Monocular Hand Pose Estimation Using Variable Metric Gradient-Descent

Martin de La Gorce - Nikos Paragios
MAS laboratory, Ecole Centrale de Paris, France
{martin.de-la-gorce,nikos.paragios}@ecp.fr

Abstract

In this paper, we propose a novel model-based approach to recover 3D hand pose from 2D images through a compact articulated 3D hand model whose parameters are inferred in a Bayesian manner. To this end, we propose generative models for hand and background pixels leading to a log-likelihood objective function which aims at enclosing hand-like pixels within the silhouette of the projected 3D model while excluding background-like pixels.Segmentation and hand pose estimation are unified through the minimization of a single likelihood function, which is novel and improve overall robustness. We derive the gradient in the hand parameter space of such an area-based objective function, which is new and allows faster convergence rate than gradient free methods. Furthermore , we propose a new constrained variable metric gradient descent to speed up convergence and finally the so called smart particle filter is used to improve robustness through multiple hypotheses and to exploit temporal coherence. Very promising experimental results demonstrate the potentials of our approach.

1 Introduction

Hand gestures play a fundamental role in inter-human communication. An efficient hand motion tracking system would provide natural ways of human-machine interaction in immersed environments. Data gloves could be use as input devices but are expensive while present hardware may inhibit free movements. Vision-based tracking in monocular video stream provides the most natural, non-invasive form of hand motion capture. However the design of an accurate and fast vision-based hand tracking is a difficult task and has been an active search area. To the best of our knowledge, one cannot yet claim the existence of a golden solution to hand tracking in the literature.

Hand tracking methods [1, 14, 16, 19] rely on various assumptions and can be roughly classified into two approaches: (i) single-view tracking and (ii) multi-view tracking. Single-view approaches assume that the hand is filmed from a unique point of view. Such a constraint makes accurate tracking often unattainable as no depth information is available. Model-based and view-based approaches are the most prominent to recover pose from single-view information. View-based methods approximate the inverse mapping from image to hand parameters [14, 16] while model-based methods use a 3D articulated hand model. The projection of the hand model is matched to the image through the minimization of cost function generally based on edges [7, 13, 18, 19], silhouettes - obtained through color segmentation - [6, 23] or optical flow [15].

Multi-view tracking refers to a more powerful information space since the hand is recorded from multiple view-points and often stereo-matching is possible. Model-driven algorithms are the most common selection of such methods. The hand model is then matched to the 3D disparity map obtained from the stereo-matching algorithm [2, 3]. However, one can question the accuracy of such disparities estimation due to the absence of strong features on a hand. More advanced methods assuming the presence of two or more distant cameras have been investigated in [4, 22]. They are based on simultaneous silhouette matching on each image.

In this paper we propose a model-based inference method to recover 3D hand pose from monocular images. The hand is described by an articulated model with limit constraints on each degree of freedom. To evaluate the likelihood of a plausible candidate 3D configuration, we synthesize the corresponding hand silhouette projections in the image plane and measure their likelihood given a generative model for the background and hand skin pixels. In order to avoid convergence in local minima, we combine our approach with the smart-particle filter [2]. The latter method exploits depth information , an important limitation that is not present in our method. The pose density is approximated with a number of particles, which inherit an optimization procedure on top of the perturbation model. Our method bears some concept similarities with previous variational approaches [4, 6, 18, 19] that have considered hand silhouette to be the feature space. However those methods rely on an independent preliminary segmentation of the hand silhouette or edges detection. In our method the segmentation and hand pose estimation are unified through the minimization of a single likelihood function, which is novel and improves overall robustness. Opposed to [6] that was based on a downhill simplex optimization method and [18, 19] that has ignored gradient information, we derive the expression of the gradient in the hand parameter space. To this end, we propose a polygonal approximation of the hand silhouette that facilitates the calculation of the likelihood and its gradient and allows direct use of results from active polygons [21]. The knowledge of this gradient allows to use more advanced optimization methods and speeds up significantly the local search in the parameters space. We proposed a new variable metric constrained gradient descend to further improve convergence rate.

The remainder of this paper is organized in the following fashion: In section 2 the articulated hand model along with the corresponding constraints in the parameter space are presented, while in section 3 we derive the silhouette computation in the image plane. The likelihood function and the optimization are part of section 4 while we conclude with experimental results and discussion.

2 Hand Model & Constraints

The hand is modeled as an articulated object composed of 17 solid components inducing 28 degrees of freedom (DOF). Any possible configuration is described by a vector $\theta \in \mathbb{R}^{28}$. The coordinate system of each part of the hand is resumed by a 4 by 4 matrix K_i which gives the rigid transformation to the absolute frame in homogeneous coordinates.

Such a model should respect (i) kinematic constraints to avoid unrealistic hand configuration during the tracking and (ii) motion constraints. Such conditions improve the performance of the tracker as it impose a natural behavior on the model and reduce the search space.

Kinematics Constraints: Kinematic constraints [10, 11] are either static or dynamic.

Static constraints are independent of the hand's pose, include joint angle limits calculated for all possible hand configurations and are primarily derived from the hand's bone structure. Dynamic constraints are angle limits which depend on the other joint in some specific configuration. They are derived from the tendons structure within the hand. Both types of constraints are resumed in a set of linear inequalities $A\theta \leq b$ defining a convex polyhedron in $\mathbb{R}^{28}$ as feasibility region.

Motion constraints: In the most general case one can assume that the movement is not erratic. Tracking performance can be substantially improved by adding knowledge of what is a natural movement. Among the most simplier movements are the constant speed and acceleration models in the parameter space. While such an approximation seems simplistic, the use of such constraints within to a multiple hypotheses testing approach is a fair compromise between complexity and precision. The constrained constant speed model is given by:

$$\begin{aligned} P(X_{t+1}|X_t) &= \frac{1}{Z} exp(-\frac{1}{2}(DX-X)^t \Sigma_d^{-1}(DX-X)) \text{ if } A\theta_t^n < b \\ &= 0 \text{ else} \end{aligned} \tag{1}$$

Where $X = [\theta, \dot{\theta}]^t$ is the position and speed state vector, $D = \begin{bmatrix} 1 & \triangle t \\ 0 & 1 \end{bmatrix}$, Z is a normalizing factor, and Σ_d a covariance matrix which can be estimated from data using the method proposed by Ghahramani [5]. Samples of the distribution $P(X_{t+1}|X_t)$ can be obtained through repeated sampling of the unconstrained Gaussian distribution until the constraint $A\theta_t^n < b$ is verified. Once such model has been presented and constraints have been introduced, the next step consists of recovering the projection of such a model from the image information.

3 Hand Silhouette Computation

Different hand surface models have been proposed in the literature. The skin surface has been modeled as a triangulated surface in [2], a simplex mesh in [7] while small set of simple primitives (conics and convex polyhedron) have been adopted in [4, 18, 19]. We choose the latter for the main reason that the use of mesh would leads to undifferentiable silhouette position with respect to parameters θ and would hinder the local search (see [8]). We consider a model that refers to an ellipsoid for each phalange, a polyhedron for the palm and deformable polyhedrons for the skin between fingers. The parameters of each ellipsoid and the polyhedron are estimated during the calibration stage. Such a model is a good speed/accuracy compromise for the silhouette computation that is critical in our approach. The hand silhouette computation will be done in three main steps. [i] Calculation of the frames absolute positions K_i, [ii] Projections of the primitive and polygonal approximations of such primitives leading to the set of polygons P, [iii] Computation of the silhouette described by a complex polygon Q.

Our method is based on gradient descent and requires the calculation of the first order variation of the silhouette with respect to the parameter θ. This differentiation should be carried out for each stage of the silhouette computation.

3.1 Differentiation of the ellipsoid Projection

Each ellipsoid of the hand is projected into an ellipse in the image plane which can be described by $[x,y,1]^t C[x,y,1] = 0$. Computing the 3 by 3 matrix C and the jacobian $\partial C/\partial\theta$is a trivial task (see [18]) and will not be detailed here. We aim to differentiate the ellipse contour with respect to θ. To simplify the derivation , we approximate the ellipse with an N edges polygon. To this end, we diagonalize the C matrix:

$$C = VDV^t \text{ with } VV^t = I,\ D = diag([d1,d2,d3])$$

We order the eigen values such that $d_1 \geq d_2 \geq d_3$, and estimate the length of the small and great axes:

$$a = \sqrt{-d_3/d_1},\ b = \sqrt{-d_3/d_2} \tag{2}$$

We approximate the ellipse with the polygon $P_i = \{p_n^i\}$:

$$p_n^i = [x_n/z_n, y_n/z_n]^t \tag{3}$$

with $[x_n,y_n,z_n]^t = V \times [a\ cos(2\pi n/N), b\ sin(2\pi n/N), 1]^t$ Note that the resulting polygon is convex. The projections can now be introduced to the local optimization which requires the estimation of :

$$\frac{\partial P_j}{\partial \theta_i} = \frac{\partial P_j}{\partial C}\frac{\partial C}{\partial \theta_i}$$

In order to calculate $\frac{\partial P_j}{\partial C}$, we need the first order variation of the eigen vectors and eigen values of C. Given the definition of eigen vector we have :

$$(C - d_k I)V_{1:3,k} = [0,0,0]^t$$

let calculate the derivative with respect to C_{ij} and given that $\frac{dC}{\partial C_{ij}} = e_i e_j^t$ with $(e_i)i \in \{1,\ldots,3\}$ being a canonic base of $\mathbb{R}^3$ we get :

$$(e_i e_j^t - \frac{\partial d_k}{\partial C_{ij}})V_{1:3,k} + (C - d_k I)\frac{dV_{1:3,k}}{\partial C_{ij}} = [0,0,0]^t \ (2)$$

We consider $V_{1:3,k}^t \times (2)$:

$$V_{1:3,k}^t(e_i e_j^t - \frac{\partial d_k}{\partial C_{ij}})V_{1:3,k} = -V_{1:3,k}^t(C - d_k I)\frac{\partial V_{1:3,k}}{\partial C_{ij}} \tag{4}$$

as $V_{1:3,k}^t(C - d_k I) = [0,0,0]^t$ we get :

$$\frac{\partial d_k}{\partial C_{ij}} = \frac{V_{1:3,k} e_i e_j V_{1:3,k}}{V_{1:3,k}{}^t V_{1:3,k}} = V_{ik}V_{jk}$$

differentiation of the the constraint $\|V_{1:3,k}\| = 1$ gives $\frac{\partial V_{1:3,k}}{\partial C_{ij}}{}^t V_{1:3,k} = 0$

The first order variation of the eigen vectors is obtained by solving for each eigen value the following linear system :

$$\begin{bmatrix} C - d_k I \\ (V_{1:3,k})^t \end{bmatrix} \frac{\partial V_{1:3,k}}{\partial C_{ij}} = \begin{bmatrix} \frac{\partial d_k}{\partial C_{ij}} V_{1:3,k} - e_i V_{jk} \\ 0 \end{bmatrix}$$

where $\frac{\partial p_n}{\partial C_{ij}}$ can be obtained from $\frac{\partial d_k}{\partial C_{ij}}$ and $\frac{dV_{1:3,k}}{\partial C_{ij}}$ using equations (2) and (3). The projection of polyhedrons is rather trivial and not reported here.

3.2 Silhouette Computation

Once all primitives were projected to the image plane we need to calculate the hand silhouette which corresponds to the boundary of the union of the polygon interiors. The projection of each primitive from 3D into a polygon with respect to its associated frame position produces a set a of polygons. Each polygone is described by a list of vertices $P \equiv \{P_i = (p_1^i, \ldots, p_{n_i}^i)\}_{i=1,\ldots,N_p}$ whose first order derivative $\left[\frac{\partial p_k^i}{\partial \theta_j}\right]$ are also computed. Such vertices are listed counterclokwise. Let denote $\bar{P}_i$ the area within the i^{th} polygon.The sihlouette $[\Gamma \equiv \partial\Omega]$ corresponds to the contour of the unions $[\Omega \equiv \cup_{i=1}^{N} \bar{P}_i]$. The silhouette is described by a complex polygon Q which might have holes. Let denote $Q \equiv \{Q_i \equiv (q_1^i, \ldots, q_{n_i}^i)\}_{i=1,\ldots,N_q}$. Q_1 is the exterior silhouette, $Q_{2:N_q}$ are holes in the silhouette. The silhouette can be written as the union of segments $\Gamma = \cup_{i=1}^{N_q} \cup_{n=1}^{n_i} \overline{q_n^i q_{n+1}^i}$. The problem of computing the union of two convex polygons with m and n vertices respectively has been addressed in the literature and linear time $O(m+n)$ algorithms have been found [12, 20]. However it is not obvious how those algorithms could be extended in order to compute the union of N-polygons. Therefore we address this task through direct comparison for each pair of polygons successively.

4 Pose Estimation

We consider the problem of estimating the hand position, which means recovering appropriate estimates for θ. To this end, we introduce a cost function $L(\theta)$ based on the hand silhouette Γ that reflects the likelihood of the observed image given such a configuration. This function is a log-likelihood function build from hand and background generative models. Despite our specific modeling choice on background/foreground separation, our approach can be adapted to other well known active contours/regions-based segmentation terms. The problem of finding the correct hand position is therefore casted into a non-linear programming problem. Indeed we aim to minimize $L(\theta)$ while satisfying the constraints $A\theta \leq b$. The local minimization will be done through a variable metric gradient descent under constraints.

4.1 Generative Models

We suppose that the observed image is made of four class representing four different elements of the image : 1) the static background, 2) the skin part, 3) foreground which might occlude the hand and 4)parts of the body behind the hand. We do not suppose presegmentation of the image in order to match the hand to a segmented silhouette as done in most silhouette matching methods [4, 6]. Our method unifies segmentation and hand pose estimation in a single optimization problem thus improving overall robustness.

Under the assumption of a static camera, we assume that the background is stationary or changing in a gradual fashion. The background model based on a mixture of Gaussians [17] for each pixel is an excellent compromise between low complexity and fairly good approximation of stationary signals. This yield the background log-likelihood :

$$f_{bk}(x) = -\log(\sum_i w_x^i \mathcal{N}(\mu_x^i, \Sigma_x^i)(I(x)) \tag{5}$$

We model the 3 other classes using a kernel-based approximation (Parzen windows) of the RGB histogram. Some minimal interaction is requested from the user toward recovering an initial form of this non-parametric approximation. Current effort is made toward automatic recovering of these histograms. We note D_{hd}, D_{for} and D_{bd} the respective approximated distributions on the RGB space. Both histograms are thresholded such that non-zeros probability are given to any color.In the absence of spatial inter-pixel dependencies within each part , we obtain the observations log-likelihoods :

$$f_{hd}(x) = -\log(D_{hd}(I(x)),\ \ f_{bd}(x) = -\log(D_{bd}(I(x)),\ \ f_{for}(x) = -\log(D_{for}(I(x))$$

The overall log-likelihood of the hand configuration given the observed image, or the cost function, is finally expressed as :

$$\begin{aligned} L(\theta) &= \int_{\Omega(\theta)} min(f_{hd}(x), f_{for}(x)) + \int_{\bar{\Omega}(\theta)} min(f_{bk}(x), f_{bd}(x), f_{for}(x))dx \\ &= \int_{\Omega(\theta)} f(x)dx + K \end{aligned} \tag{6}$$

where

$$f(x) = min(f_{hd}(x), f_{for}(x)) - min(f_{bk}(x), f_{bd}(x), f_{for}(x)) \tag{7}$$

$$K = \int_{image} min(f_{hd}(x), f_{bd}(x), f_{for}(x))dx \tag{8}$$

with K being a constant that can be pre-calculated for a given frame. As we we calculate (6) several times for a single frame, we can speed up the computation using pre-calculated integrals and the green's divergence theorem as proposed in [9]. The integrals over the area Ω are then reduced to an integral over the silhouette Γ thus reducing the computational complexity. The lowest potential of this cost function with respect to the Θ parameters refers to the most optimal pose configuration. We aim to use the gradient information to speed up convergence. Therefore, the differentiation of such a function with respect to model parameters has to be carried out.

4.2 Likelihood Differentiation

In order to compute the likelihood and its derivative, we use the result obtained by [21] in the context of active polygons. The derivative of the functional $L(\theta)$ with respect to a vertex of the polygon is shown to be:

$$\frac{\partial L}{\partial q_k^i} = J(q_k^i - q_{k-1}^i)\int_0^1 f((1-t)q_{k-1}^i + tq_k^i)tdt + J(q_{k+1}^i - q_k^i)\int_0^1 f((1-t)q_{k+1}^i + tq_k^i)tdt \tag{9}$$

with $J = \left[\begin{smallmatrix} 0 & 1 \\ -1 & 0 \end{smallmatrix}\right]$. Denoting $l_k^i \equiv |q_k^i - q_{k+}^i|$ the lengths of the segment $\overline{q_k^i, q_{k+}^i}$ and n_k^i its outward unit normal vector, we get $J(q_{k+1}^i - q_k^i) = n_k^i l_k^i$. One should point out that the information of the functional f is being integrated along adjacent edges for a vertex, and that each edge gives an orthogonal contribution to its extremity vertices. The resulting vector $\partial L/\partial q_k^i$ can be interpreted as a data force acting on the silhouette vertex

q_k^i. Combining the Likelihood derivative with respect to the silhouette vertices and their derivative with respect to θ, we get :

$$\frac{\partial L(\theta)}{\partial \theta_j} = \sum_{i=1}^{n_q} \sum_{k=0}^{n_i} \frac{\partial L}{\partial q_k^i} \frac{\partial q_k^i}{\partial \theta_j} \tag{10}$$

Through this simple matrix multiplication, forces on the silhouette vertices are transcribed to forces on the hand parameters.

4.3 Optimization

Several methods exist to recover the lowest potential of such a cost function. Due to the explicit calculation of the gradient, efficient methods as quasi-Newton could be used. However using a quadratic step penalization based on the chamfer distance between silhouettes rather than the popular BFGS Hessian approximation yield faster convergence due to nonlinearity of the gradient. Our new optimization method can be assimilated to a variable metric gradient descent under linear constraints. To this end, we update the state vector with the following function :

$$\theta \leftarrow \theta + argmin_{\{\Delta_\theta, A(\theta+\Delta_\theta)\leq b\}} \left(\frac{dL}{d\theta}\Delta_\theta + \frac{1}{\rho}\Delta_\theta^t C_\theta \Delta_\theta \right) \tag{11}$$

The solution of (11) is obtained using a standard quadratic programming method and ρ being an adaptive coefficient controlling the step length. Indeed , without the linear constraints, the solution of (11) would be given by $\Delta_\theta = \rho C_\theta^{-1} dL/d\theta$. The coefficient ρ is adapted such that the decrease of the cost functional is within some range around the predicted one. C_θ is a preconditioning matrix which can be seen as a variable metric. C_θ could be chosen as the approximation of the Hessian using the BFGS method, however due to nonlinearities we choose C_θ such that the quadratic term in eqn (11) locally matches the quadratic chamfer distance between the silhouettes for small variations i.e:

$$D_{qc}(\Gamma(\theta), \Gamma(\theta+\Delta_\theta)) = \Delta_\theta^t C_\theta \Delta_\theta + o(\|\Delta_\theta\|^2) \tag{12}$$

with $D_{qc}(\Gamma_1, \Gamma_2)$ the quadratic chamfer distance between curves that is defined as

$$D_{qc}(\Gamma_1, \Gamma_2) \equiv \int_{\Gamma_1} min_t (\Gamma_1(s) - \Gamma_2(t))^2 ds \tag{13}$$

One can calculate C_θ given the set $Q = (q_k^i)$ of silhouette vertices. Some calculation leads to:

$$D_{qc}(\Gamma(\theta), \Gamma(\theta+\Delta_\theta)) = o(\|\Delta_\theta\|^2) + \frac{1}{3}\sum_{i,k} l_k^i [(\Delta_{q_k^i}.n_k^i)^2 + (\Delta_{q_{k+}^i}.n_k^i)^2 + (\Delta_{q_k^i}.n_k^i \times \Delta_{q_{k+}^i}.n_k^i)^2] \tag{14}$$

with $\Delta_{q_k^i} = \sum_j \frac{\partial q_k^i}{\partial \theta_j} \Delta_{\theta_j}$.

The second term in right side of eqn (14) is quadratic with respect to Δ_θ and therefore can rewritten under the form of eqn.(11). This quadratic term penalizes large steps and lead to a natural scaling between rotations and translations. Directions with little influence on the silhouette are less penalized than directions with great influence. Such a variable metric will improve performance over standard quasi-newton method, however current

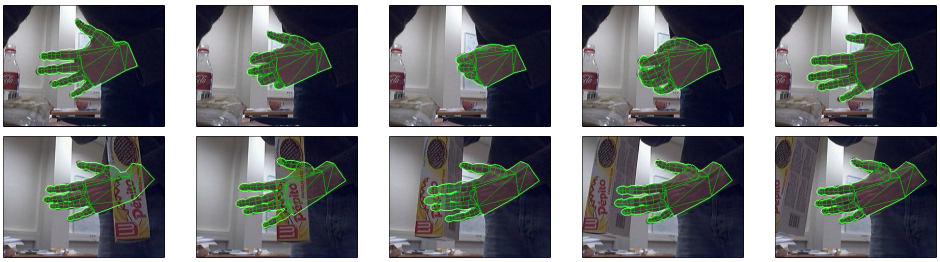

Figure 1: Hand pose estimation results : In first raw the hand is clutched and extended without loosing track. In the second raw , robustness to occlusion is demonstrated

effort is made to combine in a proper way BFGS approximation of the Hessian with the proposed metric to gain efficiency. Such a local optimization method allows to reach quite efficiently a nearby minimum. However, occlusions and depth ambiguities tend to create multiple local minimas. Consequently, given the absence of temporal constraints in our approach the method could fail after several frames. Such a limitation can be dealt through multiple hypotheses testing. Particle filters is a common approach to implement such a framework.

4.4 Smart particle filtering

The particle filtering is based on a sampling approach, where the posterior density function of the parameter $X = [\theta, \dot{\theta}]^t$ for the frame at time t is approximated by a weighted particle set $\{X_t^n, \pi_t^n\}_{n=1}^{2N}$. Several methods exist for the implementation of such a framework, in particular there are different approaches to address the particles resampling. In the most general case, given a *random* perturbation as shown in eqn.(1), one can generate a new hand configuration that is then evaluated using the observed image. The importance of each sample is updated according to the fitness measure between the generated solution and the data. Hand pose estimation aims to recover parameters in a high-dimensional space and therefore classical particle filter approach will fail to produce optimal results - unless a huge number of particles is considered - given the dimensionality of the search space. In order to improve over classical particle filter, while unfortunately still not ensuring global optimal, we adopt the concept of "smart particle filter" [2]. Such an approach combines multiple hypotheses generation with local gradient descent. After propagating the particles, a local gradient descent is performed on parameters θ_t^n and the resulting new particle set is re-weighted such that the original Bayesian distribution is not altered. This allows efficient particle filtering using far fewer samples. We refer the reader to the original paper [2] for a fully detailed explanation.

In order to validate the proposed technique, several sequences with important variations of the hand configurations were considered. User-aided initial hand configuration is provided in the first frame. Experimental results are shown in [Fig. (1)][1]. In the first raw the hand is clutched and extended without loosing track. In the second raw , robustness to occlusion is demonstrated as tracking do not fail when the foreground object occludes parts of the hand.

[1] We strongly encourage the reviewers to see the videos associated with this submission.

We also compared our variable metric optimization method to standard optimization methods for minimization of non linear function with linear constraints, using synthetic data. The graph in [Fig.(2)] compares convergence rate (number of function evaluation versus the the latest best log-likelihood) for the Matlab 7 Medium-Scale Optimization method based on sequential quadratic programming, BFGS update of the Hessian and line search using merit function, with and without gradient information provided. Our method converge faster than the two standard method. As expected, convergence is faster when gradient is provided as a single numerical estimation of the gradient requires 28 function evaluations.

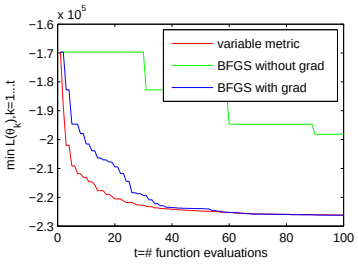

Figure 2: Convergence rates

5 Discussion

In this paper we have proposed a novel optimization method for hand pose estimation from monocular images. Our method is based on hand model made of ellipsoids and polyhedrons and structured as a 28 degrees of freedom kinematic tree. Anatomical limitations are considered through linear constraints on parameters within the minimization process. Pose is determined through the optimization of an objective function defined as an integral within the silhouette of the hand once projected to the image plane. Such a function aims to separate the characteristics of the skin with the ones of the cluttered background,foreground and body. Opposite to prior work, we estimate the derivative of the cost function with respect to the model parameters, leading to a natural method for determining the actual 3D pose. Furthermore, toward addressing limitations of local optimization methods we have considered a multiple-hypotheses testing algorithm. Introducing multiple hypotheses in the process eliminates the risk of convergence to local minima that is often the case of gradient descent optimization techniques. To this end, particle filters are combined with constrained variable metric gradient descent methods.

Efficient and automatic initialization of the method is one of the most important limitation. The case of mobile observation is also a natural extension of our method where more advanced tools for background-foreground separation are to be built. Furthermore, modeling and understanding hand gestures as a succession of articulation parameters through autoregressive models could be a natural extension of the proposed framework. Such an extension could lead to sign-language recognition that is one of the most challenging tasks of gesture analysis.

References

[1] V. Athitsos and S. Sclaroff. Estimating 3D hand pose from a cluttered image. In *CVPR*, volume II, pages 432–439, Madison, WI, June 2003.

[2] M. Bray, E. Koller-Meier, and L. Van Gool. Smart particle filtering for 3d hand tracking. In *AFGR04*, pages 675–680, 2004.

[3] Q. Delamarre and O. Faugeras. Finding pose of hand in video images: A stereo-based approach. In *FG '98: Proceedings of the 3rd. International Conference on Face & Gesture Recognition*, page 585, Washington, DC, USA, 1998. IEEE Computer Society.

[4] Q. Delamarre and O. Faugeras. 3d articulated models and multi-view tracking with silhouettes. In *ICCV99*, pages 716–721, 1999.

[5] Z. Ghahramani and G. Hinton. Parameter estimation for linear dynamical systems, 1996.

[6] P. H. H. Ouhaddi. 3d hand gesture tracking by model registration. In *Proceedings International Workshop on Synthetic-Natural Hybrid Coding and Three Dimensional Imaging (IWSNHC3DI'99), Santorini, Greece*, pages 70–73, 15-17 September 1999.

[7] T. Heap and D. Hogg. Towards 3d hand tracking using a deformable model. In *FG '96: Proceedings of the 2nd International Conference on Automatic Face and Gesture Recognition (FG '96)*, page 140, Washington, DC, USA, 1996. IEEE Computer Society.

[8] S. Ilic and P. Fua. Implicit meshes for modeling and reconstruction. pages II: 483–490, 2003.

[9] I. Jermyn and H. Ishikawa. Globally optimal regions and boundaries. In *ICCV (2)*, pages 904–910, 1999.

[10] J. J. Kuch and T. S. Huang. Vision based hand modeling and tracking for virtual teleconferencing and telecollaboration. In *ICCV '95: Proceedings of the Fifth International Conference on Computer Vision*, page 666, Washington, DC, USA, 1995. IEEE Computer Society.

[11] J. Lee and T. L. Kunii. Model-based analysis of hand posture. *IEEE Comput. Graph. Appl.*, 15(5):77–86, 1995.

[12] J. O'Rourke, C.-B. Chien, T. Olson, and D. Naddor. A new linear algorithm for intersecting convex polygons. *Comput. Graph. Image Process.*, 19:384–391, 1982.

[13] J. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. In *Proceedings of the Fifth International Conference on Computer Vision (ICCV '95)*, pages 612–617, June 1995.

[14] R. Rosales, V. Athitsos, L. Sigal, and S. Scarloff. 3D hand pose reconstruction using specialized mappings. In *ICCV*, volume I, pages 378–385, Vancouver, Canada, July 2001.

[15] D. S. J. O. Shan Lu, Dimitris Metaxas. Using multiple cues for hand tracking and model refinement. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages II: 443–450, 2003.

[16] N. Shimada. Real-time 3-d hand posture estimation based on 2-d appearance retrieval using monocular camera. In *RATFG01*, pages xx–yy, 2001.

[17] C. Stauffer and W. Grimson. Adaptive Background Mixture Models for Real-time Tracking. pages II:246–252, Colorado, USA, 1999.

[18] B. Stenger, P. Mendonca, and R. Cipolla. Model-based hand tracking using an unscented kalman filter. In *BMVC01*, page Session 2: Tracking and Sequences, 2001.

[19] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla. Filtering using a tree-based estimator. In *ICCV*, volume II, pages 1063–1070, Nice, France, October 2003.

[20] G. T. Toussaint. A simple linear algorithm for intersecting convex polygons. *The Visual Computer*, 1(2):118–123, 1985.

[21] G. Unal, A. Yezzi, and H. Krim. Information-theoretic active polygons for unsupervised texture segmentation. *IJCV*, 62(3):199–220, May 2005.

[22] A. Utsumi and J. Ohya. Multiple-hand-gesture tracking using multiple cameras. volume 1, pages 473–478, 1999.

[23] Y. Wu, J. Y. Lin, and T. S. Huang. Capturing natural hand articulation. In *ICCV*, pages 426–432, 2001.