A Direct Method for Modeling Non-rigid Motion with Thin Plate Spline

Jongwoo Lim
Department of Computer Science
University of Illinois at Urbana Champaign
Urbana, IL 61801
jlim1@uiuc.edu

Ming-Hsuan Yang
Honda Research Institute
800 California Street
Mountain View, CA 94041
myang@honda-ri.com

Abstract

Thin plate spline (TPS) transformations have been applied to non-rigid shape matching with impressive results. However, existing methods often use a sparse set of point correspondences which are established prior to shape matching. A straightforward approach to finding point correspondences and computing TPS parameters imposes expensive computations, thereby motivating us to develop an efficient solution. In this paper, we present a direct method for recovering non-rigid object motion from its appearance in which the point correspondences are simultaneously established while estimating TPS parameters. The motion parameters are estimated in a stiff-to-flexible approach and the principal appearance deformations are learned that can be utilized for motion analysis and recognition. Numerous experiments demonstrate the efficiency and efficacy of the proposed algorithm in modeling the motion details of non-rigid objects undergoing shape deformation and pose variation.

1. Introduction

One main challenge confronting the vision community is how to effectively model object appearance of which the variation can be attributed to rigid or non-rigid motion. Numerous methods have been developed to handle rigid object motion in the past, and recently more algorithms have been proposed for the case of non-rigid objects. Evidently, affine transformations have long been applied to model global rigid object motion successfully. However, it is also well known that affine transformation is not effective when an object undergoes non-rigid motion. Non-rigid object motions, such as facial expressions and movements of human bodies, often carry important information for vision applications. Thus it is imperative to develop an effective model for non-rigid deformation and an efficient algorithm for recovering its parameters directly from images.

Thin plate spline (TPS) warp has been widely used in image alignment and shape matching. Given a set of n corresponding 2D points, the TPS warp is described by $2(n+3)$ parameters which include 6 global affine motion parameters and $2n$ coefficients for correspondences of the control points. These parameters are computed by solving a linear system.

In this paper, we propose a direct method for estimating the TPS warp between images based on the appearance of the target object. In contrast to existing works that make use of a sparse set of control point correspondences, our algorithm does not require any detection of feature points in images, nor does it need to establish correspondences of those feature points. The control points in our algorithm are positioned rather arbitrarily (e.g., points on a grid), and need not to be on image features or boundaries. Instead of first establishing correspondences and then solving the linear system, we estimate the warp parameters by solving an optimization problem with gradient descent techniques. Furthermore, we take a stiff-to-flexible approach in order to better estimate non-rigid motion. Borne out by numerous experiments, we show that the proposed algorithm is able to model non-rigid object motion undergoing large shape deformation and pose variation.

The rest of this paper is organized as follows. We first discuss the most relevant works on modeling non-rigid object motion. Next we present the application of thin plate spline for modeling non-rigid object motion in images, and provide a computationally efficient solution to estimate the warp parameters. Empirical results are presented in Section 4, followed by discussions on the potency and limitations of the proposed algorithm. Finally, we conclude this paper with comments on future work.

2. Prior Work

There is a rich literature on modeling non-rigid object motion from visual cues ranging from contour, appearance, to optical flow. Contour-based methods mainly use edge information near or on the boundary of the target object,

and aim to find a transformation so that the object shapes are best matched by minimizing an energy function. Algorithms such as snakes [13], active shape model [6], active contour with Condensation [12], and geodesic contours [17] have demonstrated their capabilities of modeling non-rigid objects. These methods differ in the shape representation including energy functions defined in terms of edges and curvature [13], statistical distributions of points on contour [6], curvature normals [12], and energy functions defined by level sets [17]. Gradient descent algorithms are then applied to minimize energy functions [13, 17] or to fit model to the image [6]. Alternatively, factored sampling based methods can be applied to best match image observation to the model [12]. One key weakness shared by the above-mentioned methods is that initialization of points or edges on or near the object contour is imperative to ensure their success. Furthermore, these methods utilize only the contour of an object for matching and ignore the abundant and rich texture information at one's disposal.

Appearance-based methods utilize the texture information of an object for estimating non-rigid deformation. Existing algorithms resort to deformable templates with fiducial points [10], local parametric image patches [1], texture within triangulated mesh [18], elastic bunch graph matching [21] or a combination of shape and texture [5] [8]. The best image warp is obtained by minimizing the sum of square difference of pixel values between the template and an observed image, where affine transformation [10, 18], graph matching [21] or eight-parameter projective model [1] is employed. Applications such as facial expression recognition have shown great promise of this approach [14]. However the templates are not updated over time and thus cause problems when the imaging conditions (such as lighting, view angles) differ significantly from that of the stored template. Furthermore, fiducial points [10, 21, 7] or local patches [1, 18] need to be manually labeled prior to shape matching.

In [20] Weiss developed an EM-based motion segmentation method by fitting a mixture of smooth flow fields to the spatio-temporal image data. By exploring low rank constraints [11] of the optic flow matrix, Bregler et al. [3] proposed an algorithm for non-rigid object tracking. Although these methods have their success in modeling non-rigid motion with feature points, optical flow estimation is usually sensitive to illumination change, occlusion, and noise, thereby limiting the potency of these methods for non-rigid object tracking.

The use of thin plate spline warp for mapping points between two frames given their correspondence was first advocated by Bookstein [2]. One attractive feature of TPS formulation is that it consists of affine and non-affine warping transformations, thereby allowing it to capture the global rigid and local non-rigid motions. Consequently, TPS has been widely applied to shape matching including medical imaging with impressive results on estimating non-rigid motion. Chui and Rangarajan [4] presented a method that simultaneously estimates the correspondence of points and non-rigid TPS warps by resorting to the EM algorithm. Although their experiments demonstrate excellent results in point matching, it not clear whether this algorithm can be extended to use texture information for modeling non-rigid motion.

3. Algorithm

We first briefly review the TPS warp and then present an efficient algorithm that computes correspondence as well as TPS parameters simultaneously. We also propose a stiff-to-flexible approach in order to better model non-rigid deformations.

3.1 Thin Plate Spline Warp

Given n control points in a plane $(\hat{x}_i, \hat{y}_i) \in I\!R^2$ and their corresponding function values $\hat{v}_i \in I\!R$, $i = 1, \ldots, n$, the thin plate spline interpolation $f(x, y)$ specifies a mapping $f : I\!R^2 \rightarrow I\!R$ whose bending energy E_f is minimal,

$$E_f \;=\; \int\int_{I\!R^2} (f_{xx}^2 + 2f_{xy}^2 + f_{yy}^2) dxdy$$

and the interpolated value at a point (x, y) is given by

$$f(x,y) \;=\; a_1 + a_2 x + a_3 y + \sum_{i=1}^{n} w_i U\left(||(\hat{x}_i, \hat{y}_i) - (x, y)||\right)$$

where $U(r) = r^2 \log r^2$. Note that the interpolated spline function consists of two parts: affine transformation parameterized by $\mathbf{a}$ and non-affine warping specified by $\mathbf{w}$.

Since the spline $f(x, y)$ needs to have square-integrable second derivatives, we have the following constraints

$$\sum_{i=1}^{n} w_i \;=\; 0, \text{ and } \quad \sum_{i=1}^{n} w_i \hat{x}_i \;=\; \sum_{i=1}^{n} w_i \hat{y}_i \;=\; 0.$$

The TPS parameters $\mathbf{a}$ and $\mathbf{w}$ can be computed by solving the following linear equation:

$$\begin{bmatrix} A & P \\ P^\top & O \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{a} \end{bmatrix} \;=\; \begin{bmatrix} \mathbf{v} \\ \mathbf{0} \end{bmatrix} \tag{1}$$

where $A_{ij} = U(||(\hat{x}_i, \hat{y}_i) - (\hat{x}_j, \hat{y}_j)||)$, the i-th row of P is $(1, \hat{x}_i, \hat{y}_i)$, O and $\mathbf{0}$ are 3×3 and 3×1 zero matrices, and $\mathbf{w}$, $\mathbf{a}$ and $\mathbf{v}$ are vectors formed from w_i, a_1, a_2, a_3 and $\hat{v}_i$, respectively. We denote the leftmost $(n+3) \times (n+3)$ matrix in the linear system as K. When noise exists in the target value $\hat{v}_i$, we need to introduce the regularization parameter

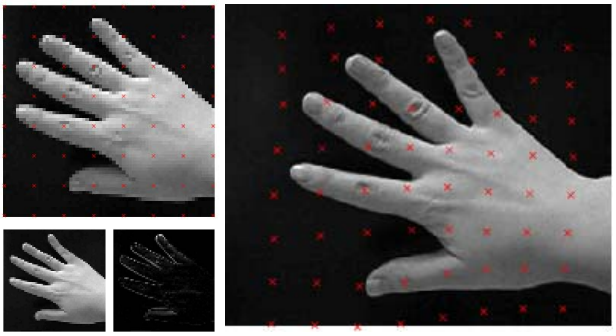


Figure 1. Modeling non-rigid motion with thin plate spline. (Upper Left) a template image with control points (red $\times$). (Right) a target image overlaid with warped control points. (Lower Left) the back-warped image of the target region, and its matching residue to the template image.

λ to control the amount of smoothing in TPS. That is, the submatrix A in Equation 1 is replaced by $A + \lambda \mathbf{I}$, where $\mathbf{I}$ is an $n \times n$ identity matrix.

In this paper, we are interested in warping 2D points using TPS defined by pairs of control points, i.e., we want to map points (x, y) to (x', y') given n control point correspondences $(\hat{x}_i, \hat{y}_i)$:$(\hat{x}'_i, \hat{y}'_i)$. Toward that end, we need two TPS functions for x and y coordinate separately. From Equation 1, the TPS warp which maps $(\hat{x}_i, \hat{y}_i)$ onto $(\hat{x}'_i, \hat{y}'_i)$ can be recovered by

$$\begin{bmatrix} \mathbf{w}_x & \mathbf{w}_y \\ \mathbf{a}_x & \mathbf{a}_y \end{bmatrix} = K^{-1} \begin{bmatrix} \hat{\mathbf{x}}' & \hat{\mathbf{y}}' \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{2}$$

where $\hat{\mathbf{x}}'$ and $\hat{\mathbf{y}}'$ are the vectors formed with $\hat{x}'_i$ and $\hat{y}'_i$ respectively. The transformed coordinates (x'_j, y'_j) of points (x_j, y_j) are given by

$$\begin{bmatrix} \mathbf{x}' & \mathbf{y}' \end{bmatrix} = \begin{bmatrix} B & Q \end{bmatrix} \begin{bmatrix} \mathbf{w}_x & \mathbf{w}_y \\ \mathbf{a}_x & \mathbf{a}_y \end{bmatrix} \tag{3}$$

where $B_{ji} = U(||(x_j, y_j) - (\hat{x}_i, \hat{y}_i)||)$, the j-th row of Q is $(1, x_j, y_j)$, and the j-th row of the resulting vectors $\mathbf{x}'$ and $\mathbf{y}'$ are the interpolated x and y coordinates x'_j and y'_j, respectively. We denote the matrix $[B \;\; Q]$ as M.

3.2 Efficient TPS Warp Estimation

Most of prior works follow the procedure described in the previous section. Namely given the correspondence of control points, the TPS coefficients are first estimated and then the points of interest are transformed to new locations using the interpolated spline.

We consider the problem of modeling non-rigid deformations in images based on object appearance. Assume that we are given a template of an object with control points and a target image of the same object undergoing non-rigid motion, the goal is to find the TPS warp which deforms the template so that it aligns well with the object in the target image. In the first frame, the target region is initialized either manually or by some object detector, and then the non-rigid motions are estimated throughout the sequence. The control points are only used to determine the amount of non-rigid deformation, thus thy do not need to lie on specific feature points. In this paper, we position control points on a regular grid as illustrated in Figure 1.

The template is assumed to be fixed throughout the image sequence, and the control points in the template are also fixed. To find the best TPS warp for each frame, we build the back-warped image by warping each pixel in the template image and finding the corresponding pixel value by linear interpolation of neighboring pixels in that frame. Then we compare the template and the back-warped image to update the TPS parameter (the detailed algorithm will be explained later in this section). Combining equations 3 and 2, we have the following equation that directly relates the warped positions of pixels with the positions of control points:

$$\begin{bmatrix} \mathbf{x}' & \mathbf{y}' \end{bmatrix} = MK^{-1} \begin{bmatrix} \hat{\mathbf{x}}' & \hat{\mathbf{y}}' \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{4}$$

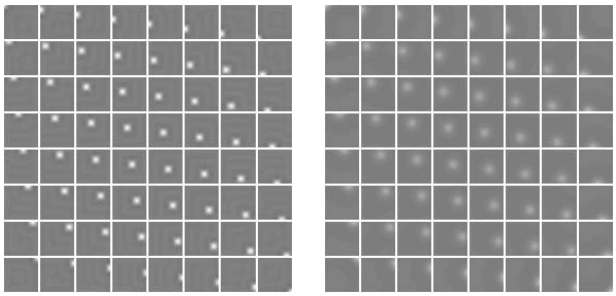
Note that in the above equation, the transformed coordinate (x'_i, y'_i) can be expressed in terms of a linear combination of the coordinates of the control points in the target image $(\hat{x}'_i, \hat{y}'_i)$. More importantly, the left two matrices (M and K) remain constant if the control points $(\hat{x}_i, \hat{y}_i)$ and the set of points (x_j, y_j) in the template remain constant. Inverting a large matrix K is computationally intensive. In our approach, we pre-compute MK^{-1} during the initialization stage, and the subsequent warps can be computed very efficiently.

Let the template image be I_0 and the target image be I, we model the non-rigid motion estimation problem as the minimization problem of the following error function

$$E(\boldsymbol{\mu}) = \sum_{\mathbf{p}=(x,y)} ||I(w(\mathbf{p}; \boldsymbol{\mu})) - I_0(\mathbf{p})||^2 \tag{5}$$

where $w(\mathbf{p}; \boldsymbol{\mu})$ is a TPS warp with parameter $\boldsymbol{\mu}$. Note that the TPS kernel matrices M and K remain constant during the estimation process due to the constancy assumption of the template and the control points. Thus the TPS warp is solely determined by the coordinates of control points in the target frame ($\hat{x}'$ and $\hat{y}'$). As shown in Equation 4, we let the TPS parameter vector $\boldsymbol{\mu}$ as $(\hat{\mathbf{x}}'^{\top}, \hat{\mathbf{y}}'^{\top})^{\top}$ and denote the top $(n+3) \times n$ submatrix of K^{-1} as $K^{-1}_{(n+3)\times n}$. Then we can write the warped coordinate of a point $\mathbf{p} = (x, y)$ in the template image as

$$w(\mathbf{p}; \boldsymbol{\mu}) = \left[\; M_{\mathbf{p}} K^{-1}_{(n+3)\times n} \;\middle|\; M_{\mathbf{p}} K^{-1}_{(n+3)\times n} \; \right] \boldsymbol{\mu}$$



(a) (b)

Figure 2. Support of pixels in the template image to the 8×8 control points. Each patch represents how much each pixel in the template image contributes to the update of the control point. (a) From the TPS kernel with a small regularization parameter. (b) From the TPS kernel with a large regularization parameter. As the regularization increases, the support spreads out and more pixels affects the update of each control point.

where $M_{\mathbf{p}}$ is the row corresponding to $\mathbf{p}$ in the precomputed matrix M.

Motivated by the direct methods for estimating affine parameter using object appearance [9, 19], we apply gradient descent techniques to find the parameter which minimizes Equation 5. Using Taylor series expansion, the warped image $I(w(\mathbf{p};\boldsymbol{\mu}))$ can be expressed as $I(w(\mathbf{p};\boldsymbol{\mu}_0+\delta\boldsymbol{\mu}))$:

$$I(w(\mathbf{p};\boldsymbol{\mu}_0+\delta\boldsymbol{\mu})) \;=\; I(w(\mathbf{p};\boldsymbol{\mu}_0)) + \delta\boldsymbol{\mu}^\top \nabla I(w(\mathbf{p};\boldsymbol{\mu}_0)) \;+\; \text{h.o.t} \tag{6}$$

where $\nabla I(\cdot) \;=\; \left(\frac{\partial}{\partial \boldsymbol{\mu}_1} I(\cdot) \,\middle|\, \cdots \,\middle|\, \frac{\partial}{\partial \boldsymbol{\mu}_k} I(\cdot)\right)^\top$. Each term $\frac{\partial}{\partial \boldsymbol{\mu}_1} I(\cdot)$ can be computed using the gradient image of I. Plugging Equation 6 into the Equation 5 and ignoring the higher-order-terms, the error function can be rewritten in terms of $\delta\boldsymbol{\mu}$,

$$E(\delta\boldsymbol{\mu}) = \sum_{\mathbf{p}} ||I(w(\mathbf{p};\boldsymbol{\mu}_0)) - I_0(\mathbf{p}) + \nabla I(w(\mathbf{p};\boldsymbol{\mu}_0))^\top \delta\boldsymbol{\mu}||^2,$$

and $\delta\boldsymbol{\mu}$ which decreases the error function can be computed by

$$\delta\boldsymbol{\mu} \;=\; (W^\top W)^{-1} W^\top I_t$$

where $W_{ji} \;=\; \frac{\partial}{\partial \boldsymbol{\mu}_i} I(w(\mathbf{p}_j;\boldsymbol{\mu}_0))$ and I_t is the vector with $I(w(\mathbf{p}_j;\boldsymbol{\mu}_0)) - I_0(\mathbf{p}_j)$ as its j-th row value. Figure 2 shows how the image gradient and the temporal derivative at each pixel in the template contribute to the update of the control point.

3.3 Stiff-to-Flexible Approach

Thin plate spline warp offers great flexibility in shape matching, and thus care needs to be taken when some control points lie in textureless regions or the initial TPS parameters are not close to the desired ones. Akin to the coarse-to-fine strategy which is widely used in rigid motion estimation algorithms, we propose a stiff-to-flexible approach.

When the regularization parameter λ becomes larger, the estimated warp becomes more "rigid". This can be shown easily with the patterns of support at each control point as shown in Figure 2. Smaller λ makes the region of interest at each control point smaller (Figure 2 (a)), and thus each control point can move more freely without being affected by appearance changes far from it.

When a new frame arrives, the initial TPS parameters are set to the ones obtained in the previous frame. In general, the initialized parameters are expected to be close to the true ones since the interframe motion in a video is usually small. However a simple and straightforward application of our algorithm may not find the correct estimate due to several reasons. Control points regions without much texture do not contain sufficient visual information for correct convergence. Also the classic aperture problem may arise since when the object moves as, depending on their texture, some control points may not move along with others. Hence we start with some large λ_0 and update all control points to follow the global motion by enforcing the warp to be more rigid, and then repeat with smaller λ_k's to capture local non-rigid deformations until we have a good estimate.

4. Experiments

We apply the proposed algorithm to model non-rigid motion in videos with large shape deformation and pose variation. Some empirical results are presented in this section and more can be found in the accompanied videos at `http://vision.ucsd.edu/~jwlim/tps/`.

We first illustrate that our algorithm is able to handle the articulated hand motions. Figure 3 shows how the TPS control points area estimated in order to account for non-rigid hand motion. In this video, a hand is moving with fingers stretched outward and inward with a dark background. One may think that simple background facilitates this motion estimation problem easy. However, the simple background also introduces stability problem as many control points are located in the background. Since there is little information near the extents of control points located in the background, such points can be easily affected by image noise. Nevertheless with the our stiff-to-flexible approach, we are able to maintain global arrangement of the control point correctly throughout the sequence.

In our experiments, the template is initialized with a neutral hand image and a 8×8 grid of 64 control points is employed to model hand motion. There is no strict restriction on how the control points are laid out on the grid, however a uniform grid is preferable since the support for each control point becomes similar to each other. Figure 2 shows

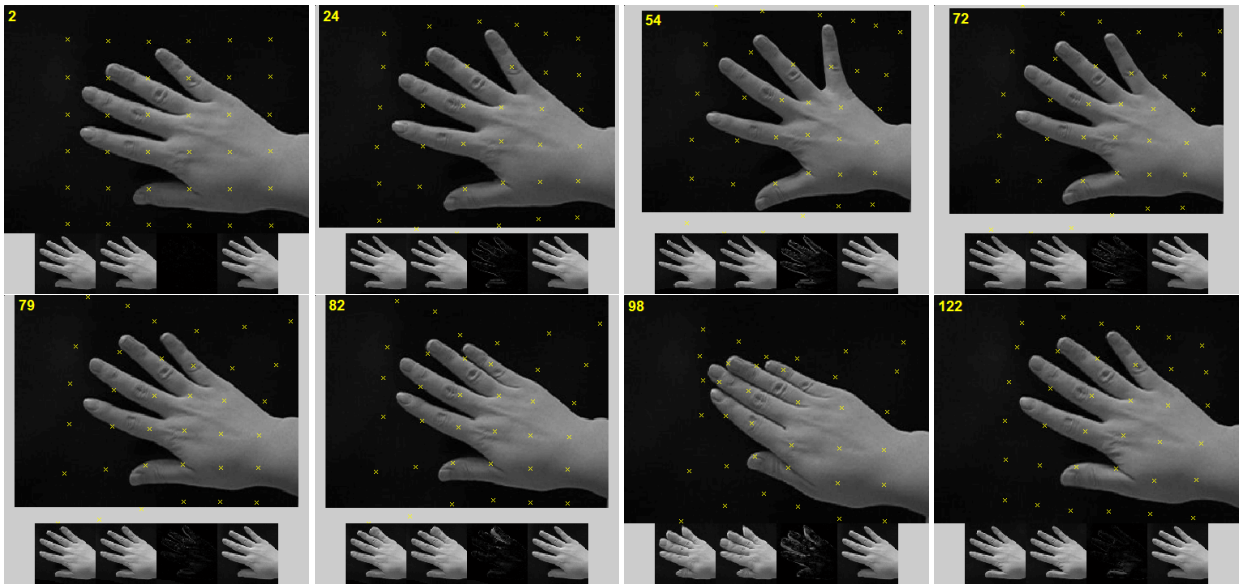


Figure 3. Articulated hand motion. The warped images using our stiff-to-flexible TPS approach are shown from left to right on the bottom row, followed by the mismatch error of the final warped image, and the template on the far right.

how the support for each control points looks like. Note also that there is a scarce of texture information and thus it is difficult for to extract features or establish feature point correspondences between frames. Consequently algorithms using features will not model such hand motion well.

The second test video contains a tiger strolling around as Figure 4 shows the proposed algorithm is able to account for the non-rigid object motion in the torso region of this tiger. Like the articulated hand motion, the torso region deforms non-rigidly as the tiger strolls, thereby easily defeating methods that only models rigid motion. As presented in this figure, the proposed algorithm models the motion details of the torso region accurately as the warped image regions match the templates well. Note that our algorithm does not use any specific feature points to model non-rigid motion, but rather minimizes the mismatch error between the warped image as well as the template. Nevertheless, the proposed algorithm fails to model the motion well after the 40th frame due to motion blur and abrupt camera motion.

The third video contains a male who changes his facial expression and pose throughout this video. The motion details and shape deformation cannot be modeled well with affine transformation alone. Nevertheless, our algorithm is able to faithfully account for motion details as shown in Figure 5. In the figure, the warped images from stiff-to-flexible are shown from left to right, followed by mismatch reside of the final warped image and the temple shown on the far right. The results show that our algorithm is able to model motion caused by facial expression and pose variation. The bottom row shows the basis for the TPS warping, thereby enabling us to analyze facial motions.

5. Concluding Remarks

We have described a direct method for estimating non-rigid object motion from its appearance without resorting to sparse point correspondences. Explicitly establishing point correspondences may not be robust when there are only few distinguishable feature points in images, and estimating the TPS coefficients in every video frame is computationally expensive. We develop an efficient algorithm to compute non-rigid warps in a video sequence when the template and the control points are fixed. The proposed stiff-to-flexible approach enables us to estimate the TPS parameters robustly.

Our future work will focus on applications of the proposed method for motion analysis and recognition. We will also extend our visual tracking algorithms [15, 16] to handle non-rigid object motion with the presented algorithm.

Acknowledgments

We thank the anonymous reviewers, Jeffrey Ho as well as David Kriegman for their comments, and Lorenzo Torresani for providing the strolling tiger video. Jongwoo Lim is supported partially under the grant NSF CCR 00-86094 from the National Science Foundation.

References

[1] M. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models

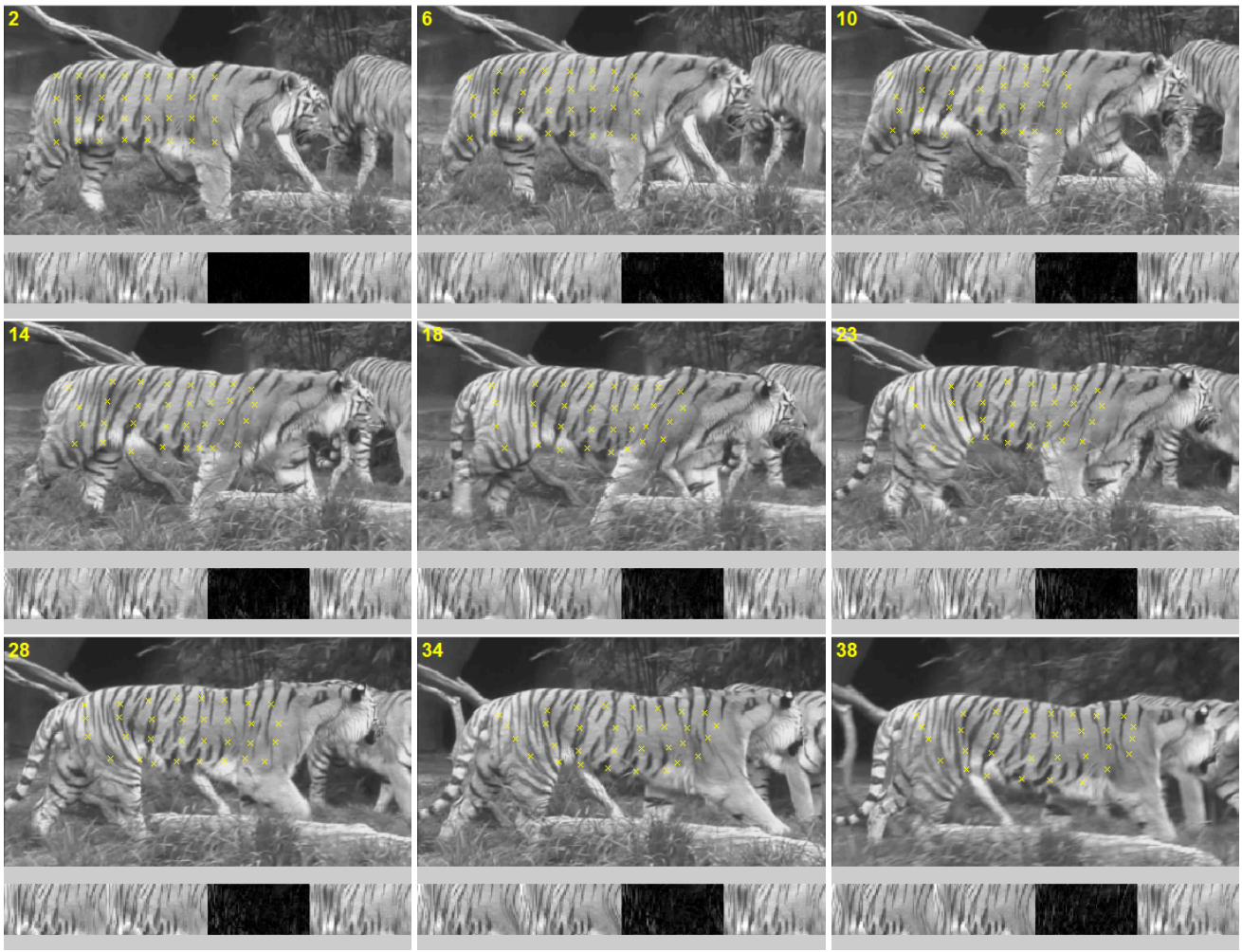


Figure 4. Strolling tiger. The warped images using our stiff-to-flexible TPS approach are shown from left to right on the bottom row, followed by the mismatch error of the final warped image, and the template on the far right.

of image motion. In *Proceedings of the Fifth IEEE International Conference on Computer Vision*, pages 374–381, 1995.

[2] F. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, 1989.

[3] C. Bregler, A. Hertzmann, and H. Biermann. Recovering non-rigid 3D shape from image streams. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 2690–2696, 2000.

[4] H. Chui and A. Rangarajan. A new algorithm for non-rigid point matching. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 44–51, 2000.

[5] T. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *Proceedings of the Fifth European Conference on Computer Vision*, volume 2, pages 484–498, 1998.

[6] T. Cootes, C. J. Taylor, D. Cooper, and J. Graham. Active shape models - Their training and application. *Computer Vision and Image Understanding*, 61:38–59, 1995.

[7] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001.

[8] F. De La Torre, Y. Yacoob, and L. Davis. A probabilistic framework for rigid and non-rigid appearance based tracking and recognition. In *Proceedings of the Fourth International Conference on Automatic Face and Gesture Recognition*, pages 491–498, 2000.

[9] G. Hager and P. Belhumeur. Real-time tracking of image regions with changes in geometry and illumination. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 403–410, 1996.

[10] P. L. Hallinan, G. G. Gordon, A. Yuille, P. Giblin, and D. Mumford. *Two- and Three- Dimensional Patterns of the Face*. A. K. Peters, 1998.

[11] M. Irani. Multi-frame optical flow estimation using subspace constraints. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pages 626–633, 1999.

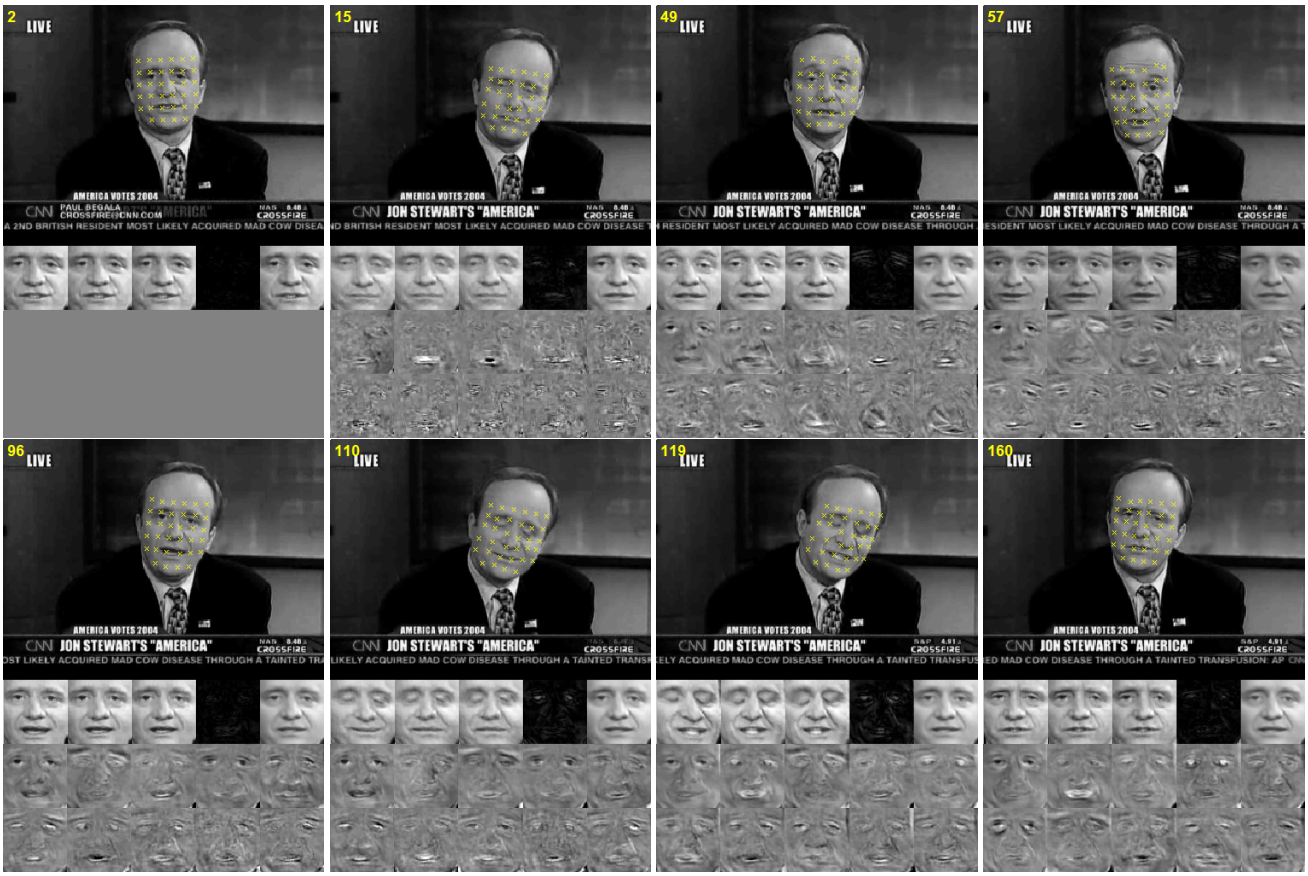


Figure 5. Facial motion. The warped images using our stiff-to-flexible TPS approach are shown from left to right on the second row, followed by the mismatch error of the final warped image, and the template on the far right. Two rows at the bottom are the appearance basis learned using the online appearance learning algorithm [15].

[12] M. Isard and A. Blake. Contour tracking by stochastic propagation of conditional density. In B. Buxton and R. Cipolla, editors, *Proceedings of the Fourth European Conference on Computer Vision*, LNCS 1064, pages 343–356. Springer Verlag, 1996.

[13] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1987.

[14] A. Lanitis, C. J. Taylor, and T. F. Cootes. Automatic interpretation and coding of face images using flexible models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):743–756, 1997.

[15] J. Lim, D. Ross, R.-S. Lin, and M.-H. Yang. Incremental learning for visual tracking. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*. MIT Press, 2005.

[16] R.-S. Lin, D. Ross, J. Lim, and M.-H. Yang. Adaptive discriminative generative model and its applications. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*. MIT Press, 2005.

[17] N. Paragios and R. Deriche. Geodesic active contours and level sets for the detection and tracking of moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(3):266–280, 2000.

[18] S. Sclaroff and J. Isidoro. Active blobs. In *Proceedings of the Sixth IEEE International Conference on Computer Vision*, pages 1146–1153, 1998.

[19] J. Shi and C. Toamsi. Good features to track. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[20] Y. Weiss. Smoothness in layers: Motion segmentation using nonparametric mixture estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 520–526, 1997.

[21] L. Wiskott, J. Fellous, N. Kruger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):775–779, 1997.