tag{20}$$

with respect to $\Delta\mathbf{q}$, which results in

$$\Delta\mathbf{q} = -H^{-1}\mathbf{SD}_{\text{simIC}}(\mathbf{z})\mathbf{E}_{\text{simIC}}(\mathbf{z}) \tag{21}$$

where the Hessian matrix is

$$H = \mathbf{SD}_{\text{simIC}}(\mathbf{z})\mathbf{SD}_{\text{simIC}}(\mathbf{z})^T \tag{22}$$

It should be noted that in the simultaneous IC algorithm, in addition to it being an approximation, the computational benefit of the inverse-composition is lost as the Hessian matrix has to be recomputed every iteration as the appearance parameters are included in the steepest-decent image calculation.

B. Simultaneous FA Algorithm

Incorporating the appearance variation within the FA algorithm is performed in the same manner as the IC algorithm, where we include the appearance variation in the template. This changes the original FA minimisation equation in (4) to

$$\left\| Y(\mathcal{W}(\mathbf{z};\mathbf{p}+\Delta\mathbf{p})) - T(\mathbf{z}) - \sum_{i=1}^{m}(\lambda_i + \Delta\lambda_i)A_i(\mathbf{z}) \right\|^2 \tag{23}$$

As we can see from the above equation, we only have to linearise one term and not two like for the simultaneous IC scenario. Linearising this equation gives us

$$\left\| \begin{array}{l} Y(\mathcal{W}(\mathbf{z};\mathbf{p})) + \nabla Y^T \frac{\partial \mathcal{W}(\mathbf{z};\mathbf{p})}{\partial \mathbf{p}}^T \Delta\mathbf{p} - T(\mathbf{z}) \\ \sum_{i=1}^{m} \lambda_i A_i(\mathbf{z}) - \sum_{i=1}^{m} A_i(\mathbf{z})\Delta\lambda_i \end{array} \right\|^2 \tag{24}$$

which results in no second-order terms, which was the case in the simultaneous IC algorithm. Solving this equation by minimising with respect to $\Delta\mathbf{q}$, gives us

$$\Delta\mathbf{q} = H^{-1}\mathbf{SD}_{\text{simFA}}(\mathbf{z})\mathbf{E}_{\text{simFA}}(\mathbf{z}) \tag{25}$$

where the Hessian matrix is

$$H = \sum_{\mathbf{z}} \mathbf{SD}_{\text{simFA}}(\mathbf{z})\mathbf{SD}_{\text{simFA}}(\mathbf{z})^T \tag{26}$$

and the steepest-decent images are

$$\mathbf{SD}_{\text{simFA}}(\mathbf{z}) = \left[\begin{array}{c} \frac{\partial \mathcal{W}(\mathbf{z};\mathbf{p})}{\partial \mathbf{p}} \nabla T \\ [A_1(\mathbf{z}), \ldots, A_m(\mathbf{z})]^T \end{array} \right] \tag{27}$$

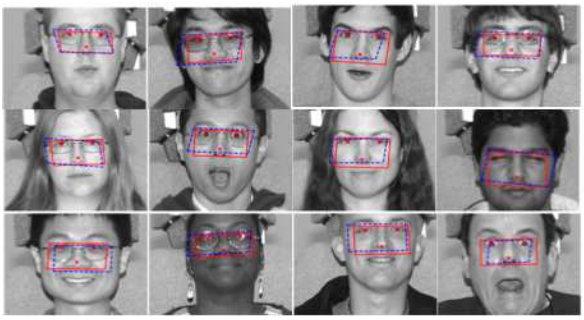
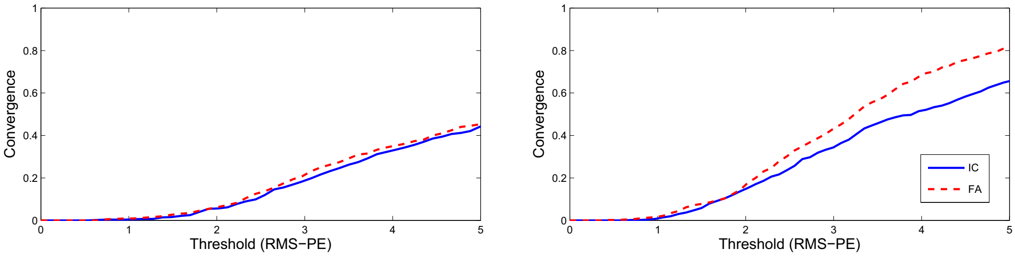


Fig. 2. Example images from the MultiPIE database [8] used in our alignment experiments. Eye and nose ground-truth points are denoted on each face with red “x’s” with the resulting red solid line bounding box taken around those coordinates. A blue dashed line bounding box denotes an instance where the ground-truth points are perturbed by random noise ranging between 5-10 root mean squared point error (RMS-PE). The images within the MultiPIE database exhibit a large amount of expression variation making alignment challenging.

The error image is described by Equation (19). In the simultaneous FA algorithm, we update both the warp parameters $\mathbf{p}$ and appearance parameters λ additively (i.e. $\mathbf{p} \leftarrow \mathbf{p} + \mathbf{\Delta p}$ and $\lambda \leftarrow \lambda + \Delta\lambda$). Like the other algorithms, we continue this iterative process until convergence or the maximum number of iterations is met. In the next section we show in our experiments that even though the original FA and IC algorithms yield the same alignment performance, the same can not be said for the simultaneous algorithms due to the appearance variables.

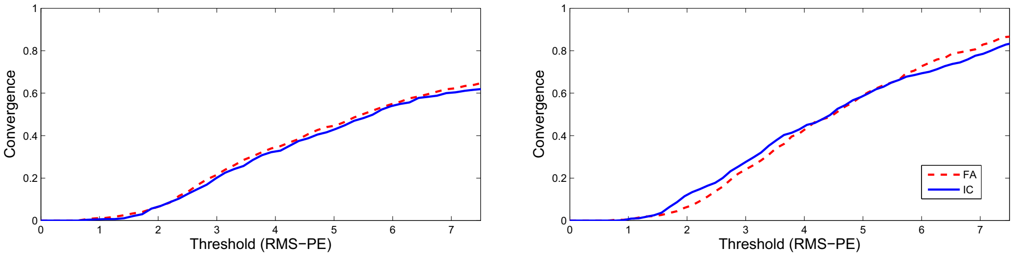
IV. Experiments

All experiments in this paper were conducted on the frontal portion of the MultiPIE face database [8]. A total of 1128 images were employed in this subset of the database taken across 141 subjects. Face images varied substantially in expression (see Fig. 2 for examples) making detection and alignment quite challenging. This data set was separated into subject independent training and testing sets with 560 and 568 images in each set respectively. In our experiments we assume that all warped images are referenced as an 80 × 80 image. We obtained the poor initial alignment by synthetically adding affine noise to the ground-truth coordinates of the face (located at the eyes and nose). We randomly generated affine warps $\mathcal{W}(\mathbf{z};\mathbf{p})$ in the following manner. We used the top left corner (0, 0), the top right corner (79, 0) and the center bottom pixel (39, 79). We then perturbed these points with a vector generated from white Gaussian noise. The magnitude of this perturbation was controlled to give a desired root mean squared (RMS) pixel error (PE) from the ground-truth coordinates. During learning, the initially misaligned images were defined to have between 5-10 RMS-PE. This range of perturbation was chosen as it approximately reflected the range of alignment error error seen when employing the freely available OpenCV [9] face detector (see Fig. 1 for examples of this perturbation). In all experiments we conducted in this paper our warped training and testing examples are assumed to have zero bias and unit gain.



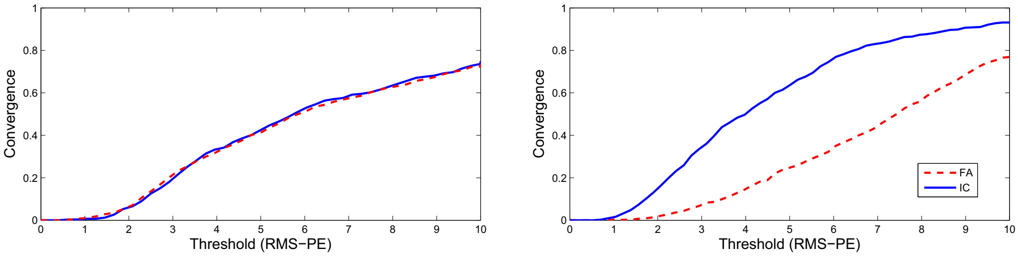
(a) (b)

Fig. 3. Alignment results showing the difference between the FA and IC algorithms without (a) and with (b) appearance variation included in the template with a RMS-PE of 5.



(a) (b)

Fig. 4. Alignment results showing the difference between the FA and IC algorithms without (a) and with (b) appearance variation included in the template with a RMS-PE of 7.5.



(a) (b)

Fig. 5. Alignment results showing the difference between the FA and IC algorithms without (a) and with (b) appearance variation included in the template with a RMS-PE of 10.

To gauge what affect solving both for alignment and appearance has on the alignment performance on unseen faces, we compared the simultaneous FA and IC algorithms to just the normal FA and IC algorithms (no appearance variation). To do this we used an alignment convergence curve (ACC) for 5, 7.5 and 10 RMS-PE. These curves are shown in Figures 3, 4 and 5 respectively. These curves have a threshold distance in RMS-PE on the x-axis and the probability of trials that achieved convergence (i.e., final alignment RMS-PE below the threshold) on the y-axis. A perfect alignment algorithm would receive an ACC that has 1 (or 100%) convergence for all threshold values.

Across all these results, it should first be noted that for the various initial perturbations, both the FA and IC algorithms without appearance variation achieve the same alignment performance (Figures 3(a), 4(a) and 5(a)). This backs up the results by Baker et al. in [7]. However, the interesting results come when the appearance variation parameters are included. Specifically in Fig. 3(b) where adding the appearance variation sees both the FA and IC algorithms improve in alignment convergence over Fig. 3(a), with the simultaneous FA outperforming the IC counterpart by quite a margin ($\sim 8\%$). A major reason for this is that due to the simultaneous FA algorithm being a truer solution than the simultaneous IC algorithm (Section III(b)), this results in greater refinement.

In Fig. 4(b), it can be seen that including the appearance variation does improve the alignment convergence over the normal FA and IC algorithms (Fig. 4(a)) as well. However, both the simultaneous FA and IC algorithms achieve approximately the same performance, which is different from the results in Fig. 3(b). A possible explanation of this maybe that even though the simultaneous FA algorithm may provide greater alignment as seen in Fig. 3(b), the fact that the initial error is farther away ($\sim$ 7.5 pixels) may mean that this improvement is somewhat diminished as sometimes the algorithm may get caught in local minima. It appears that due formulation of the simultaneous IC algorithm including the appearance variation in the steepest-decent images (compare Equations (18) to (27)), more of a blurring effect occurs within this process, making it less prone of to getting caught in local minima. The trade-off with this though is that closer refinement is difficult to achieve, as seen in Fig. 3(b).

This above hypothesis is somewhat reinforced by the results in Fig. 5(b) with an initial error of 10 RMS-PE. For this experiment, only the simultaneous IC algorithm achieved a improvement in alignment convergence using the appearance variation parameters, with the simultaneous FA algorithm achieving the same as the original FA algorithm. We believe this to be caused again by the algorithm getting caught in local minima. Again, a possible explanation of the simultaneous IC algorithm being superior for when the initial error is large, is that due to it including the appearance variation in the steepest-decent images, it can circumvent this problem.

V. Conclusion

In this paper we presented a comparison between the simultaneous FA and IC algorithms, which has not been done before. In this comparison we showed that these two algorithms were not equivalent theoretically due to the inclusion of the appearance variation parameters into the template image. As the role of the template is switched between the FA and IC algorithms, this resulting in the simultaneous IC algorithm containing second-order terms, which had to be omitted. This switching of roles also affected the calculation of the steepest-decent images for both algorithms, with the simultaneous IC algorithm having both the gradients of the template and appearance images within them compared to just the template for the simultaneous FA algorithm. Due to this we found through our experiments on the MultiPIE database that the simultaneous FA algorithm performance significantly better when the initial alignment error was low ($\sim$ 5 RMS-PE) as the gradients were much finer, while the simultaneous IC gradients were much coarser. This can also account for the results when the initial error was larger, with the results of the simultaneous FA algorithm diminishing as it is more likely to get caught in local minima with the finer gradients. Conversely, with the coarser gradients in the simultaneous IC algorithm, the alignment performance improved.

VI. Acknowledgements

This research was supported by the Australian Research Council Grant No:LP0562101.

References

[1] S. Baker and I. Matthews, "Equivalence and efficiency of image alignment algorithms," in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, Kauai Island, HI, USA, 2001, pp. 1090–1097.

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

[3] I. Matthews and S. Baker, "Active appearance models revisited," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 135–164, 2004.

[4] R. Gross, S. Baker, and I. Matthews, "Generic vs. person specific active appearance models," *Image and Vision Computing*, vol. 23, no. 11, pp. 1080–1093, 2005.

[5] M. Black and A. Jepson, "Eigentracking: Robust matching and tracking of articulated objects using a view-based representation," *International Journal of Computer Vision*, vol. 36, no. 2, pp. 101–130, 1998.

[6] S. Baker, R. Gross, and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework: Part 3," Carnegie Mellon University, Robotics Institute, Tech. Rep. CMU-RI-TR-03-35, November.

[7] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework: Part 1," Carnegie Mellon University, Robotics Institute, Tech. Rep. CMU-RI-TR-02-16, July.

[8] R. Gross, S. Baker, and T. Kanade, "The CMU multiple pose, illumination and expression (MultiPIE) database," Carnegie Mellon University, Robotics Institute, Tech. Rep. CMU-RI-TR-07-08, 2004.

[9] Open Source Computer Vision Library, `http://www.intel.com/research/mrl /research/opencv`

VII. Appendix A

Section III illustrates the derivation of the Simultaneous Inverse Composition Algorithm for image alignment including appearance variation. The derivation presented involves neglecting a second order term in order to make a solution possible. Here, we expand what was briefly mentioned in Section III by illustrating to the reader the complications which arise by retaining the second order term.

Equation 16 (reproduced in Equation 28) is the expression which contains the second order term.

$$\left[\begin{array}{l} T(\mathbf{z}) + \nabla T^T \frac{\partial \mathcal{W}(\mathbf{z};\mathbf{0})}{\partial \mathbf{p}}^T \Delta\mathbf{p} - Y(\mathcal{W}(\mathbf{z};\mathbf{p})) \\ + \sum_{i=1}^m \lambda_i A_i(\mathbf{z}) + \sum_{i=1}^m \lambda_i \nabla A_i{}^T \frac{\partial \mathcal{W}(\mathbf{z};\mathbf{0})}{\partial \mathbf{p}}^T \Delta\mathbf{p} \\ + \sum_{i=1}^m \Delta\lambda_i A_i(\mathbf{z}) + f(\Delta\mathbf{p}, \Delta\lambda) \end{array}\right]^2 \tag{28}$$

where $\Delta\lambda = [\Delta\lambda_1, ..., \Delta\lambda_m]^T$ and $f(\Delta\mathbf{p}, \Delta\lambda)$ is the second order term

$$f(\Delta\mathbf{p}, \Delta\lambda) = \sum_{i=1}^m \Delta\lambda_i \left[\frac{\partial \mathcal{W}(\mathbf{z};\mathbf{0})}{\partial \mathbf{p}} \nabla A_i\right]^T \Delta\mathbf{p} \tag{29}$$

.

In order to find the $\Delta\mathbf{p}$ and $\Delta\lambda$ that minimise Equation 28 using Gauss-Newton gradient descent, the partial derivatives of Equation 28 are required.

The partial derivatives for all terms in Equation 28 except for the the partial derivatives of $f()$ are shown in Section III. The partial derivatives of $f()$ are shown in Equations 30 and 31 respectively.

$$\frac{\partial f()}{\partial \Delta\mathbf{p}} = \sum_{i=1}^m \Delta\lambda_i \frac{\partial \mathcal{W}(\mathbf{z};\mathbf{0})}{\partial \mathbf{p}} \nabla A_i \tag{30}$$

$$\frac{\partial f()}{\partial \Delta\lambda_i} = \Delta\mathbf{p}^T \frac{\partial \mathcal{W}(\mathbf{z};\mathbf{0})}{\partial \mathbf{p}} \nabla A_i \tag{31}$$

As both partial derivatives are still functions of $\Delta\mathbf{p}$ and $\Delta\lambda$, the $\Delta\mathbf{q}$ which minimises Equation 28 requires obtaining $\Delta\mathbf{q}$ which satisfies the equation

$$0 = \left(\mathbf{SD}_{\text{simIC}} + \frac{\partial f(\Delta\mathbf{q})}{\partial \Delta\mathbf{q}}\right) \left[\begin{array}{c} \mathbf{E}_{\text{simIC}} + \mathbf{SD}_{\text{simIC}}^T(\mathbf{z})\Delta\mathbf{q} \\ + f(\Delta\mathbf{q}) \end{array}\right] \tag{32}$$

Minimising this expression is at best, non-trivial, and is the justification for the removal of the second order term.