Non-Rigid Face Tracking Using Short “Track-Life” Features

Simon Lucey
ICT Centre, CSIRO, Australia
simon.lucey@csiro.au

Jun-Su Jang
Robotics Institute, Carnegie Mellon University, USA
jsjang@cs.cmu.edu

Abstract

We define a “generic” non-rigid face tracker as any system that exhibits robustness to changes in illumination, expression and viewpoint during the tracking of facial landmarks in a video sequence. A popular approach to the problem is to detect/track an ensemble of local features over time whilst enforcing they conform to a global non-rigid shape prior. In general these approaches employ a strategy that assumes: (i) the feature points being tracked, ignoring occlusion, should roughly correspond across all frames, and (ii) that these feature points should correspond to the landmark points defining the non-rigid face shape model. In this paper, we challenge these two assumptions through the novel application of interest point detectors and descriptors (e.g. SIFT & SURF). We motivate this strategy by demonstrating empirically that salient features on the face for tracking on average only have a “track-life” of a few frames and rarely co-occur at the vertex points of the shape model. Due to the short track-life of these features we propose that new features should be detected at every frame rather than tracked from previous frames. By employing such a strategy we demonstrate that our proposed method has natural invariance to large discontinuous changes in motion. We additionally propose the employment of an online feature registration step that is able to rectify error accumulation and provides fast recovery from occlusion during tracking.

1. Introduction

In general there are two strategies to non-rigid face tracking: (i) *model-update* [3, 13] and (ii) *track-by-detection* [2, 4, 5, 19, 20]. The model-update strategy involves fitting a model of the face, derived from the previously tracked frame, to the current image frame. Based on this new fit one then updates the model, based on the properties of the currently tracked image, and then repeat the process for the next image frame in the sequence. A good example of this strategy in non-rigid face fitting can be found in the Active Appearance Models (AAMs) literature [3, 13] where the model update at each frame is constrained to be consistent with a linear shape model, and in many cases with an appearance model (i.e., the “simultaneous algorithm” [13]) of a typical face. As changes from image to image are often

Figure 1. In this paper we attempt to demonstrate that many useful features on the face for tracking only have a short “track life” (green points). That is they only co-occur on the face for few frames rather than co-occur across most frames (red points). Some features are spurious offering no correspondence between frames (yellow points). Additionally, all these feature points tend not to occur at the predefined vertex points used commonly to define a face (e.g., mouth corner, eye corner, etc.). To take advantage of this observation, we propose that new features for tracking should be detected at every frame, rather than tracked over time from the first initial frame.

small and predictable, fitting results of previous images in the sequence constitute prior knowledge that is used to bias and improve the model for the fitting process in subsequent images. However, this strategy has problems. Specifically, it can be cumulatively effected by registration errors in previous frames until the track becomes erroneous; a problem that is often referred to in computer vision literature as “drift”.

The track by detection strategy has found numerous expressions in non-rigid face fitting literature notably: Active Shape Models (ASMs) [2, 20], Constrained Local Models (CLMs) [4, 19] and Pictorial Structure Matching (PSM) [5]. In these approaches local detectors are learnt around fiducial points on the face with the responses of these fiducial detectors being constrained during the fit by the global non-rigid shape prior. Unlike model-update methods, these approaches do not suffer from drift as the model is static and does not change over time. These approaches have demonstrated excellent performance on faces in recent times, especially in the presence of large variations in illumination and identity, but perform poorly when generically tracking

subtle expression changes over time.

Of late in the vision community there has been much interest in the use of interest point detectors [1, 11, 15] for finding local correspondences between images. Notable methods like Scale Invariant Feature Transform (SIFT) [11] and Speeded Up Robust Features (SURF) [1] have found success through their ability to detect distinct regions of an image (e.g., corners, blobs, and T-junctions) and represent them as descriptor features that have high invariance to noise, detection errors, and geometric and photometric deformations. The goal of this paper is to employ these interest point detectors to investigate the “track-life” of local features in a video sequence containing a non-rigid face deforming over time. We define track-life as the number of frames a local feature exists for while still giving meaningful correspondence information between frames. When correspondence between frames no longer occurs the track-life of that feature is deemed to be over. As a result of this study we make the following contributions:-

- Demonstrate empirically salient local features on the face, as detected through methods like SIFT and SURF, have *short* track-lives. In our experiments a local feature on the face with correspondence between frames has a track-life on average of approximately 3-4 frames. Additionally, we note that the location of these salient local features rarely corresponds with the vertex positions on our face mesh. (Section 3)

- Based on this observation we propose a model-update strategy to non-rigid generic face tracking that enables the employment of these short track-life features. Our approach attempts to *detect* new features at every frame rather than track features from the previous frame so as to utilize all possible tracking information on the face. We empirically compare this approach to an AAM inspired method, based on the Lucas & Kanade algorithm [12], for model-update and demonstrate that the approach has dramatically better performance in the presence of discontinuous motion. (Sections 4 and 5)

- Finally, we propose a drift rectification strategy based on similar work [9, 16] conducted in 3D rigid face tracking. This approach is motivated through the concept that drift should be related to the amount of “new” information seen in a video sequence rather than the number of frames. Through this strategy we demonstrate a notable minimization of the effect of drift in a visual face sequence. We additionally demonstrate good invariance to occlusion and the ability to recover from errors when they occur. (Section 6).

Problem Definition: We should emphasize that all the work presented in this paper is aimed squarely at the space of non-rigid face tracking using a model-update paradigm. As a result all experiments conducted in this paper shall assume that the first frame of each video sequence has been initialized to the ground-truth position.

2. Background

Interest Point Detectors and Feature Descriptors: A plethora of interest point detectors have been described in literature [1, 8, 10, 11, 15]. They all share a similar goal to detect distinct regions of an image (e.g., corners, blobs, and T-junctions). The most valuable property of an interest point detector is its repeatability [1] (i.e. whether it reliably finds the same interest points under different viewing conditions). The most widely used detector in vision is probably the Harris corner detector [8], proposed in 1988, based on the eigenvalues of the second-moment matrix. However, Harris corners are not scale-invariant. Scale-adapted Harris measures were subsequently developed [10] to improve repeatability. A good review and comparison on interest point detectors can be found in [15].

Once an interest point region has been located a feature descriptor of that region must then be extracted that has high invariance to noise, detection errors, and geometric and photometric deformations. A wide variety of descriptors have been proposed [14] like Gaussian derivatives, moment invariants, complex features, steerable filters, phase-based local features, and descriptors representing the distribution of smaller-scale features within the interest point neighbourhood. The latter, introduced by Lowe [11], have been shown to outperform the others [14].

For the purposes of this paper we will be employing SIFT [11] and SURF [1] to detect interest points and obtain feature descriptors. The SIFT method was employed due to its high performance in both interest point detection [15] and robust feature descriptor extraction [14]. The similarly performing SURF [1] method was chosen due to its computational efficiency through the use of integral images during the Harris measure estimation. In our own experiments we were able to extract SURF features at over 35 frames per second on a 640×480 image sequence running on an Intel Dual Core 2.3GHz machine. Both SIFT and SURF methods generated 128 dimensional descriptor vectors. For the rest of this paper we shall employ an abuse of terminology as we shall refer to the SURF and SIFT methods as “interest point detectors” even though they are methods for interest point detection & feature descriptor extraction.

Matching Features: The matching between features in the source (previously tracked) and destination (currently tracked) frames is carried out as follows. An interest point in the destination image is compared to an interest point in the source image by calculating the Euclidean distance between their descriptor vectors. A matching pair is detected, if its distance is closer than 0.7 times the distance of the second nearest neighbour. This is the nearest neighbour ratio matching strategy in [1, 11]. Based on this matching a correspondence between these features is inferred.

Point Distribution Model: Apart from a select few excep-

tions most work on non-rigid face tracking employs a point distribution model (PDM) [3],

$$\mathbf{x}_i = \mathcal{W}(\mathbf{y}_i; \mathbf{p}) = s\mathbf{R}(\mathbf{y}_i + \mathbf{V}_i\mathbf{q}) + \boldsymbol{\tau} \quad (1)$$

where $\mathbf{R}$ is the rotation matrix, s is a scalar scaling factor and $\boldsymbol{\tau}$ the translation vector where $\{s, \mathbf{R}, \boldsymbol{\tau}\}$ defines the global similarity transform. Principal component analysis (PCA) [6] was employed to obtain shape eigenvectors $\mathbf{V} = [\mathbf{V}_1^T, \dots, \mathbf{V}_N^T]^T$ that preserved 95% of the similarity normalized shape variation in the train set¹. The parameter vector $\mathbf{q}$ represents the non-rigid parameters described by the PDM where the submatrix $\mathbf{V}_i$ corresponds to the i th of N pre-defined landmarks on the face. We refer to the mean shape of the i th landmark as $\mathbf{y}_i$. We shall refer to all these rigid and non-rigid parameters of the PDM collectively as $\mathbf{p} = \{s, \mathbf{R}, \boldsymbol{\tau}, \mathbf{q}\}$ which is used in conjunction with the warping function $\mathcal{W}$. Procrustes analysis [3] is applied to all shape training observations in order remove all similarity before the PCA step. In our work we employed a $N = 68$ point landmark model. A final step in our PDM construction is the formation of a triangulated mesh, through Delaunay triangulation [6], from the 68 landmarks in the PDM. Examples of this mesh can be seen in Figure 1.

3. Measuring “Track-Life”

In this section we want to investigate the usefulness of interest point detectors like SIFT and SURF for non-rigid face tracking. To conduct this test we obtained 5 video sequences containing 5 different people captured with a variety of head movements and expressions. Each sequence was 180 frames long. In all 5 sequences we had each frame individually hand labeled to conform to a 68 point landmark model consistent with the PDM described in Section 2. To ascertain the usefulness of the SIFT and SURF features we conducted the following experiment.

1. Detect features in the source (i.e. at time index t) and destination (i.e., at time index $t + 1$) image frames (using either SIFT or SURF).
2. Remove feature points in the source image that do not lie within the hand labeled ground-truth face mesh.
3. Match interest points between the source and destination frames using the method described in Section 2 and remove the remaining spurious features.
4. Remove corresponding features that do not lie within the face mesh of the destination image based on the hand labeled ground-truth meshes.
5. Map the M source $\mathbf{x}_m$ and destination $\mathbf{x}'_m$ coordinates to $\mathbf{y}_m$ and $\mathbf{y}'_m$ respectively. The coordinates $\mathbf{y}_m$ and $\mathbf{y}'_m$ describe the position of the source and destination coordinates in the mesh shape PDM coordinate system. This is an important step as it allows for

¹The PDM employed in this paper was trained from the MultiPIE [7] dataset from over 1300 hand labeled shape examples containing variations in expression and pose.

Figure 2. A normalized histogram of feature track-lives using SIFT to estimate feature correspondence.

placing the source and destination landmarks into the same coordinate system for comparison. A barycentric combination strategy is employed with respect to the source and destination face meshes to solve for the mapping to the mean mesh. See Section 4 for a full description on the barycentric combination procedure.

6. The source $\mathbf{x}_m$ and destination $\mathbf{x}'_m$ points are deemed to be corresponding if $\|\mathbf{y}_m - \mathbf{y}'_m\|^2 \leq Th$. Setting $Th = 1$ gave good results through visual inspection of the tracked features on the video sequences.

The 68 point hand labeling of the video sequences was essential in these experiments to approximately model the non-rigid deformation of corresponding interest point detectors on the face.

Experiments: These steps were applied for each frame (with the exception of the final frame) of each video sequence. Statistics were obtained from the tracked sequences specifically: (i) the percentage of spurious SIFT = 23% and SURF = 26%, (ii) the average track-life SIFT = 3.8 and SURF = 3.4 frames, (iii) the maximum track-life SIFT = 45 and SURF = 42 frames. A normalized histogram of the track-lives for SIFT features can be seen in Figure 2.

The results of this experiment indicates that features on the face, using a generic salience criteria like SIFT and SURF, are only useful for tracking on “average” for approximately 3 – 4 frames. It is clear from these results that we should be attempting to detect new features at each frame rather than tracking existing features from previous frames as the number of useful features for tracking would decline exponentially over time.

4. Fitting Short-Track Life Features

Let us assume for convenience that our interest point detector gives us $M = N$ feature point correspondences where N is the number of landmarks points defining the PDM and M is the number of corresponding points between source and destination frames. Let us also assume that the source $\{\mathbf{x}_m\}_{m=1}^M$ and corresponding destination $\{\mathbf{x}'_m\}_{m=1}^M$ point sets correspond in position and ordering to the N points defined in the PDM. The problem of updating our

---

Input:- initial warp parameter vector $\mathbf{p}$ at time frame $t = 1$

1. Load the source I and destination I' images at time frames t and $t + 1$ respectively.
  2. Apply interest point detectors to the source and destination images.
  3. Remove any features in the source image that do not lie within the face mesh defined by $\mathbf{p}$.
  4. Apply nearest neighbor ratio matching to obtain corresponding source and destination coordinates.
  5. Estimate the warp update $\Delta\mathbf{p}$ by solving Equation 2.
  6. Update parameters: $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$, $t \leftarrow t + 1$.
  7. repeat steps 1-6 until $t - 2 > T$ where T is the total number of frames in the sequence.
-

Figure 3. Our proposed algorithm for non-rigid face tracking using *short track-life* features. Note that new features are re-estimated at every frame (most only lasting a few frames) which results in a varying number M of corresponding features at every frame t . In our proposed method no features are carried over between image frames, this is instead accomplished through the warp parameter vector $\mathbf{p}$ which has a fixed dimensionality across all frames.

current face warp parameter vector $\mathbf{p}$ becomes,

$$\arg \min_{\Delta\mathbf{p}} \sum_m^M w_m \|\mathcal{W}(\mathbf{y}_m; \mathbf{p} + \Delta\mathbf{p}) - \mathbf{x}'_m\|^2 \quad (2)$$

where we know $\{\mathbf{y}_m\}_{m=1}^M$ (the source points warped back to the mean face mesh), the warp vector of the source frame $\mathbf{p}$ and the corresponding destination points $\{\mathbf{x}'_m\}_{m=1}^M$. The weights w_m are employed to reflect our confidence in the correspondence between $\mathbf{x}_m$ and $\mathbf{x}'_m$. Our final goal then is to estimate the warp displacement $\Delta\mathbf{p}$ to minimize the error between the corresponding source and destination points. In practice this problem can be solved by taking the first order Taylor series approximation of the warp function with respect to the warp parameter $\mathbf{p}$.

$$\arg \min_{\Delta\mathbf{p}} \sum_m^M w_m \|\mathbf{x}_m + \frac{\partial \mathbf{x}_m}{\partial \mathbf{p}}^T \Delta\mathbf{p} - \mathbf{x}'_m\|^2 \quad (3)$$

Since the warp function defined in Equation 1 is not a linear function the solution to Equation 3 must be found iteratively (i.e. $\mathbf{p} \rightarrow \mathbf{p} + \Delta\mathbf{p}$) until convergence occurs. After convergence the final fit $\mathbf{p}$ becomes the initial warp in the next frame. An outline of our proposed fitting algorithm can be seen in Figure 3. We should note that the Jacobian matrix $\frac{\partial \mathbf{x}_m}{\partial \mathbf{p}}$ has to be recomputed at each iteration.

In practice, however, if we are going to detect new features at every frame the number of corresponding features M will vary with none of them corresponding to the vertex points defined in the PDM. A problem then exists in

how to obtain an $\frac{\partial \mathbf{x}_m}{\partial \mathbf{p}}$ that relates the warp function of the PDM with the feature point correspondences.

Barycentric Combination: Let us assume that given that we know a point $\mathbf{x}_m$ lies in triangle t in our current mesh. Triangle t is defined by three vertex points $\{\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k\}$ in the PDM mesh. These landmark points can be obtained from the current warp estimate $\mathbf{p}$. We can define $\mathbf{x}_m$ as a linear combination of these three vertex points,

$$\mathbf{x}_m = \mathbf{x}_i + \alpha_{i,m}(\mathbf{x}_i - \mathbf{x}_j) + \beta_{i,m}(\mathbf{x}_i - \mathbf{x}_k) \quad (4)$$

where the values $\alpha_{i,m}$ and $\beta_{i,m}$ can be solved as a linear system. Employing the partial derivative of $\mathbf{x}_m$ w.r.t. $\mathbf{x}_i$,

$$\begin{aligned} \frac{\partial \mathbf{x}_m}{\partial \mathbf{x}_i} &= \gamma_{i,m} \\ &= 1 + \alpha_{i,m} + \beta_{i,m} \end{aligned} \quad (5)$$

we can define,

$$\frac{\partial \mathbf{x}_m}{\partial \mathbf{p}} = \sum_{l=\{i,j,k\}} \gamma_{l,m} \frac{\partial \mathbf{x}_l}{\partial \mathbf{p}} \quad (6)$$

as the barycentric combination [6] of $\{\frac{\partial \mathbf{x}_l}{\partial \mathbf{p}}\}_{l=\{i,j,k\}}$. Now given M corresponding points between source and destination frames the warp update can be estimated through the application of Equation 3.

5. Related Methods

Lucas & Kanade: In literature there have been a number of efforts to apply the model-update strategy to non-rigid face tracking. Perhaps the most notable and effective can be found in AAM literature. Many instantiations of the AAM algorithm exist in literature. Perhaps the most elegant and simplest interpretation of the algorithm can be found in the work of Matthews and Baker [13] which simply poses an AAM as an extension of the classic Lucas & Kanade [12] (LK) algorithm,

$$\arg \min_{\Delta\mathbf{p}} \sum_m^M w_m \|I(\mathcal{W}(\mathbf{y}_m; \mathbf{p} + \Delta\mathbf{p})) - I'(\mathbf{x}'_m)\|^2 \quad (7)$$

where I and I' are the source and destination image frames and M now becomes the total number of pixels in the mean face mesh. In a similar step to the Taylor series expansion of Equation 2 we must linearize Equation 7 due to the non-linear relationship between the warping function and the image pixel intensities. In practice we end up with the following linear approximation,

$$\arg \min_{\Delta\mathbf{p}} = \sum_m^M w_m \|I'(\mathbf{x}'_m) + \frac{\partial I'(\mathbf{x}'_m)}{\partial \mathbf{p}}^T \Delta\mathbf{p} - I(\mathbf{x}_m)\|^2 \quad (8)$$

which is iteratively solved $\mathbf{p} \rightarrow \mathbf{p} + \Delta\mathbf{p}$ until convergence occurs. This procedure can be quite computationally

expensive as it requires the re-estimation of the Jacobian matrix $\frac{\partial I'(\mathbf{x}_m)}{\partial \mathbf{p}}$ at each iteration. Additional extensions to LK algorithm have been suggested in terms of computation (the “inverse compositional” (IC) algorithm [13]) and an additional appearance constraint (the “simultaneous inverse compositional” (SIC) algorithm [13]), however, both these extensions are outside the scope of this paper. Even taking these extensions into account, since the source image is updated after every frame $I \rightarrow I'$ the Jacobian must be re-estimated at least every frame.

Robust Error Function: In our experiments we want to compare our proposed algorithm (using both SIFT and SURF) with the AAM inspired LK algorithm. Both algorithms being compared require an iterative solution and have the ability to place a weight w_m on each corresponding feature/pixel as illustrated in Equations 3 and 7 respectively. To handle noisy/outlier correspondences and pixels a robust error function [18] was employed for both methods instead of straight unweighted least-squares. It has been demonstrated for AAM fitting [18] and for 3D rigid face tracking using SIFT features [9] that a decaying exponential $\eta(e) = \frac{1}{c} \exp(-c \cdot e)$ gives good performance given that c is suitably chosen and e is the reconstruction error. This results in the following weight update after each iteration,

$$w_m \leftarrow w_m \exp(-c \cdot e) \quad (9)$$

of both Equations 3 and 7. Through a cross-validation procedure we found a $c = 0.1$ to perform well for both algorithms.

Experimental Comparison: In Figure 4 one can see the comparison, in terms of tracking error, of our proposed algorithm using the SIFT and SURF interest point detectors as well as the LK algorithm on a first frame initialized video sequence. It is interesting to note that the SIFT and SURF features give similar performance to one another. Based on this result we have elected to use the SURF interest point detector throughout the rest of the experiments in this paper due to its computational efficiency. In this experiment the LK algorithm was expected to perform best due to the continuous nature of the facial shape and appearance changes in the video sequence (see top row of Figure 7 for an example keyframes from the sequence). The LK algorithm, however, only outperformed both versions of our algorithm by a slight margin. All three techniques are equally effected by drift over time.

Synthesizing Jitter: The tracking results in Figure 4 do not really showcase the advantages of our proposed approach. It is rare in “real-world” video sequences for a person, or the camera, to stay still as the human face non-rigidly deforms. In Figure 5 we demonstrate the advantage of our algorithm by introducing synthetic jitter to the same video

Figure 4. Comparison of the tracking results for the LK algorithm with two variants of our proposed algorithm based on the interest point detectors SIFT and SURF. Inspecting these results one can see that there is minimal difference in performance between the SIFT and SURF versions of our proposed algorithm. The AAM inspired LK algorithm slightly outperforms our proposed algorithm on this stable face sequence. All three algorithms are effected by drift over time.

sequence. The jitter was introduced by randomly shifting in translation each image frame in both the x - and y -directions by a maximum amount from the ground-truth position bounded by the variable **MAG** (so in theory the maximum displacement between frames could be $2\mathbf{MAG}$ in both the x - and y - directions). Inspecting Figure 5(a) one can see that the LK algorithm fails in the presence of slight ($\mathbf{MAG} > 1$ pixel) jitter. Alternately our proposed algorithm in Figure 5(b) sees almost no change in tracking performance irrespective of the amount of synthetic jitter introduced. One could argue that the experiment in Figure 5 is unfair as the ability for the LK algorithm to handle larger displacements can be enhanced through a coarse-to-fine fitting strategy. However, even taking these enhancements into account the LK algorithm would not be able to handle displacements in the order of 25 – 50 pixels as depicted in (b).

Another drawback of the LK algorithm in comparison to our own is computational speed. The LK algorithm requires the re-estimation of the large image intensity Jacobian matrix at each iteration. Even with the application of the inverse-composition algorithm [13] the technique still requires the re-estimation of the Jacobian matrix at each frame. The speed of our proposed algorithm, using the computationally efficient SURF interest point detector, allows for the processing of frames in close to real-time. To gain an indication of these speed differences, in our MATLAB code (which was not optimized for speed) we were able to process at 3 sec/frame using the LK algorithm and 0.1 seconds/frame using our algorithm with SURF² features.

6. Drift Rectification

Irrespective of the model-update scheme employed drift will still always occur, ending up (eventually) with divergence from the face. Inspired by similar work being conducted in 3D rigid face tracking [9, 16] we have explored

²Our SURF interest point detector is derived from the latest release in OpenCV.

Figure 5. In this figure we depict the effect of synthesized camera jitter on the tracking process. The value **MAG** refers to the absolute maximum translational shift from ground-truth the image was perturbed by in both the x - and y - directions. It is interesting to note that the LK algorithm in (a) is susceptible to slight ($\text{MAG} > 1$ pixels) amounts of jitter, whereas our proposed algorithm using the SURF interest point detector sees barely any drop in performance for values as large as **MAG** 25 (pixels). The dramatic difference in performance can be attributed to the method used by our algorithm to track features over time.

strategies to rectify the amount of drift occurring in our non-rigid face tracker and recover when divergence occurs. Our strategy is not about stopping drift, it is simply about minimizing that drift. Imagine we had a face sequence of only 2 frames that were being infinitely replayed over time. Applying our current model-update strategy to this sequence we would eventually end up with divergence due to the culmination of tracking errors. However, the amount of information being carried in that sequence is only 2 frames irrespective of how long it plays. Specifically, drift should be related to the amount of new information the tracker observes *not* the number of frames.

Inspired by this insight our strategy is to create an adaptive database ($\mathcal{DB}$) of templates which the tracker can call on so as to make drift a function of new information observed in the video sequence rather than the number of frames. The templates in the $\mathcal{DB}$ should be representative of the images it has previously seen in the face. We employ the feature descriptors obtained from SURF for this purpose. This strategy, however, has two problems: (i) how do we adaptively update the $\mathcal{DB}$?, and (ii) how do we combine the $\mathcal{DB}$ within our current tracking process?

Updating the database: The first frame is used to initialize the $\mathcal{DB}$ with its first template. The $\mathcal{DB}$ is actually a set of indexes $j \in \mathcal{DB}$ to previous frames that have been tracked. For each of these stored indexes the interest point detector feature locations and descriptors for that frame, as well as the warp estimate $\mathbf{p}_j$ and covariance Σ_j at that frame are stored. We refer collectively to all these values from frame j as template j . The importance of the covariance Σ_j shall be expanded upon in the next section. In our current system we employ the maximum number of feature matches between each template in the current $\mathcal{DB}$ as an indicator of how much new information the current frame contains. If the number of matches falls below a threshold then that image frame’s descriptors are stored as an additional template in the $\mathcal{DB}$.

Tracking with the database: The $\mathcal{DB}$ is applied to the tracking process through a Kalman filter framework [17].

We define the estimate of the warp vector $\mathbf{p}_{t|k}$ at time frame t given the previous frame k as,

$$\mathbf{p}_{t|k} = \mathbf{p}_k + \Delta\mathbf{p}_{k,t} + \epsilon_{k,t} \quad (10)$$

where $\mathbf{p}_k$ is the parametric warp vector at time frame k . We refer to $\Delta\mathbf{p}_{k,t}$ as the additive update (from Equation 2) between frames k and t , and $\epsilon_{k,t}$ is the uncertainty associated with that update. The uncertainty is assumed to be modeled by,

$$\epsilon_{k,t} \sim \mathcal{N}(\mathbf{0}; \Sigma_{k,t}) \quad (11)$$

where $\Sigma_{k,t}$ is the covariance of the update estimate. Further information of suitable methods for estimating $\Sigma_{k,t}$ can be found in [17]. Equation 10 can double as the state and observation equations for the Kalman filter where k is equal to $t - 1$ and j respectively (assuming independence). The index j refers to the template in the $\mathcal{DB}$ with the smallest $\|\Sigma_{j,t}\|$. The outputs of the Kalman filter [17] are then $\mathbf{p}_t$ and Σ_t which is the covariance of that current estimate. In some instances a template in the $\mathcal{DB}$ can be updated using Σ_t . The current template t replaces template j from the $\mathcal{DB}$ if the covariances satisfy the following equation,

$$\|\Sigma_t\| < \|\Sigma_j\| \quad (12)$$

Drift Experiments: A comparison of our model-update method with (drift-rectified) and without (accumulated) the drift rectification procedure can be seen in Figure 6 for two video sequences with the first frames initialized. In (a) one can see some improvement in performance over time with the drift rectification procedure, however, in (b) the benefit is far more pronounced. The reason behind the discrepancy can be found in the nature of the two video sequences employed. Sequence 1 was a shorter sequence (180 frames) and did not contain too much repetition which indicates high information content. Sequence 2 was a longer sequence (241 frames), however, there was a lot of repetition indicating lower information content and thus less drift when compared to the accumulated result.

Occlusion Experiments: The effect of occlusion was evaluated with our proposed system with drift rectification. In

Figure 6. In this figure we depict a comparison between our proposed drift rectification procedure on two video sequences. The first video sequence (a) is shorted and does not have much repetition explaining why there is only a marginal performance benefit between the drift-rectified and accumulated results. The second video sequence (b) is longer and contains more repetition explaining why there is a substantial performance improvement between the drift-rectified and accumulated results.

Figure 7 one can see keyframes from the tracking result of the same video sequence with three different synthetically induced occlusion states: (a) no occlusion, (b) small occlusion (20% of the face occluded), and (c) large occlusion (60% of the face occluded). Our system performed well across all three sequences with near perfect performance for both (a) and (b). Even in the presence of substantial occlusion (c) the system is able to snap back in subsequent frames.

“Real World” Experiments: In our final experiment we tested our system on a “real-world” video sequence. The sequence was a short excerpt (896 frames) taken from the British comedy series “The Office” where the first frame was initialized. The subject’s face is challenging due to the facial hair, changing camera view, occlusion, extreme expressions and changes in head pose. As one can see from the keyframes in Figure 8 our system is able to cope robustly with all these variations and when divergence does occur, see (e), the system snaps back on subsequent frames.

7. Conclusions

In this paper we investigated the utility of interest point detectors like SIFT and SURF for generic non-rigid face tracking. We demonstrated that these features have a short “track-life” and should therefore employ a strategy of detecting new features at every frame. A novel non-rigid model-update algorithm was proposed based on these features which demonstrated remarkable invariance to discontinuous variations in motion. An additional drift rectification step was proposed that demonstrated robust performance in the presence of occlusion and error recovery.

References

- [1] H. Bay, T. Tuytelaars, and L. V. Gool. SURF: Speeded up robust features. In *European Conference on Computer Vision (ECCV)*, volume 3951, pages 404–417, 2006.
- [2] T. Cootes, D. Cooper, C. J. Taylor, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, January 1995.
- [3] T. F. Cootes and G. J. Edwards. Active appearance models (AAMs). *IEEE Trans. PAMI*, 23(6):681–685, June 2001.
- [4] D. Cristinacce and T. F. Cootes. Feature detection and tracking with constrained local models. In *British Machine Vision Conference*, pages 929–938, 2006.
- [5] P. Felzenszwalb and D. Huttenlocher. Efficient matching of pictorial structures. In *IEEE International Conference on Computer Vision and Pattern Recognition*, June 2000.
- [6] D. A. Forsyth and J. Ponce. *Computer Vision*. Prentice Hall, 2003.
- [7] R. Gross, J. S. Baker, I. Matthews, and T. Kanade. Multi-PIE. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2008.
- [8] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, 1988.
- [9] J. Jang and T. Kanade. Robust 3D head tracking by online feature registration. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2008.
- [10] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, 1998.
- [11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.
- [12] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.
- [13] I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135 – 164, November 2004. In Press.
- [14] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. PAMI*, 27:1615–1630, 2005.
- [15] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(7):43 – 72, November 2005.
- [16] L. Morency, J. Whitehill, and J. Movellan. Generalized adaptive view-based appearance model: Integrated framework for monocular head pose estimation. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2008.
- [17] G. Strang. *Introduction to Applied Mathematics*. Wellesley-Cambridge Press, 1986.
- [18] B. Theobald. Evaluating error functions for robust active appearance models. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 149–154, 2006.
- [19] Y. Wang and S. Lucey. Enforcing convexity for improved alignment with constrained local models. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.
- [20] Y. Zhou, L. Gu, and H. Zhang. Bayesian tangent shape model: Estimating shape and pose parameters via Bayesian inference. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 109–116, 2003.

Figure 7. Keyframes taken from a face sequence in the presence of: (a) no occlusion, (b) small synthetic occlusion (covering 20% of the face), and (c) large synthetic occlusion (covering 60% of the face). Our system successfully tracks the face with minimal drift in both (a) and (b). In (c) even when the system starts to diverge it then recovers.

Figure 8. Keyframes taken from a clip from the British comedy series “The Office” where we have initialized the first frame and let our non-rigid face tracker run. The keyframes in the sequence can be seen from (a) - (f). One can see our system handles changes in scale (b) and occlusion (c,d). When the system does fail (e) our system snaps back (f).