Proc. VIIth Digital Image Computing: Techniques and Applications, Sun C., Talbot H., Ourselin S. and Adriaansen T. (Eds.), 10-12 Dec. 2003, Sydney

Contour Tracking without Prior Information

Albert Akhriev[1] and Chang-Yeong Kim[2]

[1] Institute of Information Technologies, RRC “Kurchatov Institute”, Moscow, Russia, aaahaaah@hotmail.com
[2] Color Application Project Team, Samsung Advanced Institute of Technology, Suwon, South Korea, cykim@samsung.com

Abstract. Despite the great advance made in model-based contour tracking, tracking without any specific knowledge about outline contour of an object is still very important in video-sequence processing. This paper addresses the problem of contour tracking while contour’s evolution is restricted by a global motion model only.

1 Introduction

The dynamic snake representation [3] has emerged in recent years as a powerful tool for contour tracking, but simplicity and efficiency of this approach is played down by instability caused by cluttered background, which is typical for real video-sequences. The latter circumstance arises from the snake equation [9] that promotes contour smoothness rather than shape immutability. Various global and local constraints have been proposed to deal with non-rigid nature of real objects, which results in difficult tradeoff problem. Under restrictive constraint the outline contour doesn’t exactly follow the object of interest and may be trapped by a strong occasional edge, but too “soft” contour frequently follows noisy edge instead of actual one.

Some researchers have reported successful contour tracking without global motion constraint. Nguyen *et al.* [6] applied watershed algorithm accompanied by first-order prediction scheme. Fu *et al.* [5] have proposed objective function that utilized color, curvature, area, optic flow information and operated with non-overlapping contour pieces rather than with separate control points. Nevertheless, it’s natural to protect contour’s shape from distortion embedding global motion model into active contour algorithm [1], [4], [8], [10]. Xue *et al.* [13] constructed objective function that is insensitive with respect to affine transformation. Chang and Lin [14] refined velocity field by global affine transformational constraint. Peterfreund [7] extended dynamic snake approach by velocity consistency constraints and weak-perspective motion model.

Along with global motion constraint another important issue is interframe deformation model. In [11], [8], [2] deformation has been represented as a linear combination of shape vectors drawn from some basis, which in its turn depends on the current contour’s shape. We refer such a factorization as a *linear* interframe deformation model.

Our goal is contour tracking without *a priory* information about object. Global motion constraint and linear deformation model are the only allowed restrictions. Our approach has three remarkable features. (1) Unlike other approaches, where the space of shape deformations is severely restricted, we admit all possible deformations to be able to follow unexpected motion, but we rearrange them enforcing global motion model. (2) Our contour prediction method naturally incorporates an arbitrary linear deformation model. (3) Simple and efficient scaling technique makes the tracking process less context dependent.

2 Tracking Algorithm

1. Initialization. $t := 1$. Interframe interval is set to one by definition $\Delta t \equiv 1$.
   (a) Manually delineate the outline contour $\mathbf{r}^1$ of an object of interest in the 1^{st} frame of video-sequence. Initial contour must be closed. The accuracy of delineation depends on interface tool being used. The higher accuracy the less probability of tracking failure.
   (b) Optionally the outline contour $\mathbf{r}^2$ in the 2^{nd} frame can be delineated. Then velocities at the points of initial contour will be estimated, otherwise they will be set to zero. The estimation of initial velocities is useful (but *not* crucial) since a 2^{nd} order motion model is used (section (2.2)).
   (c) Pick up a number of control points evenly spaced along the contour. They will be tracked.
2. Predict contour position $\mathbf{r}^{t+1}$ on the next frame using previous states $\mathbf{r}^t$, $\mathbf{r}^{t-1}$ (section (2.2)).
3. Optimize contour location $\mathbf{r}^{t+1}$ in the next frame for a better fit to image features (section (2.3)).
4. $t := t + 1$. Re-sample control points to maintain uniform spacing along the contour. Proceed with Step 2 untill the last frame.

We represent a contour by a $2N$-vector $\mathbf{r}^t$ consisting of coordinates of N control points evenly spaced along the contour at the moment t. A vector of *interframe* deformation $\mathbf{v}^t$ relating contours of successive frames is represented similarly

$$\mathbf{r}^t = \left(x_0^t, y_0^t, \ldots, x_k^t, y_k^t, \ldots, x_{N-1}^t, y_{N-1}^t\right)^{\mathrm{T}} \tag{1}$$

$$\mathbf{r}^{t+1} = \mathbf{r}^t + \mathbf{v}^t \tag{2}$$

Assume that the deformation at the kth control point at the moment t is a linear function of a parameter vector $\mathbf{\Theta}$ with the matrix defined over coordinates x_k^t, y_k^t

$$\begin{pmatrix} v_{2k+0}^t \\ v_{2k+1}^t \end{pmatrix} = \begin{pmatrix} x_k^{t+1} - x_k^t \\ y_k^{t+1} - y_k^t \end{pmatrix} = \begin{pmatrix} 1 \; x_k^t \; y_k^t \; 0 \; 0 \; 0 \; \ldots \\ 0 \; 0 \; 0 \; 1 \; x_k^t \; y_k^t \; \ldots \end{pmatrix} \mathbf{\Theta} \tag{3}$$

where the matrix represents an affine or a more complex transformation. Setting in turn each motion parameter to unity ($\Theta_i = 1$) and others to zero ($\Theta_j = 0$, $i \neq j$) we obtain a basis $\{\mathbf{v}_i\}$, its dimension equal to the number of motion parameters. The first two vectors $\mathbf{v}_0$ and $\mathbf{v}_1$ correspond, respectively, to horizontal and vertical translations in the image plane. Keeping their directions intact we orthogonalize $\{\mathbf{v}_i\}$ to get an orthonormal basis $\{\mathbf{h}_i\}$ of *basic deformations.* We call the subspace spanned by $\{\mathbf{h}_i\}$ the *motion subspace*, because any deformation induced by a linear transformation lies within this subspace. Note, that basic deformations depend on the location of the contour control points $\mathbf{r}^t$ and the selected motion model (the number of independent parameters). All other transformations of contour shape will be called *non-basic.* For convenience, the basic deformations are combined into an $2N \times M$ matrix $\mathbf{H}$ with vectors of horizontal $\mathbf{h}_0$ and vertical $\mathbf{h}_1$ translation in the first two columns

$$\mathbf{H} = (\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_{M-1}), \qquad \mathbf{h}_j^{\mathrm{T}} \mathbf{h}_i = \delta_{ji} \tag{4}$$

where N is the number of control points, M is the number of independent motion parameters. We have tried affine, weak-perspective, quadratic and cubic motion models with 6, 8, 12 and 20 parameters respectively.

2.1 Motion Stiffness Matrix

The idea of the motion stiffness matrix is to define the deformation energy such that it is zero for basic deformations and non-zero otherwise. Assume that the contour consists of evenly spaced control points connected with each other by elastic links. The contour deformation energy can be naturally defined as a quadratic form in the deformation vector $\mathbf{v}$

$$\mathcal{E}(\mathbf{v}) \approx \mathbf{v}^{\mathrm{T}} \mathbf{K} \, \mathbf{v} \tag{5}$$

where a $2N \times 2N$ stiffness matrix $\mathbf{K}$ depends on control point positions and contour elasticity. Multiplying the stiffness matrix by the projection operator, we obtain the energy term that is insensitive to basic deformations (4)

$$\mathcal{E}^*(\mathbf{v}) = \mathbf{v}^{\mathrm{T}} (\mathbf{I} - \mathbf{H}\mathbf{H}^{\mathrm{T}})^{\mathrm{T}} \mathbf{K} \, (\mathbf{I} - \mathbf{H}\mathbf{H}^{\mathrm{T}}) \, \mathbf{v} = \mathbf{v}^{\mathrm{T}} \mathbf{K}^* \mathbf{v} \tag{6}$$

where $\mathbf{I}$ is a $2N \times 2N$ identity matrix. Regardless of *how* the original matrix $\mathbf{K}$ was constructed, a basic deformation $\mathbf{h}_i$ always satisfies the equation $\mathcal{E}^*(\mathbf{h}_i) = 0$.

Having no *a priory* knowledge about contour's behavior, we'll construct stiffness matrix "manually", decomposing deformation into discrete Fourier series $\mathbf{v} = \sum_{q=0}^{2N-1} a_q \mathbf{f}_q$, where $\mathbf{f}_q$ is $2N$-dimensional discrete Fourier harmonic

$$\begin{array}{ll} \mathbf{f}_{4i-2+0} = (\ldots,\ \sin(2\pi k i/N),\ 0,\ \ldots)^{\mathrm{T}} & \mathbf{f}_{4i-2+2} = (\ldots,\ \cos(2\pi k i/N),\ 0,\ \ldots)^{\mathrm{T}} \\ \mathbf{f}_{4i-2+1} = (\ldots,\ 0,\ \sin(2\pi k i/N),\ \ldots)^{\mathrm{T}} & \mathbf{f}_{4i-2+3} = (\ldots,\ 0,\ \cos(2\pi k i/N),\ \ldots)^{\mathrm{T}} \end{array}$$

where k is the index of control point. The number of control points N must be *odd.* Since coordinates x and y are mixed (see (1)) cosine and sine appear twice.

There is some advantage in use of Fourier decomposition. Constructed on Fourier harmonics, following matrix is computationally effective (it can be computed in $O(N^2)$ operations), it acts like kernel of a low-pass filter in shape space, and it was *experimentally* advocated (brackets [.] means integer part of a digit)

$$\mathbf{K}_f = \sum_{q=0}^{2N-1} \lambda_q \mathbf{f}_q \mathbf{f}_q^{\mathrm{T}}, \qquad \lambda_q = \omega_q^2, \qquad \omega_q = [(q+2)/4] \tag{7}$$

The eigenvalues λ_q increase as a square of frequency ω_q, hence, the lower the harmonic's frequency the less the energy of deformation. Note that $\mathbf{K}_f$ is *shape-independent*, because $\mathbf{f}_q$ depends on indices of control points and doesn't depend on their actual coordinates, i.e. the contour $\mathbf{r}^t$ and the deformation $\mathbf{v}^t$ are independent vectors. After projection $\mathbf{K}_f^* = (\mathbf{I} - \mathbf{H}\mathbf{H}^{\mathrm{T}})^{\mathrm{T}}\mathbf{K}_f(\mathbf{I} - \mathbf{H}\mathbf{H}^{\mathrm{T}})$ the matrix becomes *shape-dependent*, since basic deformations $\{\mathbf{h}_i\}$ are shape-dependent. Nevertheless, the main part of eigenspectrum remains almost intact (only the lowest eigenvalues are significantly disturbed when $M \ll N$), so sub-optimal methods of eigenspectrum estimation are applicable.

In practice, it may sometimes be useful to impose small energy penalty even for basic deformations. In fact, basic deformations are eigenvectors of $\mathbf{K}_f^*$ with zero eigenvalues $\mathbf{K}_f^*\mathbf{h}_i = 0$ (see (6)). The final stiffness matrix has the form

$$\mathbf{K}_m = \mathbf{K}_f^* + \sum\nolimits_{i=2}^{M-1} \alpha_i \mathbf{h}_i \mathbf{h}_i^{\mathrm{T}}\,, \qquad \alpha_i = 0.25\,\lambda_M^* \left(1 + \frac{1}{2}\frac{(\delta A_i - \overline{\delta A})}{\max\limits_i(\delta A_i - \overline{\delta A})}\right) \tag{8}$$

where the scale coefficient α_i is proportional to the smallest non-zero eigenvalue λ_M^* of $\mathbf{K}_f^*$, δA_i is a derivative of internal contour area by the ith basic deformation $\mathbf{h}_i$ (i.e. reaction of internal contour area on $\mathbf{h}_i$), and $\overline{\delta A}$ is the mean derivative. Notice that the coefficient α_i in (8) is not arbitrary, being a fraction of the eigenvalue of the first *non-basic* deformation, so the eigenspectrum is properly balanced. Secondly, the summation in (8) starts with 2, i.e. translations by vectors $\mathbf{h}_0$ and $\mathbf{h}_1$ are not constrained ($\lambda_0 = \lambda_1 = 0$). Thirdly, basic deformations that with smaller change of the internal contour area (δA_i) are less constrained (smaller α_i). The coefficient 0.25 was found *experimentally.*

The overall process is as follows. First, using the model of shape deformations (e.g. (7)) we construct a stiffness matrix $\mathbf{K}$. Second, by projection (6), the deformations induced by the selected linear transformation (3) are shifted into null-space of $\mathbf{K}^*$ (no energy cost). Third, in order to constrain transformations, we impart small eigenvalues to the basic deformation (8) except of translational ones, which remain unconstrained. We call the obtained matrix the *motion stiffness matrix* $\mathbf{K}_m$.

Strictly speaking, formulas (3), (5), *etc.* are applicable in the case of infinitesimal deformations. Fortunately, translation usually prevails in interframe contour's motion, and deformations that distort shape itself are usually small.

2.2 Prediction

We assume that between successive frames a contour undergoes inertial motion in viscous environment. Generalized Lagrange equation describes such a motion

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\mathbf{r}}}\right) - \frac{\partial L}{\partial \mathbf{r}} + \frac{\partial D}{\partial \dot{\mathbf{r}}} = 0\,, \qquad L = \dot{\mathbf{r}}^{\mathrm{T}}\mathbf{M}\dot{\mathbf{r}}\,, \qquad D = \dot{\mathbf{r}}^{\mathrm{T}}\mathbf{C}\dot{\mathbf{r}} + \eta\dot{\mathbf{r}}^{\mathrm{T}}\mathbf{K}_m\dot{\mathbf{r}} \tag{9}$$

where $\mathbf{M}$ is a diagonal matrix of control point masses ($m_{ii} = m$), $\mathbf{C}$ is a diagonal dissipation matrix ($c_{ii} = c$), $\mathbf{K}_m$ is a motion stiffness matrix, η is a constant.

Under the assumption of small interframe deformations (translation may be arbitrary large), the shape-dependent matirx $\mathbf{K}_m$ is almost constant within the time interval $[t, t{+}1]$, consequently, the velocities and deformations are indistinguishable. Formal differentiation of (9) yields a second order motion equation subject to the initial conditions $\mathbf{r}_0 = \mathbf{r}^t$, $\dot{\mathbf{r}}_0 = \mathbf{r}^t - \mathbf{r}^{t-1}$

$$\mathbf{M}\ddot{\mathbf{r}} + \mathbf{C}\dot{\mathbf{r}} + \eta\mathbf{K}_m\dot{\mathbf{r}} = 0 \tag{10}$$

Motion equation (10) differs from the conventional dynamic snake [3] by $\dot{\mathbf{r}}$ in the 3^{rd} term, and describes inertial propagation in a viscous environment in which *non-basic deformations dissipate faster than basic ones.* In this way, equation (10) favours the adopted motion model (3). The predicted contour in the next frame can be easily expressed in terms of eigenpairs $\{\lambda_i, \mathbf{e}_i\}$ of $\mathbf{K}_m$

$$\hat{\mathbf{r}}^{t+1} = \mathbf{r}^t + \sum\nolimits_{i=0}^{2N-1} \left(\frac{1 - e^{-\mu_i}}{\mu_i} \left(\mathbf{e}_i^{\mathrm{T}} \left(\mathbf{r}^t - \mathbf{r}^{t-1} \right) \right) \mathbf{e}_i \right), \qquad \mu_i = \frac{c + \eta\lambda_i}{m} \tag{11}$$

By default we set: $m = 1$, $c = 0.1$ (small common dissipation) and $\eta = m/\lambda_M$, where λ_M is the smallest eigenvalue of the *non-basic* eigen-deformation. The coefficient η defines "operating point" that is set between basic ($\lambda < \lambda_M$) and non-basic ($\lambda \geq \lambda_M$) eigen-deformations. Prediction can be made less time consuming (but also less accurate) if it is applied to only every second control point.

2.3 Optimization

Starting from predicted contour $\hat{\mathbf{r}}^{t+1}$ on the *next* frame, the optimal one $\mathbf{r}^{t+1}$ minimizes following energy

$$\mathcal{E}(\mathbf{r}) = \left(\mathbf{r} - \mathbf{r}^t\right)^{\mathrm{T}} \left(\xi_m\mathbf{K}_m + \xi_v\mathbf{K}_v\right) \left(\mathbf{r} - \mathbf{r}^t\right) + \xi_c\mathbf{r}^{\mathrm{T}}\mathbf{K}_c\mathbf{r} + \xi_a\|\mathbf{r} - \mathbf{r}_a^{t+1}\|^2 \tag{12}$$

where $\mathbf{r}^t$, $\mathbf{r}^{t+1}$ are contour positions in the current and the next frames, respectively, the $2N$-vector $\mathbf{r}_a^{t+1}$ represents points of attraction in the next frame (usually edge points), ξ_m, ξ_v, ξ_c, ξ_a are constants, and $(\mathbf{r} - \mathbf{r}^t)$ is a vector of deformation from the current contour $\mathbf{r}^t$ to the next one. When the deformation vector belongs to the motion subspace spanned by basic deformations $\{\mathbf{h}_i\}$ (section 2), the quadratic form with the motion stiffness matrix $\mathbf{K}_m$ contributes a small energy penalty since $\{\mathbf{h}_i\}$ are eigenvectors of $\mathbf{K}_m$ with small eigenvalues. Therefore, the better transformation from $\mathbf{r}^t$ to $\mathbf{r}^{t+1}$ satisfies the selected motion model (3) the less energy penalty. In this way, the term with $\mathbf{K}_m$ enforces the global motion constraint, making deformations to lie inside the subspace generated by the selected motion model. The cyclic pentadiagonal matrices $\mathbf{K}_v$ and $\mathbf{K}_c$ in (12) are borrowed from Kass *et al.* [9]. The quadratic form with $\mathbf{K}_v$ enforces a coherent movement of neighbor control points. It is almost useless. The quadratic form with $\mathbf{K}_c$ enforces smoothness of the resultant contour $\mathbf{r}$. It prevents instability when the contour tends to loop. The last term in (12) represents the potential energy. Each control point has its own point of attraction on the next frame (section (2.4)), which lies, in our case, near color edges.

Starting from the predicted contour position (11) in the next frame, we minimize (12) from 3 to 5 times, each time re-estimating points of attraction $\mathbf{r}_a^{t+1}$. Usually, the optimization stage moves the control points by less than one pixel (dominance of the global motion constraint). Nevertheless, being accumulated through the tracking process, the resultant modification can be very significant.

By default we set: $\xi_m = 0.5$, $\xi_v = 0.0125$, $\xi_c = 0.0375$, $\xi_a = 0.45$. Matrices $\mathbf{K}_v$ and $\mathbf{K}_c$ are scaled so that they have "unit" spectrum energies: $\mathrm{Tr}(\mathbf{K}_c) = \mathrm{Tr}(\mathbf{K}_v) = \mathrm{Tr}(\mathbf{I})$. The key point is the proper scaling of matrix $\mathbf{K}_m$. By taking into account that $\xi_m \approx \xi_a$ and $\xi_v \approx \xi_c \approx 0$, a simplified solution of (12) can be obtained: $\mathbf{x} = (s\mathbf{K}_m + \mathbf{I})^{-1}\mathbf{y}$, where s is the scale to be found, $\mathbf{x} = \mathbf{r} - \mathbf{r}^t$, $\mathbf{y} = \mathbf{r}_a^{t+1} - \mathbf{r}^t$. Suppose that vector $\mathbf{y}$ is disturbed by white noise with zero mean and unit variance (the value does not matter). For the variance of $\mathbf{x}$ we have

$$2 \leq \mathrm{var}(\mathbf{x}) = \sum_{k=0}^{2N-1} \frac{1}{(s\lambda_k + 1)^2} \leq 2N \quad \Longrightarrow \quad \sum_{k=0}^{2N-1} \frac{1}{(s\lambda_k + 1)^2} = 2N\gamma \tag{13}$$

where $\{\lambda_k\}$ are eigenvalues of $\mathbf{K}_m$ ($\lambda_0 = \lambda_1 = 0$) and $\gamma \in [0 \ldots 1]$ is a constant. If $\gamma \to 1$, then $s \to 0$ (global motion constraint, embedded in $\mathbf{K}_m$, has no effect). If $\gamma \to \frac{1}{N} \approx 0$, then $s \to \infty$ (the global motion constraint prevails). By default we set $\gamma = 0.5$. The latter means that the energy of disturbance of the solution vector is almost equally affected by the first and the last terms of (12), i.e. *the energy of the global motion constraint has about the same influence as potential one.* Given a value of γ, equation (13) can be easily solved for s by dichotomy.

2.4 Points of Attraction

Let's denote by $\mathbf{p}_k^t$ and $\mathbf{p}_k^{t+1}$ the kth control point of the contour in the *current* frame and its estimation in the *next* frame, respectively. The *search path* S_k^{t+1} is a short chain of pixels centered at $\mathbf{p}_k^{t+1}$ and stretched along the normal, [1]. Our goal is to find a single point of attraction $\mathbf{a}_{k,i}^{t+1}$ that belongs to S_k^{t+1} in the next frame for each control point $\mathbf{p}_k^t$ in the current frame. The index i enumerates pixels of the search path. By default, S_k^{t+1} contains 11 pixels.

We define *point vicinity* V_k^t as the intersection of the circle centered at $\mathbf{p}_k^t$ with the region surrounded by the closed contour $\mathbf{r}^t$. A large circle radius (10 pixels) provides suppression of texture and reflections of adjacent objects. We assume that *local* interframe deformations are small and the point vicinity V_k^{t+1} in the next frame will be almost the same up to translation. *Point color* is a mean color vector calculated over point vicinity: $\mathbf{C}_k^t = \langle \mathbf{C}_{x,y}^t \rangle$, where $\mathbf{C}_{x,y}^t$ is a RGB-vector at a pixel $(x, y) \in V_k^t$ of the *current* frame. Prior to correlation with the next frame, the current point color is filtered over time in order to prevent abrupt changes: $\mathbf{C}_k^t := \alpha\mathbf{C}_k^t + (1 - \alpha)\mathbf{C}_k^{t-1}$, $(\alpha = 0.1)$.

For every pixel $\mathbf{a}_{k,i}^{t+1}$ of the search path S_k^{t+1}, we calculate $\mathbf{C}_{k,i}^{t+1}$ as a mean color of pixels of the *next* frame covered by V_k^t shifted to $\mathbf{a}_{k,i}^{t+1}$. Following distribution, giving very "soft" preference for the points with well-correlated colors, we treat as the probability of *color similarity* distribution along the search path

$$p_k^c(i) = 1.2 \cdot \left(\max_i \|\mathbf{C}_{k,i}^{t+1} - \mathbf{C}_k^t\| \right) - \|\mathbf{C}_{k,i}^{t+1} - \mathbf{C}_k^t\| \tag{14}$$

Color gradient, considered as a probability of edge appearance, "softly" arranges points of the search path, where color gradient at $\mathbf{a}_{k,i}^{t+1}$ is the derivative in the direction of maximal color alteration and n_g is a deviation of gradient noise

$$p_k^g(i) = \max(\, \|\nabla \mathbf{C}_{k,i}^{t+1}\|,\, n_g \,) \tag{15}$$

In practice, the LAB color representation is preferable to the RGB one. Color edges can be emphasized by a moderate attenuation of the lightness ($L := L/2$).

We also utilize region-based information in the spirit of Swain and Ballard [12], constructing the probability map as the ratio of known object's color histogram on the current frame and the color histogram of the search region of the next frame. Pixel value of the resultant map has the meaning of probability of object's appearance at a point of the next frame. It is considered as the "fourth" color and participates in eq. (15) along with ordinary colors.

The last probability $p_k^p(i)$ slightly stimulates proximity between $\mathbf{p}_k^{t+1}$ and attraction point, and consistency between normal and gradient direction at an attraction point. The optimal point of attraction maximizes the overall probability. Its index on the search path is defined as: $i^{opt} = \arg\max_i (\, p_k^c(i)\, p_k^g(i)\, p_k^p(i))$.

3 Results and Conclusion

We have presented approach that tracks arbitrary contour through a videosequence under global linear motion model. It contains three main parts: (1) prediction method with embedded motion model, (2) snake-like optimization driven by motion model, (3) scaling method that balances terms in energy equation.

It is important for a tracking program to have as few adjustible parameters as possible. The *single* crucial parameter (available for immediate tuning in our demo-program) is ξ_m in eq. (12). In most cases, its default value $\xi_m = 0.5$ leads to satisfactory results, but sometimes it has to be corrected to make the global motion constraint more ($\xi_m > 0.5$) or less ($\xi_m < 0.5$) dominant. All other numerical parameters mentioned in this paper are experimentally motivated and conservative. Their default values have been used in all reported experiments.

A least-squares in nature, our approach, as well as the Kalman filter, is sensitive to non-Gaussian image noise. Still, it is totally different from the *Kalman Snake* and similar approaches. The Kalman filter evaluates the consistency between the predicted state and the measurement. There is no problem if the object boundary is missed. In that case, the Kalman filter will rely on the prediction. However, when the predicted contour passes near a casual edge the Kalman filter can take a wrong decision relating to data consistency. By contrast, we carefully balance the eigenspectrum of energy matrices, alloting fixed quotas for different energy terms in eq. (13). Bad measurements cannot quickly distort the contour because the adaptively scaled motion stiffness matrix preserves its shape.

Note that the motion stiffness matrix $\mathbf{K}_m$ must be dense, since under global motion constraint each control point affects the fate of all others. However, we are currently considering the reduction of motion constraints from global to local ones. It seems possible to construct a banded matrix $\mathbf{K}_m$ so that every n consecutive control points ($n < N$) will be forced to share the same motion.

It would be interesting for practicians to see as many tracking results as possible in order to compare with their own approaches. We have selected several standard sequences that could be problematic for the dynamic snake and other methods. Typical tracking results are shown in Figs. 1 to 5. A sequence begins from the left-top frame. Figs. 1 to 4 demonstrate the situation where many parts of object boundaries have very weak gradient and there are strong background edges, which could easily trap a locally-parallel contour. Nevertheless, the shape stiffness, provided by matrix $\mathbf{K}_m$, protects the contour from quick distortion. However, too "stiff" contour unable to follow variable object, Fig. 5. We resolve that trade off problem on post-processing step, adjusting object's contour by other means. Despite gradual degradation of sharp corners and failure to follow accurately the varying contour, the tracker locks on the object well even in complex situations of cluttered background. The tracking rate, with 127 control points, was roughly 10 fps on a 2GHz P4 computer. The weak-perspective motion model was experimentally found to do best. Experiments have also confirmed significance and reliability of our contour prediction scheme (section 2.2).

It is often difficult to find the right balance between the color similarity (14) and gradient (15) distributions. Color correlation tends to produce false points of attraction inside an object, whereas a strong casual edge may trap the contour (see Fig. 2, where rightmost contour points were finally trapped by the strong vertical edge). Lacking an *a priory* shape model, it seems unrealistic to be able to protect the contour from such a trap, but we can weaken destructive effects utilizing spatial color information. The possible option is the color histogram method. Having a predicted contour (11) in the next frame, its position can be corrected by direct correlation of color histograms calculated over the internal regions of the current contour $\mathbf{r}^t$ and predicted contour $\hat{\mathbf{r}}^{t+1}$ shifted within a small search range: $-5 \leq \Delta x, \Delta y \leq 5$ pixels. The result is shown in Fig. 6 where a highly variable contour was tracked through the entire sequence.

References

1. Blake, A., Isard, M.: Active Contours. Springer-Verlag, London (1998)
2. Pentland, A., Sclaroff, S.: Closed-Form Solution for Physically Based Shape Modeling and Recognition. IEEE Trans. PAMI **13** (1991) 715–729
3. Terzopoulos, D., Szeliski, R.: Tracking with Kalman snakes. In *Active Vision*, Blake A. and Yuille A., Eds., Cambridge, MA:MIT Press (1992) 3–20
4. Ip, H.H.S., Shen, D.: An affine-invariant active contour model (AI-snake) for model-based segmentation. Image and Vision Computing **16** (1998) 135–146
5. Fu, Y., Erdem, A.T., Tekalp, A.M.: Tracking visible boundary of objects using occlusion adaptive motion snake. IEEE Trans. IP **12** (2000) 2051–2060
6. Nguyen, H.T., Worring, M., Boomgaard, R., Smeulders, A.W.M.: Tracking nonparameterized object contours in video. IEEE Trans. IP **11** (2002) 1081–1091

7. Peterfreund N.: Robust Tracking of Position and Velocity With Kalman Snakes. IEEE Trans. PAMI **21** (1999) 564–569
8. Zhong, Y., Jain, A.K., Dubuisson-Jolly, M.P.: Object tracking using deformable templates. IEEE Trans. PAMI **22** (2000) 544–549
9. Kass M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. IJCV **1** (1988) 321–331
10. Hamarneh, G., Gustavsson, T.: Combining snakes and active shape models for segmenting the human left ventricle in echocardiographic images. IEEE Conf. Computers in Cardiology (2000) 115–118
11. Fisker, R., Schultz, N., Duta, N., Carstensen, J.M.: A general scheme for training and optimization of the Grenander deformable template model. Proc. CVPR Conf. **I** (2000) 698–705
12. Swain, M.J., Ballard, D.H.: Color Indexing. IJCV **7** (1991) 11–32
13. Xue, Z., Li, S.Z., Teoh, E.K.: AI-EigenSnake: an affine-invariant deformable contour model for object matching. Image and Vision Computing **20** (2002) 77–84
14. Chang, Y.L., Lin, Y.T.: Apparatus and Method for Tracking Deformable Objects. US Patent No. 5,999,651 (1999)

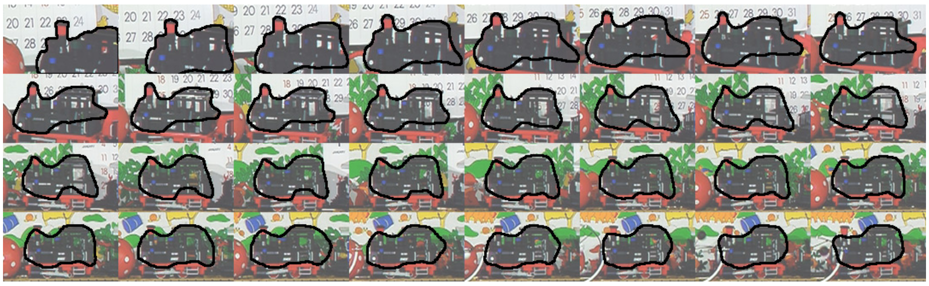

Fig. 1. Tracking of train, “Mobile” sequence, frames 1..299, $\xi_m = 0.75$ in eq. (12).

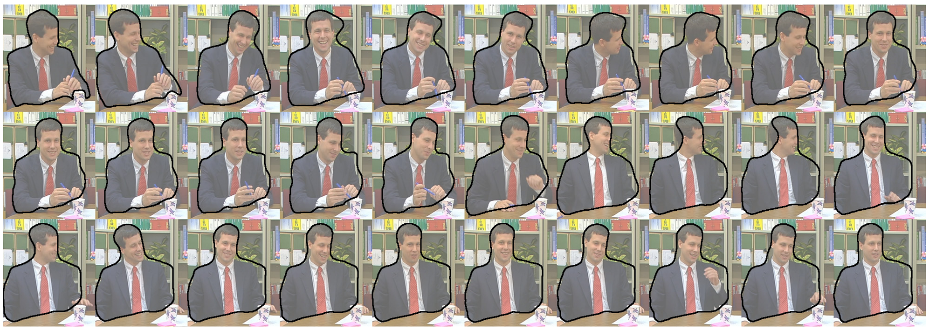

Fig. 2. Tracking of man, “Paris” sequence, frames 1..1065, default parameters.

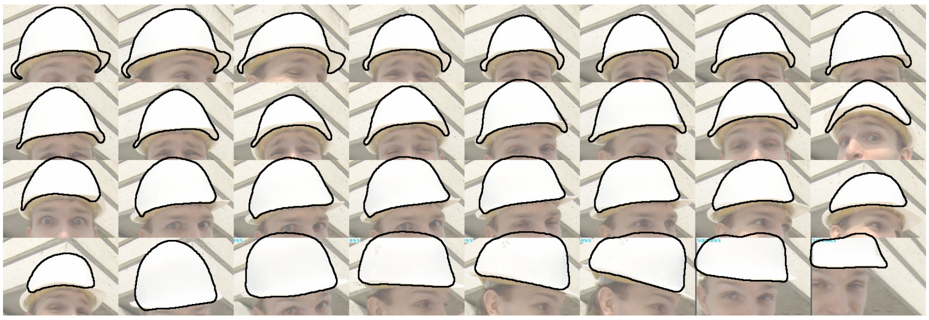

Fig. 3. Tracking of helmet, "Foreman" sequence, frames 1..180, default parameters.

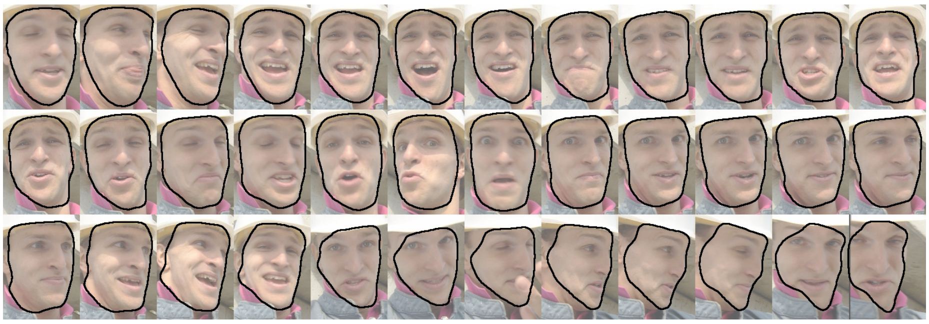

Fig. 4. Tracking of face, "Foreman" sequence, frames 1..180, default parameters.

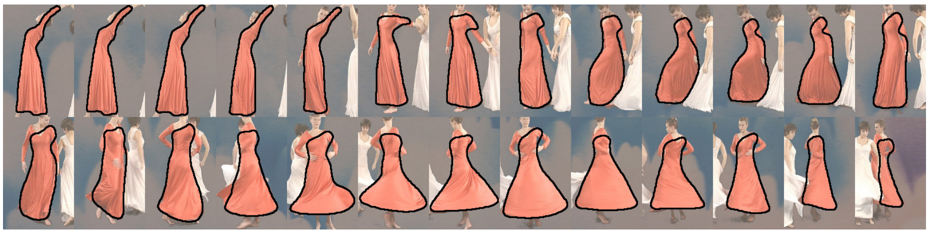

Fig. 5. Tracking of "red" lady, "Dance" sequence, frames 1..155, $\xi_m = 0.2$ in eq. (12).

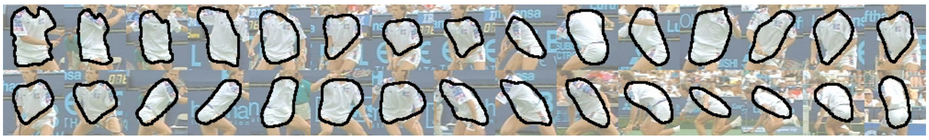

Fig. 6. Tracking of player's shirt in the "Stefan" sequence with additional correction of predicted contour, frames 1..299, default parameters.