Towards Facial Pose Tracking

Peter Torma

Eötvös Loránd University, Budapest
Mindmaker Ltd., Budapest

Csaba Szepesvári

Mindmaker Ltd., Budapest

Abstract

This paper presents a novel facial-pose tracking algorithm using LS-N-IPS (Local Search N-Interacting Particle System), an algorithm that has been introduced recently by the authors. LS-N-IPS is a probabilistic tracking algorithm that keeps track of a number of alternative hypotheses at any time, the particles. LS-N-IPS has three components: a dynamical model, an observation model, and a local-search operator that has to be chosen by the algorithm designer. The main novelty of the algorithm presented here is that it relies on shading information to guide the local search procedure, the idea of the search being to apply a sort-of Hough-transformation to the mapping that renders poses to images. Here we introduce this algorithm and report results on the task of tracking of synthetic facial masks using grey-scale image sequences.

Categories and Subject Descriptors (according to ACM CCS): I.2.10 [Artificial Intelligence]: Vision and Scene Understanding

1. Introduction

Facial pose estimation is an important research area of computer vision whose many possible practical applications include (among others) intelligent user interfaces or interactive computer game controls, just to mention two. However the problem is quite challenging, as the designed algorithm should be able to cope with changing illumination, or changes of facial expression. Previous work are based on three sort of ideas. Some researchers employ feature extraction for eyes and mouth, followed by pose calculation from the feature coordinates [2, 14]. As the result of feature extraction can occasionally yield gross errors, precision of pose estimation can be lost in which case tracking might become unreliable. In the statistical approaches to feature extraction, regularities of the images are exploited to project the image space into a substantially lower dimensional space[11, 6, 3]. Unfortunately these algorithms would require a very large amount of "training" data (to be able to cope with extreme variability of images, eg. changing illumination, positions, orientations, faces) and such large training databases are expensive to obtain. Further, the mapping to the lower dimensional space introduces a sort of quantization error and tracking may loose precision. Elastic graph matching is also a possible solution (after appropriate feature extraction)[15, 10], but these algorithms usually distinguish only five poses (frontal, half profile (2), profile (2)).

In this work we will not attempt to cope with all the problems of the above models but introduce a solution for a simpler problem. With usual computer graphics techniques a facial mask model is generated, and a video is recorded when the mask is moved in space. The problem we are considering here is the tracking of the rotation angles on this animated video, assuming that the model of the mask and the direction of light are known. We take this is an important first step towards pose tracking of faces using true face models. Using models in tracking (geometric, lighting, camera, dynamics, etc.) is a very appealing approach since it includes the possibility of the extension of models to more complex ones (allowing one to model more aspects of reality) whilst keeping the principles of computation fixed.

Turning now to the tracking aspect of our problem, let us note that model based tracking has two main branches. One of them uses the assumption that the object does not move much from frame to frame and employs a local search around the previous object position to locate the object on the next frame[4]. Algorithms in this group tend to be very precise when locked on the object but may have problems if the environment is highly cluttered, or unexpectedly big motion occurs between the frames. The other approach makes use of some filtering algorithm[1, 7, 8], most notably particle filtering methods[9, 5, 13]. The particle filters keep multiple hypothesis about the object state, increasing the filter's robustness in cluttered environments. However, particle filters often give very crude position information unless an excessively large number of particles is used.

N-IPS is a successful particle filter method[†], also known as CONDENSATION in the image processing literature[7]. N-IPS, however, suffers from inefficiency problems if the observation density is uninformative and/or uniformly very small except in a small neighborhood of the true state since then particles which are not in this small vicinity of the "true" state will all have roughly equal observation likelihoods and the filter becomes effectively decoupled from the observations. In order to simplify the exposition, we shall call such densities "peaky" throughout this article.

In this article we use a recent modification of N-IPS, called LS-N-IPS[16] in visual tracking problems. Since LS-N-IPS was designed to overcome the problem of N-IPS with peaky observation densities, therefore it is natural to consider it in visual tracking problems. LS-N-IPS combines local search with particle filtering and thus can be thought of as a combination of the two main streams of vision based tracking research mentioned above. As a consequence, the algorithm inherits the high precision of local search based object tracking methods and the robustness of the particle filter based methods, even when a small number of particles is used.

Our aim here is to construct an algorithm which is capable of tracking faces against varying illumination conditions, as a "minimalist" goal. Therefore it seems reasonable to use some sort of *model-based pose from shading* algorithm. The key is that we assume that we have a geometric model of the object to be tracked (in our case a wire-frame of the facial mask) and our aim is to estimate the pose of the object on a sequence of images. An exhaustive literature search did not yield any results using this approach (model-based tracking using shading information), so we have decided to develop a new algorithm that is being presented here. We think that the primary reason of no prior work on this subject is because pose estimation from a shaded image is a highly difficult problem and no local or heuristic algorithm is guaranteed to succeed. Global pose search, on the other hand is clearly prohibitive in a tracking problem, therefore there seems to be no easy way to estimate a pose of an object from shading information using a geometric model of the object alone. It is the combination of local search and keeping track of multiple alternative hypotheses offered by LS-N-IPS that makes it possible to successfully use local (heuristic) algorithms in tracking problems.

The article is organized as follows: In Section 2 we define the filtering problem. In Section 3 the connection between visual tracking problems and filtering is described. In Section 4 LS-N-IPS is presented and some insight is provided on its behavior. Section 5 introduces the model-based pose from shading "local search operator". Details of experiments and results are presented in Section 6. Conclusions are drawn and future work is outlined in Section 7.

2. The Filtering Problem

Let us consider the discrete time stochastic system

$$X_{t+1} = f(X_t) + V_t, \quad (1)$$

$$Y_t = g(X_t) + W_t, \quad (2)$$

where $t = 0,1,2,\ldots$ denotes the time and V_t, W_t are martingale difference series such that the observation density $p(Y_t = y|X_t = x)$ exists. $X_t \in X$ is called the *state* of the system at time t, X is called the state-space, $Y_t \in Y$ is called the *observation* at time t, Y is the observation space. $f : X \to X$, a measurable mapping, is called the *dynamics* of the system and $g : X \to Y$, another measurable mapping, is called the *observation model*.

The filtering problem consists of the estimation of the posterior distribution of X_t given the past observations $Y_{0:t} = Y_0, \ldots, Y_t$. The posterior at time step t will be denoted by π_t, (suppressing the dependence on the observations $Y_{0:t}$ and the model f, g) :

$$\pi_t(A) = P(X_t \in A|Y_{0:t}),$$

where $A \subset X$ is any measurable subset of X.

The filtering problem has an analytic solution that can be obtained by the repeated application of applying the Bayes-theorem:

$$\pi_{t+1} = \frac{G_{Y_{t+1}} F \pi_t}{(G_{Y_{t+1}} F \pi_t)(X)}, \quad (3)$$

where $F, G_y : M(X) \to M(X)$ are defined by

$$(F\pi)(A) = \int K(x,A) d\pi(x), \quad (4)$$

$$(G_y \mu)(A) = \int_A g(y|x) d\mu(x). \quad (5)$$

Here $K(x,A)$ is the (transition) kernel associated with (1) and $g(y|x)$ is the observation density: $g(y|x) = p(Y_t = y|X_t = x)$.[‡]

[†] The name N-IPS is the abbreviation of "N-Interacting Particle System" and is taken from del Moral[12]

[‡] In the literature Equation (3) is sometimes called the Zakai equa-

Unfortunately, the solution of (3) cannot be computed analytically except in a few exceptional cases when e.g. the model is simple, like in the case of Kalman filters, or when the state and the observation spaces are finite and small. In practice, these simplifications are often invalid and may lead to large errors in the estimates.

3. Visual Tracking as a Filtering Problem

Visual tracking of objects can be cast as a filtering problem as follows: In a typical tracking problem we are interested in the position, pose, and velocity of the object to be tracked, in the outer 3D space, i.e.: the state X_t will correspond in this case to the concatenation of the position, pose and the velocity information. The observation, Y_t shall correspond to the observed image at time step t. Equation (2) tells us that this image can be obtained as a function of the state of the system, plus some noise.

Knowledge of the posterior distribution, π_t, shall allow us to derive all kind of properties of the state of the system. We can, for example, compute the expected value of the state, $E[X_t|Y_{0:t}]$. Given π_t one can also compute higher order moments of it, e.g. the variance, etc.

In this article we are going to work with second-order auto-regressive dynamics. The observation model can eg. be taken the usual correlation-based pattern matching algorithm.[§]

4. The LS-N-IPS Algorithm

The "Local Search"-modified N-IPS algorithm (LS-N-IPS) works as follows (N is the number of particles):

1. Initialization:
   - Let $X_0^{(i)} \sim \pi_0$, $i = 1,2,\ldots,N$ and set $t = 0$.
2. Repeat forever:
   - Compute the proposed next states by $Z_{t+1}^{(i)} = S_\lambda(f(X_t^{(i)}) + W_t^{(i)}, Y_{t+1})$, $i = 1,2,\ldots,N$, according to the dynamical model, and the local search procedure.
   - Compute $w_{t+1}^{(i)} \propto g(Y_{t+1}|Z_{t+1}^{(i)})$, $i = 1,2,\ldots,N$, according to the observational model.[¶]
   - Sample $k_{t+1}^{(i)} \propto (w_{t+1}^{(1)},\ldots,w_{t+1}^{(N)})$, $i = 1,2,\ldots,N$.
   - Let $X_{t+1}^{(i)} = Z_{t+1}^{(k_{t+1}^{(i)})}$, $i = 1,2,\ldots,N$.

tion, or the Feynman-Kac formula for the posterior. Equation (3) also arises in biological studies, e.g. in the theory of genetic algorithms.

§ In this work, we shall employ an even simpler approach that relies on the Euclidean distances of images – an admittedly oversimplified model.

¶ Here $g(y|x)$ denotes the observation density function of the system to be filtered.

The difference between LS-N-IPS and N-IPS(or CONDENSATION) is in the update of the proposed states. LS-N-IPS uses a non-trivial local search operator, S_λ, to "refine" the predictions $f(X_t^{(i)}) + W_t^{(i)}$.

The idea here is that a good search operator, S_λ should satisfy $g(y|S_\lambda(x,y)) \geq g(y|x)$. The parameter $\lambda > 0$ defines the "search length": S_λ is usually implemented as a (local) search trying to maximize $g(y|\cdot)$ around x, in a neighborhood with size λ, e.g.

$$S_\lambda(y|x) = argmax\{g(y|\tilde{x}) \,|\, ||\tilde{x} - x|| \leq \lambda\} \qquad (6)$$

Here $||.||$ can be the scaled maximum norm, or some other appropriate norm.

If $\lambda = 0$, then no local search modification is involved, and we get the usual N-IPS algorithm, while $\lambda \to \infty$ yields to a complete search in X(which is clearly not a desired case as the system dynamics becomes irrelevant). This shows that λ is an important design parameter, which not just improves the particle representation, but also controls the effect of the dynamical model versus the observation.

5. Model-based Local pose from shading

In this section we introduce the local-search component of the algorithm. As mentioned earlier, in our problem the local search is implemented using a model-based pose from shading algorithm. To start with something simple, we restrict the state space (X) to the space of rotations. The task of the local search is then as follows: Given a rough 3D pose of a facial "mask" and a grey-scale image of the same mask whose pose is obtained from the rough pose by the application of a small rotation with unknown angles (α,β,γ), find the angles of that rotation.

The idea of the new algorithm is similar to the idea of Hough-transformation that is used for line and curve detection in image processing: Let the mask be represented by a surface with a wire frame model. For a control point of the surface of the mask, find the corresponding (projected) point on the image using the rough pose estimate. Then, for any point of selection on the image and in the vicinity of the projected control point, determine the rotation that would transform the mask such that

- the control point when projected on the image would be transformed to the selected point; and
- the shade of the so-transformed surface fits the observed image in the neighborhood of the selected point.

We shall see that such a rotation always exists. Do this for all projected control points and points in the vicinity of these projected control points. The calculated rotations can be interpreted as samples of a density over the space of rotations. The algorithm ends with searching for the maximum of this density function (that is, we employ a sort-of maximum likelihood approach): the location associated with the maximum

determines the estimate of the rotation that was to be estimated.

Now, we detail the calculation of the rotation associated with a given control point and a selected point in the neighborhood of the projection of the control point.

Let the control point in the 3D space, on the surface of the mask be $p = (p_1, p_2, p_3)$. The surface of the mask is assumed to be Lambertian. Now, let (x,y) be a point in the vicinity of the projected control point $P\,p$, where P is the projection operation. For simplicity, assume parallel projection (admittedly, a very simple "camera model"; extensions to real camera models can be done easily), so $P\,p = (s\,p_1, s\,p_2)$, where

$$s = \frac{f}{d + p_3}$$

and where f is the focal length and d is the average distance to the object from the focus point (taken to be origo). The constraint that the unknown rotation determined by the angles (α,β,γ) should be such that the projection of the rotated control point should coincide with (x,y) is expressed by

$$s R_{\alpha,\beta,\gamma}\, p = (x,y,t)^T. \tag{7}$$

Here $R_{\alpha,\beta,\gamma}$ is a rotation matrix corresponding to the rotation angles (α,β,γ), and t is an unknown real number that could in principle be computed as the intersection point of the sphere surface obtained from rotating p with all possible rotations and the line $\{(1/s\,x, 1/s\,y, r) : r \in \mathbb{R}\}$. However, we shall not compute t, as we are interested only in computing the rotation angles (α,β,γ) and the calculations can be carried out without ever computing t. Note that equation (**??**) is an under-determined system (in (α,β,γ)).

If (α,β,γ) are small then the rotation matrix $R_{\alpha,\beta,\gamma}$ can be approximated as in the equation below:

$$R_{\alpha,\beta,\gamma} \approx \begin{pmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{pmatrix}.$$

Using this approximation, one transforms equation (**??**) into a corresponding, linear in the parameters equation, where the parameters are (α,β,γ):

$$s \begin{pmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{pmatrix} p = (x,y,t)^T. \tag{8}$$

Now, using the second constraint, that the rotation should be such that the shading of the observed image should fit the shaded image resulting from the projection of the rotated mask gives rise to the equation

$$I(x,y) = l^T R_{\alpha,\beta,\gamma} n(p), \tag{9}$$

which is nothing but the Lambertian law applied to the rotated mask surface. Here $n(p)$ is the normal of the mask surface at point p, l is the light source direction and $I(x,y)$ is the intensity of the image at the point (x,y). Equations (8) and (9) yield

$$M \begin{pmatrix} \alpha \\ \beta \\ \gamma \\ t \end{pmatrix} = \begin{pmatrix} x - p_1 \\ y - p_2 \\ -p_3 \\ I(x,y) - l^T n(p) \end{pmatrix},$$

where

$$M = \begin{pmatrix} 0 & p_3 & -p_2 & 0 \\ -p_3 & 0 & p_1 & 0 \\ p_2 & -p_1 & 0 & -1 \\ -l_2 n_3 + l_3 n_2 & l_1 n_3 - l_3 n_1 & -l_1 n_2 + l_2 n_1 & 0 \end{pmatrix}$$

This is a linear system, which can be solved to obtain α,β,γ.

As the matrix M is sparse, its inverse is fairly simple to get:

$$\begin{pmatrix} -\frac{a_2 p_1}{p_3 c} & -\frac{p_3 a_3 + a_2 p_2}{p_3 c} & 0 & \frac{p_1}{c} \\ \frac{p_3 a_3 + p_1 a_1}{p_3 c} & \frac{a_1 p_2}{p_3 c} & 0 & \frac{p_2}{c} \\ -\frac{a_2}{c} & \frac{a_1}{c} & 0 & \frac{p_3}{c} \\ -\frac{p_1}{p_3} & -\frac{p_2}{p_3} & -1 & 0 \end{pmatrix},$$

where $a_1 = -l_2 n_3 + l_3 n_2$, $a_2 = l_1 n_3 - l_3 n_1$, $a_3 = -l_1 n_2 + l_2 n_1$, and $c = a_1 p_1 + a_2 p_2 + a_3 p_3$.

As we are not interested in the value of t, we derive:

$$\begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = K \begin{pmatrix} x - p_1 \\ y - p_2 \\ I(x,y) - l^T n(p) \end{pmatrix},$$

where

$$K = \frac{1}{c} \begin{pmatrix} -\frac{a_2 p_1}{p_3} & -\frac{p_3 a_3 + a_2 p_2}{p_3} & p_1 \\ \frac{p_3 a_3 + p_1 a_1}{p_3} & \frac{a_1 p_2}{p_3} & p_2 \\ -a_2 & a_1 & p_3 \end{pmatrix} \tag{10}$$

For computational efficiency, note that for point $(x+\Delta x, y+\Delta y)$ the above equation gives

$$\begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = K \begin{pmatrix} (x+\Delta x) - p_1 \\ (y+\Delta y) - p_2 \\ I(x+\Delta x, y+\Delta y) - l^T n(p) \end{pmatrix} \tag{11}$$

$$= K \begin{pmatrix} x - p_1 \\ y - p_2 \\ -l^T n(p) \end{pmatrix} + K \begin{pmatrix} \Delta x \\ \Delta y \\ I(x+\Delta x, y+\Delta y) \end{pmatrix}.$$

This allows us to replace some multiplications with additions by exploiting that points $(x+\Delta x, y+\Delta y)$ are searched sequentially by changing Δx and Δy incrementally. On the whole the local search algorithm is as follows:

1. For all control points p:
   - Calculate the projection of p: $(x,y) = P\,p$.
   - Compute K according to equation (10) and calculate

$$K \begin{pmatrix} x - p_1 \\ y - p_2 \\ -l^T n(p) \end{pmatrix}$$

- For all points $(x+\Delta x, y+\Delta y)$ in the vicinity of (x,y) calculate $(\alpha,\beta,\gamma)^T$ according to equation (11) and store the results in a list L.

2. Treat L as a "cloud of points" in the space of rotations and calculate the rotation $(\alpha,\beta,\gamma)^T$ whose associated local density in the cloud is maximal. The resulting rotation is the outcome of the algorithm.

This last "peak" search can be performed in a number of ways (eg. multi-scale iteratively zooming search, or by fitting a density model to the cloud L, or by some clustering method). Currently we employ a simple discretization based search that seems to fit our purposes well.

6. Results

The algorithm was tested on synthetic images and image sequences. Some images are shown in Figure 2 below. The vertices of the facial mask were taken from the work of Parke [?] (see also http://www.research.digital.com/CRL/books/facebook).

Before trying the local search procedure we were interested in its efficiency. Therefore, in a preliminary series of experiments 5000 random rotation matrices were drawn from the normal density $N((0,0,0)^T, S_\varepsilon)$, where S_ε is a 3×3 diagonal matrix with diagonal entries $(0.15, 0.15, 0.15)^T$. Then, the corresponding synthetic facial masks were drawn using Gouraud shading (the center of masks were fixed). The algorithm was then given the known pose with the rough pose estimate corresponding to the exact center and the rotation "estimate" $(0,0,0)^T$. The local search "length" was set to $\lambda = (\frac{\pi}{20}, \frac{\pi}{20}, \frac{\pi}{20})^T$ in the angle space. We have measured the L_2 error between the estimated and the true rotation angles. Results are shown in Figure 1, where the distribution of the L_2 errors before the local search ("unadjusted errors") and after the local search procedure ("LS-adjusted errors") are shown. The distribution of "unadjusted errors" is given as a reference point and approximates an appropriate χ^2 distribution. It should be evident from the picture that the local search procedure is indeed efficient. Interestingly, the density of errors after local search has two modes. It is yet unknown what causes the second mode corresponding to errors of magnitude 0.3. This issue needs further investigation. Obviously, the surface of the mask is not Lambertian, which might cause problems with the calculations. Despite this, the local search is still able to reduce the error in most of the cases, showing the robustness of the approach. It is reasonable to believe that the robustness follows from searching for the maximum of the empirical density of the *ensemble* of candidate rotation points.

The above algorithm was tested on tracking synthetic image sequences with the same facial mask that was used in the first experiment. We have tested both LS-N-IPS with the local search as defined above and the N-IPS algorithm. The tracking dynamics was a second-order AR dynamics also

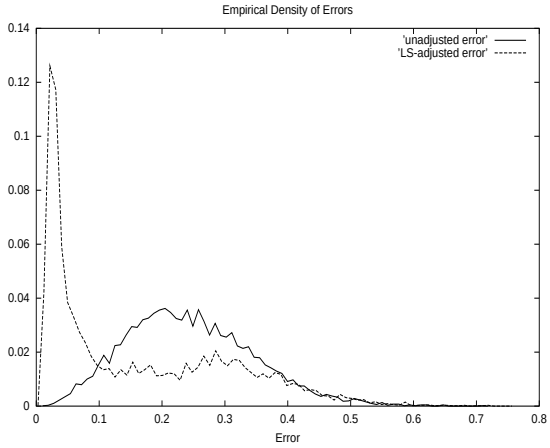

Figure 1: *Empirical density of errors before and after local search.*

used in our previous work.[?] The observation likelihood of an image I given a rotation ρ was simply obtained by rendering the image corresponding to ρ yielding I_ρ, and then taking $1/\|I_\rho - I\|_2^2$, where the images were treated as 2D matrices.

Despite these crude models, using LS-N-IPS we could achieve reliable tracking performance with as few as 250 particles: the object was not lost until the end of the tracking session and the tracking error stayed uniformly small. In the case of N-IPS, even 5000 particles were insufficient for such successful tracking sessions. Typically, with particle numbers in this range, N-IPS lost the object after 30-50 frames and could not catch up with tracking it again until the end of the session.

Figure 2 shows every 15th frame of a typical tracking session using LS-N-IPS employing 500 particles. The duration of the whole sequence is 100 frames. In all these experiments the speed of the head movements were designed such that the frame rate would correspond to roughly 30 frames/second (assuming medium speech head movements). The image resolution was 240×180, bit depth was set to 8. Images in the left hand side column are the observed images corresponding to the "true" rotations, whilst the images in the right hand side column are rendered using the estimate rotations. Clearly, the algorithm successfully tracks the face until the end of the tracking session (the error is the largest at the frames starting about at frame 45 and to frame 60).

7. Conclusions and Future Work

Admittedly, the experiments above should be regarded as preliminary and a lot remains to be done. However, these preliminary results are already quite encouraging: the proposed algorithm was proved to be at least 10 times as efficient as the basic N-IPS algorithm. Also, according to our knowledge, these is the first attempt to track objects using

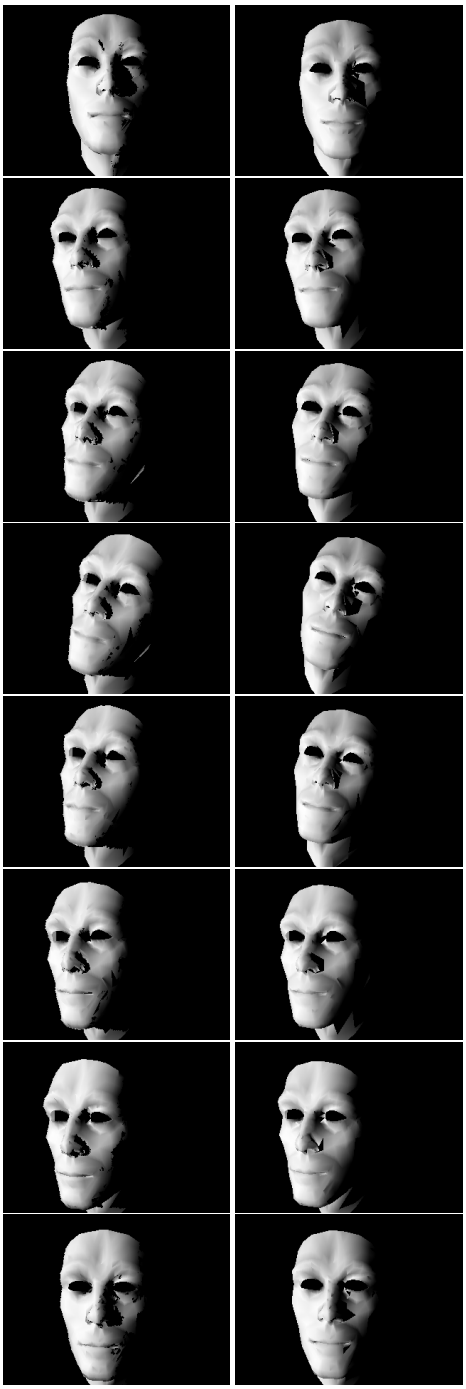

Figure 2: *Tracking result with* 500 *particles using LS-N-IPS. Although tracking accuracy varies with time, the algorithm is able to track the face throughout the whole sequence.*

shading information. We believe that the success of the algorithm can be attributed to the combination of keeping a record of appropriately weighted multiple hypotheses and employing a well-designed local search algorithm. If any of these two components are missing, the algorithm lacks robustness and/or efficiency.

As said above, a lot of open issues remain to be addressed. First of all, in the algorithm above the position of the object was assumed to be known and tracking was restricted to rotations. In theory, the algorithm should be straightforward to extend to this case and one could also use eg. factorial sampling to keep up with the increased dimensionality of the state space.?

Also, the algorithm is already pretty slow on today's computers: the most time-consuming steps of the algorithm are to render the facial mask picture for each particle (using our highly inefficient rendering code, this takes about half a second per picture (assuming a PIII 530MHz machine))[‖] and to run the local search procedure which takes roughly the same amount of time (so LS-N-IPS is twice as slow in this implementation as N-IPS). Obviously, optimizing this computation is badly needed, at this stage of research mainly for speeding up the experiments (one tracking session currently takes ca. 5-6 hours). Rendering can be sped up considerably by exploiting commercially available hardware graphics accelerators, whilst the local search procedure could be sped up by employing eg. a multi-scale search.

It would be interesting to try the algorithm on natural image sequences. Currently we assume that the direction of the light source is known. Given our model based approach it does not seem too difficult to estimate this direction from the images (by including the light source direction in the state space). It would be interesting to extend the local search to handle non-Lambertian surfaces, although the importance of this is not clear *a priori*. The current observation model is admittedly very simple. One could imagine many extensions here, the most interesting direction being to combine inputs on multiple (image) channels (color, shade, texture, edges, etc.) into a simple observation model. Further, elastic surfaces represent yet another very exciting challenge. In theory, the extension of the algorithm to elastic surfaces is straightforward by appropriately extending the state space, showing that the real challenge is to handle the explosion of it.

References

1. Rupert Curwen Andrew Blake and Andrew Zisserman. A framework for spatio-temporal control in the tracking of visual contours. *Int. J. Computer Vision*, 1993.

[‖] This step is needed for both LS-N-IPS and N-IPS.

2. Roberto Brunelli. Estimation of pose and illuminant direction for face processing. Technical Report AIM-1499, 1994.

3. T. Darrell, B. Moghaddam, and A. Pentland. Active face tracking and pose estimation in an interactive room. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'96)*, pages 67–72, San Francisco, CA, June 1996.

4. Gregory D.Hager and Peter N.Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Trans. on PAMI*, 20(10):1025–1039, 1998.

5. Arnaud Doucet. On sequential simulation based methods for Bayesian filtering. *Statistics and Computing*, 10(3):197–208, 1998.

6. Shaogang Gong, Stephen J. McKenna, and McKenna . An Investigation into Face Pose Distributions. In *Second International Conference on Automated Face and Gesture Recognition*, Killington, Vermont, October 1996.

7. Michael Isard and Andrew Blake. CONDENSATION – conditional density propagation for visual tracking. *International Journal Computer Vision*, 29:5–28, 1998.

8. Michael Isard and Andrew Blake. ICondensation: Unifying low-level and high-level tracking in a stochastic framework. *Proc 5th European Conf. Computer Vision*, 1998.

9. Lui Jun and Rong Chen. Sequential monte carlo methods for dynamic systems. *J. Amer. Statist. Assoc.*, pages 1032–1044, 1998. (to appear).

10. Norbert Kruger Laurenz Wiskott, Jean-Marc Fellous and Christoph von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 19(7):775–779, 1997.

11. E. Mandel and P. Penev. Facial feature tracking and pose estimation in video sequences by factorial coding of the low-dimensional entropy manifolds due to the partial symmetries of faces, 2000.

12. Pierre Del Moral. A uniform convergence theorem for the numerical solving of the nonlinear filtering problem. *Journal of Applied Probability*, 35:873–884, 1998.

13. Pierre Del Moral and Salut G. Non-linear filtering using monte carlo particle methods. *C.R. Acad. Sci. Paris*, pages 1147–1152, 1995.

14. Athanasios Nikolaidis. Facial feature extraction and determination of pose.

15. Michael Potzsch Norbert Kruger and Christopf von der Malsburg. Determination of face position and pose with a learned representation based on labeled graphs. *Institut for Neuroinformatik. Internal Report, Ruhr-Universitat, Bochum*, 1996.

16. Péter Torma and Csaba Szepesvári. LS-N-IPS: an improvement of particle filters by means of local search. *In Proc. Non-Linear Control Systems(NOLCOS'01) St. Petersburg, Russia*, 2001.