Face Refinement through a Gradient Descent Alignment Approach

Simon Lucey, Iain Matthews

Robotics Institute, Carnegie Mellon University
Pittsburgh, PA 15213, USA
Email: slucey@ieee.org, iainm@cs.cmu.edu

Abstract

The accurate alignment of faces is essential to almost all automatic tasks involving face analysis. A common paradigm employed for this task is to exhaustively evaluate a face template/classifier across a discrete set of alignments (typically translation and scale). This strategy, provided the template/classifier has been trained appropriately, can give one a reliable but "rough" estimate of where the face is actually located. However, this estimate is often too poor to be of use in most face analysis applications (e.g. face recognition, audio-visual speech recognition, expression recognition, etc.). In this paper we present an approach that is able to *refine* this initial rough alignment using a gradient descent approach, so as to gain adequate alignment. Specifically, we propose an efficient algorithm which we refer to as the *sequential algorithm*, which is able to obtain a good balance between alignment accuracy and computational efficiency. Experiments are conducted on frontal and non-frontal faces.

Keywords: Face Alignment, Gradient Descent Object Alignment, Inverse Compositional Algorithm.

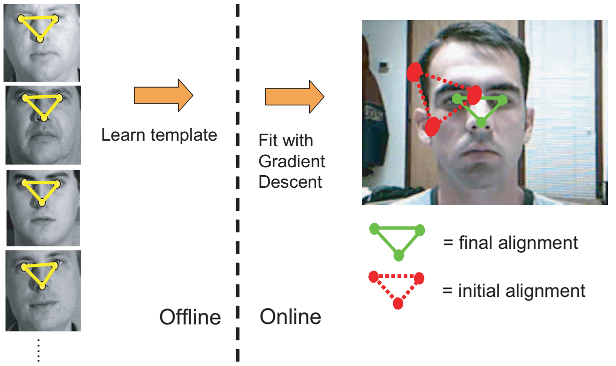


Figure 1: This figure depicts the task we want to undertake in this paper, where we have a rough approximation to where an object is (for our work in this paper the object will be a face), and we want to improve this alignment through a gradient descent fit. Faces naturally contain appearance variation, so we need to generalize from an offline ensemble of aligned face images so as to align to previously unseen subjects.

1 Introduction

Discriminative classifiers have been used with great success in the area of object detection. Most of these approaches, however, have concentrated on simply training a classifier with positive (i.e., aligned) and negative (i.e., not aligned) example images of the object. This classifier is then used to detect an object in a given image by exhaustively searching through all possible translations and scales. The now popular work (Viola & Jones 2001) of Viola and Jones is a prime example of this type of approach to object alignment. Such approaches are useful for obtaining a rough estimate of where the object is in an image, but struggle when one requires an alignment with more degrees of freedom than just translation and scale; such as an affine warp.

Another option after applying an exhaustive face detector is to exhaustively search for descriptors within the object that are largely invariant to affine variations. Typically in frontal face detection the eye region has been used in this capacity to great effect. Notable examples of these type of approaches have been (Moghaddam & Pentland 1997, Everingham & Zisserman 2006, Rurainsky & Eisert 2004, Wang & Ji 2005, Lowe 1999). A criticism of this approach, however, is that the availability of these invariant descriptors is not always assured and is very specific to the object being aligned. For example, faces undergoing view-point change often change appearance dramatically requiring the selection of different descriptors. The selection of these affine invariant descriptors is often based on heuristics, and it is still largely an open question how these descriptors change across view-point.

Gradient descent methods for object alignment, such as the Lucas-Kanade (LK) (Lucas & Kanade 1981) and the Inverse-Compositional (IC) (Baker & Matthews 2001) algorithms, provide a natural solution to these dilemmas for two reasons. First, they attempt to find a gradient descent solution to the optimal object alignment without having to resort to an impractical exhaustive search. Second, they also provide the desirable property of treating all objects in a unified way, thus not requiring any heuristically chosen affine invariant descriptors to be selected (e.g., eye detectors). A problem, however, is that gradient descent approaches have poor generalization properties when they have to deal with previously unseen intra-class object variation (Gross, Baker & Matthews 2005). A prime example of this problem is when one is trying to align a previously unseen face given one has a rough idea of where the face is located. A graphical depiction of this task can be seen in Figure 1. Unfortunately, discriminative learning methods cannot be as freely applied with gradient descent approaches as with exhaustive methods. Since gradient descent methods are inherently iterative one cannot treat misaligned images as just negative examples. Such images may be part of the solution trajectory. Inhibiting these images may actually stop the algorithm progressing towards the correct alignment.

The problem of dealing with appearance variation in gradient descent object alignment is not new. Most notably, Black and Jepson addressed the problem for the case of general appearance variation (Black & Jepson 1998). Similarly, Hager and Belhumer conducted work for the more specific case of illumination variation (Hager & Belhumeur 1998). In recent work (Baker, Gross & Matthews 2003), Baker et al. presented a unifying framework in which much of this previous work could be subsumed. Additionally, Baker et al. proposed two broad strategies for dealing with appearance variation when performing gradient descent object alignment, specifically the *simultaneous* and *project-out* algorithms. The simultaneous algorithm is able to give good alignment accuracy, but is computationally slow due to the large matrix inversions that must be performed at each iteration. Conversely, the project-out algorithm is computationally fast, due to simplifying assumptions, but suffers from poorer alignment[1] performance.

In this paper we propose a new algorithm that is able to give good alignment accuracy with reasonable computationally efficiency. We refer to this approach as the *sequential algorithm*. The task we use throughout this paper to evaluate these approaches is face alignment, where we want to be able to accurately align, using an affine warp, a subject independent template to all faces; even if that face has not been previously seen offline. Experiments are conducted on frontal and non-frontal faces, demonstrating large improvements in alignment over canonical approaches.

2 The Inverse Compositional Algorithm

The Lucas-Kanade (LK) algorithm (Lucas & Kanade 1981) has become a common tool in computer vision for the task of image alignment. The inverse compositional (IC) image alignment algorithm (Baker & Matthews 2001), developed by Baker and Matthews, is a more efficient formulation of the LK algorithm. In the IC formulation many computationally costly components of the algorithm can be pre-computed from the template, unlike the LK algorithm. The IC algorithm is essentially the minimization of the following with respect to $\Delta\mathbf{p}$,

$$||\mathbf{y}^{(\mathbf{p})} - \mathbf{t}^{(\mathbf{0})} - \mathbf{J}(\mathbf{t}^{(\mathbf{0})})\Delta\mathbf{p}||^2 \quad (1)$$

where $\mathbf{y}^{(\mathbf{p})}$ is the vectorized form of the image $Y(\mathcal{W}(\mathbf{x},\mathbf{p}))$, and $\mathbf{t}^{(\mathbf{0})}$ is the vectorized image of $T(\mathcal{W}(\mathbf{x},\mathbf{0}))$ which is an approximation to the aligned image $Y(\mathcal{W}(\mathbf{x},\mathbf{p}^*))$. An alignment function $\mathcal{W}(\mathbf{x},\mathbf{p})$ is employed to map an image position $\mathbf{x}$ to a new position $\mathbf{x}'$ based on the warp parameters $\mathbf{p}$, where $\mathbf{p}^*$ is the correct alignment we are attempting to estimate. Since the warp $\mathcal{W}(\mathbf{x},\mathbf{p})$ is non-linear we must approximate it using the linear matrix,

$$\mathbf{J}(\mathbf{t}) = [\nabla T(\mathcal{W}([0,0]^T,\mathbf{0}))\frac{\partial\mathcal{W}}{\partial\mathbf{p}},\ldots,$$
$$\nabla T(\mathcal{W}([N-1,M-1]^T,\mathbf{0}))\frac{\partial\mathcal{W}}{\partial\mathbf{p}}]^T \quad (2)$$

where the image T is an $N \times M$ image. For ease of notation $\mathbf{t}$ was used in Equation 2, rather than $\mathbf{t}^{(\mathbf{0})}$ because the $(\mathbf{0})$ represents the identity warp $\mathcal{W}(\mathbf{x},\mathbf{0})$; this convention will be used throughout the rest of this paper. One can see the approximation being used in Equation 2 is a first order Taylors series approximation to the warp.

It is easy to show that the solution to Equation 1 is,

$$\Delta\mathbf{p} = (\mathbf{J}(\mathbf{t})^T\mathbf{J}(\mathbf{t}))^{-1}\mathbf{J}(\mathbf{t})^T(\mathbf{y}^{(\mathbf{p})} - \mathbf{t}) \quad (3)$$

Due to the linear approximation in Equation 2 this solution is not explicit so we must iterate until we get convergence. This form of optimization is commonly referred to as Gauss-Newton optimization. The warp parameter $\mathbf{p}$ corresponds to the current estimate of the set of warp parameters needed to bring the two images into alignment, and $\Delta\mathbf{p}$ is the warp update that will improve the alignment. One can then update the warp estimate as follows:

$$\mathcal{W}(\mathbf{x};\mathbf{p}) \leftarrow \mathcal{W}(\mathbf{x};\mathbf{p}) \circ \mathcal{W}(\mathbf{x};\Delta\mathbf{p})^{-1} \quad (4)$$

from which we then obtain our new $\mathbf{y}^{(\mathbf{p})}$; this entire process is iterated until we obtain convergence for $\mathbf{p}$. The compositional update is required, as opposed to a simple additive update, because we are solving for the incremental warp update $\mathcal{W}(\mathbf{x};\Delta\mathbf{p})$ *not* the parameter update $\Delta\mathbf{p}$. This allows us to pre-compute our Jacobian in Equation 2 at $\mathcal{W}(\mathbf{x};\mathbf{0})$, rather than at each iteration from $\mathcal{W}(\mathbf{x};\mathbf{p})$; leading to sizeable computational savings. Please refer to (Baker & Matthews 2001) for more details.

2.1 The Simultaneous Algorithm

An immediate problem one can see with the IC algorithm denoted in Equation 1 is that we make the big assumption that the image $T(\mathcal{W}(\mathbf{x},\mathbf{0}))$ is a good approximation to $Y(\mathcal{W}(\mathbf{x},\mathbf{p}^*))$. Obviously, if the object being aligned has considerable intra-class appearance variation (e.g., faces), then this assumption can cause problems. To remedy this situation Baker et al. instead proposed the *simultaneous* algorithm (Baker et al. 2003) which attempts to minimize the following with respect to $\Delta\mathbf{q}$,

$$||\mathbf{y}^{(\mathbf{p})} - \mathbf{z} - \mathbf{Z}\Delta\mathbf{q}||^2 \quad (5)$$

where $\Delta\mathbf{q} = [\Delta\mathbf{p}^T, \Delta\boldsymbol{\lambda}^T]^T$ denotes a simultaneous updates in warp $\Delta\mathbf{p}$ and appearance $\Delta\boldsymbol{\lambda}$. We define,

$$\mathbf{z} = \mathbf{t} + \sum_{i=1}^{m} \lambda_i^{prev}\mathbf{a}_i \quad (6)$$

$$\mathbf{Z}_{\Delta\boldsymbol{\lambda}} = [\mathbf{a}_1,\ldots,\mathbf{a}_m] \quad (7)$$

and,

$$\mathbf{Z}_{\Delta\mathbf{p}} = \mathbf{J}(\mathbf{t}) + \sum_{i=1}^{m} \lambda_i^{prev}\mathbf{J}(\mathbf{a}_i) \quad (8)$$

where $\mathbf{a}_i$ refers to the ith appearance eigenvector, estimated through PCA from an offline ensemble of previously aligned objects (see Figure 1), $\boldsymbol{\lambda}^{prev}$ is the appearance from the previous iteration, and $\mathbf{t}$ denotes the mean appearance of the offline ensemble. In an analogous result to Equation 3 the solution to Equation 5 is,

$$\Delta\mathbf{q} = (\mathbf{Z}^T\mathbf{Z})^{-1}\mathbf{Z}^T(\mathbf{y}^{(\mathbf{p})} - \mathbf{z}) \quad (9)$$

where $\mathbf{Z} = [\mathbf{Z}_{\Delta\boldsymbol{\lambda}}, \mathbf{Z}_{\Delta\mathbf{p}}]$. The current warp $\mathbf{p}$ is updated by $\Delta\mathbf{p}$ according to the inverse compositional update in Equation 4 and the current appearance $\boldsymbol{\lambda} = [\lambda_1,\ldots,\lambda_m]^T$ is updated by,

[1] We would like to note that the project-out algorithm has been shown (Baker et al. 2003) to perform very well in situations where the appearance variation has been previously seen offline, and that the rank of this variation is small. In this paper we are investigating the more general case where the appearance variation has not been seen previously, and the rank of this variation is quite large.

$$\boldsymbol{\lambda} = \boldsymbol{\lambda}^{prev} + \Delta\boldsymbol{\lambda} \tag{10}$$

A problem with the simultaneous solution however, is that $\mathbf{z}$, $\mathbf{Z}$ and therefore $(\mathbf{Z}^T\mathbf{Z})^{-1}$ must be re-estimated at each iteration which slows down the algorithm dramatically. A simple speedup, which Baker et al. refer to as the *project-out* algorithm (Baker et al. 2003), can be found by assuming[2] that $\boldsymbol{\lambda}^{prev} = \mathbf{0}$ at each iteration which ensures $\mathbf{z}$ and $\mathbf{Z}$ remain constant. In reality since the appearance is not updated at each iteration then $\Delta\boldsymbol{\lambda}$ does not need to be found explicitly.

2.2 The Sequential Algorithm

Although effective, the simultaneous and project-out algorithms suffer some drawbacks due to the simplifying assumptions made in Equations 5-9. The true solution should be to try to solve $\Delta\mathbf{p}$ and $\boldsymbol{\lambda}$ simultaneously from,

$$||\mathbf{y}^{(\mathbf{p})} - \mathbf{t} - \mathbf{J}(\mathbf{t})\Delta\mathbf{p} - \sum_{i=1}^{m} \lambda_i(\mathbf{a}_i - \mathbf{J}(\mathbf{a}_i)\Delta\mathbf{p})||^2 \tag{11}$$

Unfortunately, one cannot solve this explicitly for $\Delta\mathbf{p}$ and $\boldsymbol{\lambda}$ so Baker et al. make the assumption that $\boldsymbol{\lambda} = \Delta\boldsymbol{\lambda} + \boldsymbol{\lambda}^{previous}$ where $\Delta\boldsymbol{\lambda}$ is the appearance update and $\boldsymbol{\lambda}^{previous}$ is the appearance from the previous iteration. Based on this assumption Baker et al. make the approximation,

$$\sum_{i=1}^{m} \lambda_i \mathbf{J}(\mathbf{a}_i) \approx \sum_{i=1}^{m} \lambda_i^{prev} \mathbf{J}(\mathbf{a}_i) \tag{12}$$

Thus allowing Equation 11 to be solved simultaneously for $\Delta\mathbf{p}$ and $\Delta\boldsymbol{\lambda}$ instead of for $\Delta\mathbf{p}$ and $\boldsymbol{\lambda}$. In this paper we propose a new approach that abandons the approximation in Equation 12 and attempts to solve Equation 11 directly but *not* simultaneously. We refer to this new approach as the *sequential* algorithm. We can pose this algorithm as minimizing with respect to $\mathbf{q}$ the following,

$$||\mathbf{y}^{(\mathbf{p})} - \mathbf{z}_{\mathbf{q}} - \mathbf{Z}_{\mathbf{q}}\mathbf{q}||^2 \tag{13}$$

where $\mathbf{q} \in \{\Delta\mathbf{p}, \boldsymbol{\lambda}\}$ as we are attempting to solve for $\Delta\mathbf{p}$ and $\boldsymbol{\lambda}$ sequentially. Both $\Delta\mathbf{p}$ and $\boldsymbol{\lambda}$ can be solved in a similar fashion to Equations 3 and 9 where,

$$\mathbf{q} = (\mathbf{Z}_{\mathbf{q}}^T\mathbf{Z}_{\mathbf{q}})^{-1}\mathbf{Z}_{\mathbf{q}}^T(\mathbf{y}^{(\mathbf{p})} - \mathbf{z}_{\mathbf{q}}) \tag{14}$$

First, we attempt to solve for $\Delta\mathbf{p}$ given that we know $\boldsymbol{\lambda}$ which we initially guess to be $\boldsymbol{\lambda} = \mathbf{0}$,

$$\mathbf{z}_{\Delta\mathbf{p}} = \mathbf{t} + \sum_{i=1}^{m} \lambda_i \mathbf{a}_i \tag{15}$$

$$\mathbf{Z}_{\Delta\mathbf{p}} = \mathbf{J}(\mathbf{t}) + \sum_{i=1}^{m} \lambda_i \mathbf{J}(\mathbf{a}_i) \tag{16}$$

Given that we have an estimate for $\Delta\mathbf{p}$, from Equation 14, we then obtain a new estimate of $\mathbf{y}^{(\mathbf{p})}$ by applying the inverse compositional warp in Equation 4. We can next solve for $\boldsymbol{\lambda}$ given our new estimate of $\Delta\mathbf{p}$ where,

$$\mathbf{z}_{\boldsymbol{\lambda}} = \mathbf{t} \tag{17}$$

and,

$$\mathbf{Z}_{\boldsymbol{\lambda}} = [\mathbf{a}_1, \ldots, \mathbf{a}_m] \tag{18}$$

The algorithm is iterated until $\mathbf{p}$ and $\boldsymbol{\lambda}$ reach convergence. As mentioned previously, the algorithm first solves for $\Delta\mathbf{p}$ then solves for $\boldsymbol{\lambda}$. The sequential algorithm offers substantial computational savings over the simultaneous algorithm. First, it factorizes the inversion of $(\mathbf{Z}^T\mathbf{Z})$ in Equation 9 into $(\mathbf{Z}_{\Delta\mathbf{p}}^T\mathbf{Z}_{\Delta\mathbf{p}})^{-1}$ and $(\mathbf{Z}_{\boldsymbol{\lambda}}^T\mathbf{Z}_{\boldsymbol{\lambda}})^{-1}$. Second, since $\mathbf{Z}_{\boldsymbol{\lambda}}$ contains only eigenvectors then $(\mathbf{Z}_{\boldsymbol{\lambda}}^T\mathbf{Z}_{\boldsymbol{\lambda}})^{-1} = \mathbf{I}$ thus making this inversion pointless, again adding considerably to the computational savings over the simultaneous algorithm. Finally, the number of image warps per iteration remains exactly the same as the simultaneous algorithm.

One could argue that their may be some benefit in first estimating $\boldsymbol{\lambda}$ then estimating $\Delta\mathbf{p}$, in that if the current estimate $\mathbf{p}$ is close to the true alignment $\mathbf{p}^*$ then estimating $\boldsymbol{\lambda}$ first will allow one to then gain a much more accurate $\Delta\mathbf{p}$. However, we contend this argument is very dependent on the assumption that your current $\mathbf{p}$ is close to $\mathbf{p}^*$. Empirically we found the more cautious view that $\mathbf{p}$ may be some distance away from $\mathbf{p}^*$ to give more robust results.

3 Frontal-Face Experiments

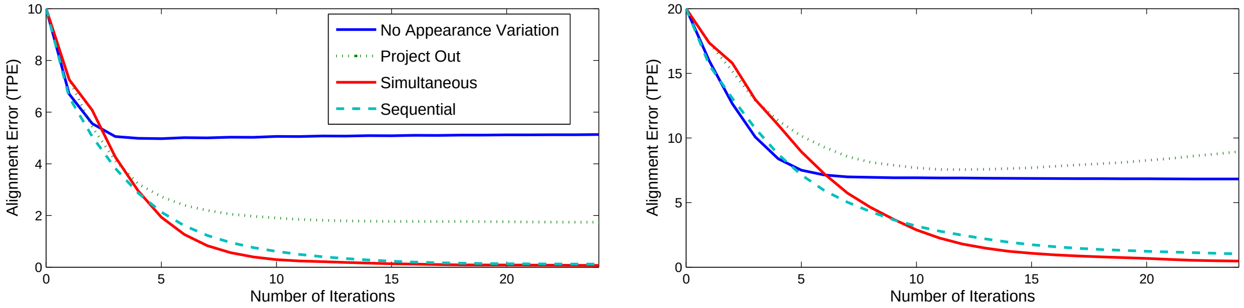
For our experiments with frontal faces we ran face alignment experiments on the FRGC 1.0 database that corresponded to the training portion of Experiment 1 (Phillips, Flynn, Scruggs, Bowyer, Chang, Hoffman, Marques, Jaesik & Worek 2005). Of the 152 images in this set 76 were used for learning the mean template and appearance variation (i.e. eigenvectors) and the other 76 were used for evaluation. All the images had three hand annotated fiducial points centered on the eyes and nose for ground truth.

3.1 Synthetic Alignment Noise

For our first lot of experiments an initial alignment error was introduced by adding random Gaussian noise to all three points so the *total point error* (TPE) was equal to: (a) 10 pixels and (b) 20 pixels. The TPE is defined as the total distance, in pixels, the current warp's points are from the ground-truth. The TPE is always taken with reference to the template, which was chosen to be of size 80 × 80 pixels. A comparison between the IC algorithms with: (i) *no appearance variation* (i.e., just the mean template), (ii) *project-out*, (iii) *simultaneous* and (iv) *sequential*, can be seen in Figures 2 and 3.

Figure 2 depicts the ideal scenario where all facial appearance variation has been observed offline previously. Figure 3 depicts the "real world" scenario where the facial appearance variation has not been observed previously. One can see in both figures that the simultaneous and sequential algorithms perform best in all cases. Interestingly, one can see in Figure 2, for the case where the appearance variation was seen offline, the final alignment error is almost zero. The biggest contrasts in performance between Figures 2 and 3 can be seen for the project-out algorithm. When the appearance variation has been previously observed, the final alignment error is reasonable. However, for the "unseen" scenario, where the appearance variation has not been observed, the final alignment error diverges. This poor result can be attributed to the assumptions made in the project-out

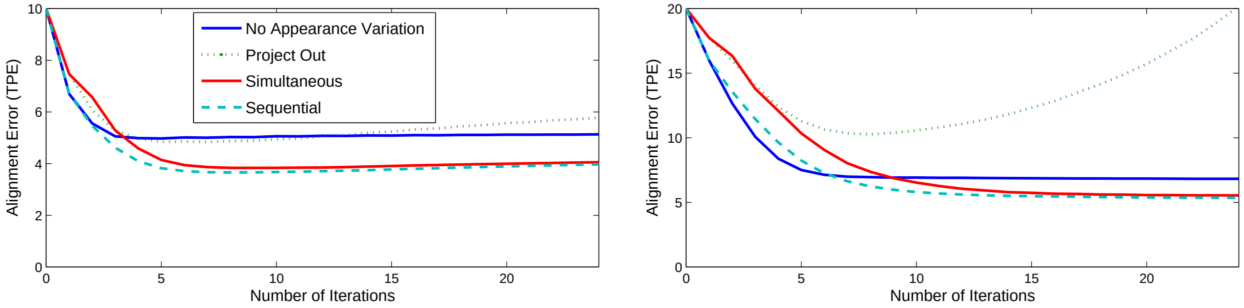
[2] Please note that the project-out algorithm mentioned here is a slight variation upon the one seen in (Baker et al. 2003), as we are solving for $\Delta\mathbf{p}$ and $\Delta\boldsymbol{\lambda}$ simultaneously rather than sequentially. Empirically however, we have found the performance of these two variants to be identical. Please refer to Appendix A for more details.



(a) Initial $TPE = 10$

(b) Initial $TPE = 20$

Figure 2: This figure depicts a comparison between IC algorithms when the appearance variation has been seen previously offline. Specifically we compare cases for: (i) *no appearance variation* (i.e. just the mean template), (ii) *project-out*, (iii) *simultaneous* and (iv) *sequential*. Results indicate that the simultaneous and sequential algorithms converge to almost perfect alignment. Our proposed sequential algorithm however, has considerably less computational load than the simultaneous algorithm.



(a) Initial $TPE = 10$

(b) Initial $TPE = 20$

Figure 3: This figure depicts a comparison between IC algorithms when the appearance variation has *not* been seen previously offline. Specifically we compare cases for: (i) *no appearance variation* (i.e. just the mean template), (ii) *project-out*, (iii) *simultaneous* and (iv) *sequential*. Results indicate that although both the simultaneous and sequential algorithms obtain the best alignment, performance is still poor when compared to the results seen in Figure 2 for the scenario when the appearance variation has been seen previously offline.

algorithm; namely that the appearance $\boldsymbol{\lambda}^{prev}$ at each iteration is always zero.

A major result from the experiments carried out in Figures 2 and 3 is the approximately equivalent performance of the sequential and simultaneous algorithms. This result is initially perplexing, as one would expect in most cases a simultaneous iterative solution to be more accurate, since we are solving for appearance and warp at the same time, than a sequential one. This result is consistent for when the initial alignment error is small ($TPE = 10$)and large ($TPE = 20$), as well as for the scenarios where the appearance variation was and was not observed respectively.

3.2 Viola-Jones Noise

In our next lot of experiments we decided to employ an exhaustive search face detector as an initializer, to get an indication of the advantages of our proposed system. The exhaustive search face detector we employed in our experiments was the publicly available implementation of the Viola-Jones face detector (Viola & Jones 2001) from the OpenCV library. The face detector outputs a bounding box defined by $[x, y, s]$, where $[x, y]$ defines the center of the box and s defines its scale. To gain a good “rough” estimate of where the fiducial points of the face are, based on this bounding box, a projection matrix is learnt that maps from this bounding box to the estimated fiducial face points. This projection matrix is learnt through least-squares optimization from an ensemble of offline aligned face images (see Figure 1).

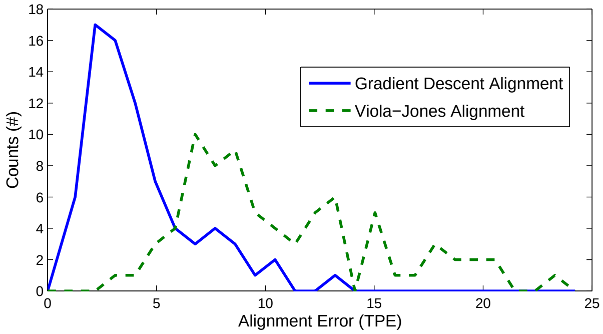


Figure 4: This figure depicts the distribution, in total point error (TPE), of the Viola-Jones face detector alignment and our proposed gradient descent alignment. One can see clearly that our approach both decreases the mean and variance of the TPE produced by the Viola-Jones detector alone.

Figure 4 depicts the distribution, in TPE, of the initial Viola-Jones and also the final distribution after we post-process these coordinates with our gradient descent method. One can clearly see that our method successfully reduces the mean TPE of the

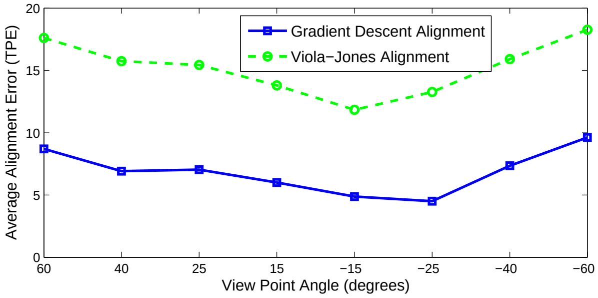


Figure 7: This figure depicts performance in terms of average total point error (TPE) for Viola-Jones alignment and our own gradient descent method across many view-points. One can see that our technique improves the average TPE across all poses.

Viola-Jones detector. An example of some of these improved alignments can be seen in Figure 5.

4 Non-Frontal Face Experiments

Experiments were performed on a subset of the FERET database (Phillips, Moon, Rizvi & Rauss 2000), specifically images stemming from the *ba*, *bb*, *bc*, *bd*, *be*, *bf*, *bg*, *bh*, and *bi* subsets; which approximately refer to rotation's about the vertical axis of 0^o, $+60^o$, $+40^o$, $+25^o$, $+15^o$, -15^o, -25^o, -40^o, -60^o respectively. The database contains 200 subjects in total, which were randomly divided into offline training and online testing sets both containing 100 subjects. In a similar fashion to the frontal face experiments templates for all poses were chosen to be of size 80×80 pixels.

In Figure 6 one can see results in terms of the average TPE across a number of different poses. We compare the TPE obtained from the Viola-Jones face detector and our gradient descent method. One can see in all cases our gradient descent method improves the average TPE. Although not perfect, our gradient descent refiner is able to substantially improve face alignment from multiple view-points. Examples of aligned images, from all poses, can be seen in Figure 5 for: (a) the Viola-Jones alignment, (b) our gradient descent alignment, and (c) the ground truth alignment.

5 Conclusion and Future Work

We presented a novel and effective approach to face refinement, on frontal and non-frontal faces, based on a gradient descent image alignment paradigm with appearance variation. Our approach is able to overcome some of the inherent computational difficulties associated with exhaustive search type object detectors when one wants to align an object with more degrees of freedom than just translation and scale. It is also a viable alternative to approaches that rely on affine invariant descriptors (e.g., the eyes) within the object, especially when the location and nature of these descriptors are unclear for the object (e.g., non-frontal faces). In this work we proposed an efficient extension to current algorithms in literature, which we refer to as the *sequential* algorithm. This approach was able to empirically deliver approximately the accuracy of the simultaneous algorithm with much less computational cost; making it of viable use in many real-time face processing applications that require human and computer interaction. As a proof of concept we were able to demonstrate how effectively our approach performs in conjunction with a Viola-Jones face detector on frontal and non-frontal faces. We want to extend our current work to deal with more alignment points and more complicated warps (e.g., piece-wise affine) involving faces across pose.

References

Baker, S., Gross, R. & Matthews, I. (2003), Lucas-kanade 20 years on: A unifying framework: Part 3., Technical Report CMU-RI-TR-03-01, Robotics Institute, Carnegie Mellon University.

Baker, S. & Matthews, I. (2001), Equivalence and efficiency of image alignment algorithms, *in* 'IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 1090–1097.

Black, M. & Jepson, A. (1998), 'Eigen-tracking: Robust matching and tracking of articulated objects using a view-based representation', *International Journal of Computer Vision* **36**(2), 101–130.

Everingham, M. & Zisserman, A. (2006), Regression and classification approaches to eye localization in face images, *in* 'International Conference on Automatic Face and Gesture Recognition', pp. 441–446.

Gross, R., Baker, S. & Matthews, I. (2005), 'Generic vs. person specific active appearance models', *Image and Vision Computing* **23**(11), 1080–1093.

Hager, G. D. & Belhumeur, P. N. (1998), 'Efficient region tracking with parametric models of geometry and illumination', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(10), 1025.

Lowe, D. G. (1999), Object recognition from local scale-invariant features, *in* 'IEEE International Conference on Computer Vision', Vol. 2, pp. 1150–1157.

Lucas, B. & Kanade, T. (1981), An iterative image registration technique with an application to stereo vision, *in* 'International Joint Conference on Artificial Intelligence', pp. 674–679.

Moghaddam, B. & Pentland, A. (1997), 'Probabilistic visual learning for object recognition', *IEEE Trans. PAMI* **19**(7), 696–710.

Phillips, P. J., Flynn, P. J., Scruggs, T., Bowyer, K. W., Chang, J., Hoffman, K., Marques, J., Jaesik, M. & Worek, W. (2005), Overview of the face recognition grand challenge, *in* 'Computer Vision and Pattern Recognition (CVPR)', pp. 947–954.

Phillips, P. J., Moon, H., Rizvi, S. A. & Rauss, P. J. (2000), 'The FERET evaluation methodology for face-recognition algorithms', *IEEE Trans. PAMI* **10**(22), 1090–1104.

Rurainsky, J. & Eisert, P. (2004), Eye center localization using adaptive templates, *in* 'CVPR Workshop on Face Processing in Video'.

Viola, P. & Jones, M. (2001), Rapid object detection using a boosted cascade of simple features, *in* 'IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)', Vol. 1, pp. 511–518.

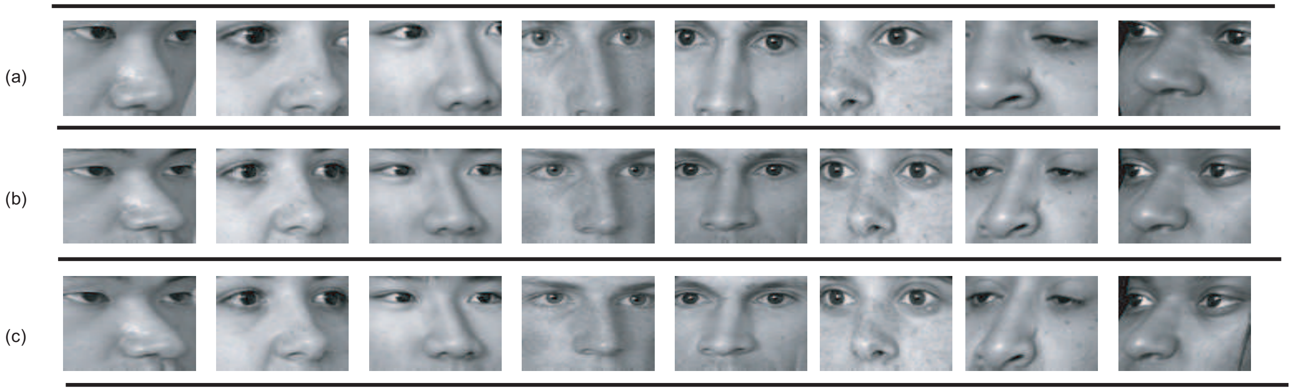
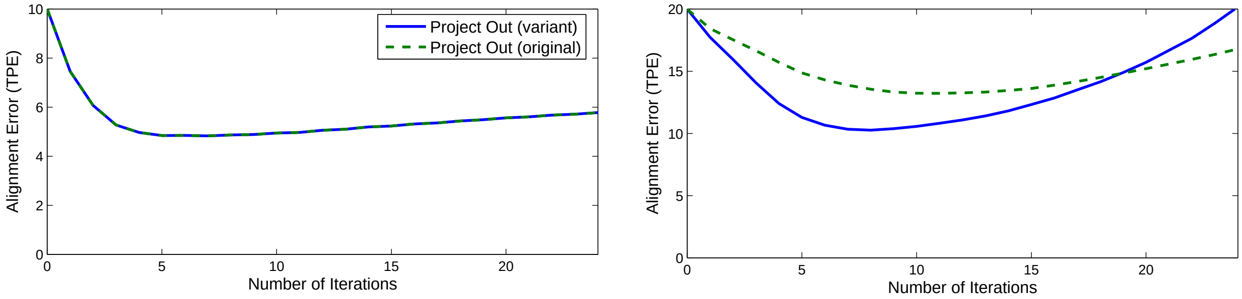


Figure 5: This figure contains examples images for: (a) Viola-Jones alignment, (b) gradient descent alignment, and (c) ground truth alignment. As one can see from these images, our algorithm performs a good job in estimating the correct alignment across a number of different view points.



(a) Initial $TPE = 10$

(b) Initial $TPE = 20$

Figure 6: This figure depicts a comparison between two variants of the project-out algorithm, where experiments were conducted on the frontal faces of the FRGC dataset. We note that for a smaller initial alignment error ($TPE = 10$) the performance of our variant and the original is identical (see (a)). However, for a larger initial alignment error ($TPE = 20$) the performance of our variant is superior to the original. Both approaches exhibit poor performance however, as they diverge from the initial alignment.

Wang, P. & Ji, Q. (2005), Learning discriminant features for multi-view face and eye detection, *in* 'IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 373–379.

Appendix

A Variants on Project-Out

In Section 2.1 we present a variant on the project-out algorithm (Baker et al. 2003) first proposed by Baker et al. We proposed in our variant that the formulation of the project-out algorithm can simply be interpreted as the normal simultaneous algorithm, with the exception that we assume $\boldsymbol{\lambda}^{prev}$ is equal to zero at each iteration. This assumption leads to large computational savings as there is no longer any need for costly matrix inversions at each iteration. This interpretation however, differs slightly to the original formulation of the project-out algorithm. The difference between our formulation and the original project-out algorithm lies in how we minimize,

$$||\mathbf{y}^{(p)} - \mathbf{t} - \mathbf{J}(\mathbf{t})\Delta\mathbf{p} - \sum_{i=1}^{m} \Delta\lambda_i \mathbf{a}_i||^2 \qquad (19)$$

with respect to $\Delta\mathbf{p}$ and $\Delta\boldsymbol{\lambda}$. In Baker et al.'s approach they decompose this problem further into the linear subspace span($\mathbf{a}_i$) spanned by the collection of vectors $\mathbf{a}_i$ and its orthogonal complement span$(\mathbf{a}_i)^\perp$. Baker et al.'s approach is now, with respect to $\Delta\mathbf{p}$ and $\Delta\boldsymbol{\lambda}$, attempting to minimize,

$$||\mathbf{y}^{(p)} - \mathbf{t} - \sum_{i=1}^{m} \Delta\lambda_i \mathbf{a}_i||^2_{\text{span}(\mathbf{a}_i)} + \\ ||\mathbf{y}^{(p)} - \mathbf{t} - \mathbf{J}(\mathbf{t})\Delta\mathbf{p}||^2_{\text{span}(\mathbf{a}_i)^\perp} \qquad (20)$$

where $||.||_L$ denotes the Euclidean L2 norm of a vector projected into the linear subspace L. Essentially this approach forces the optimization of $\Delta\mathbf{p}$ and $\Delta\boldsymbol{\lambda}$ into two disjoint spaces. One can see that the first term is always exactly zero because the term $\sum_{i=1}^{m} \Delta\lambda_i \mathbf{a}_i$ can represent any vector in span($\mathbf{a}_i$). As a result the simultaneous minimum over both $\Delta\mathbf{p}$ and $\Delta\boldsymbol{\lambda}$ can be found sequentially by minimizing the second term with respect to $\Delta\mathbf{p}$ alone, and then treating the optimal values of $\Delta\mathbf{p}$ as a constant to minimize the first term with respect to $\Delta\boldsymbol{\lambda}$.

The variant we employed in Section 2.1 actually solves Equation 19 simultaneously for $\Delta\mathbf{p}$ and $\Delta\boldsymbol{\lambda}$ rather than sequentially. Both approaches are extremely fast as nearly all steps can be pre-computed and they require no matrix inversion except in pre-computation. There is a slight computational advantage in Baker et al.'s original formulation as the final update matrix, which one multiplies the error image by, has a rank equal to the dimensionality of just the warp space; whereas our formulation employs an update matrix whose rank is equal to the dimensionality of the warp and appearance space. Empirically we found both approaches obtained identical performance when the initial alignment error is small (see Figure 7(a)), but there is some slight advantage in our approach when the initial alignment error is large (see Figure 7(b)); although in both cases performance did diverge. The experiments in Figure 7 were carried out on the frontal faces of the FRGC dataset.