Computer Vision and Image Understanding 114 (2010) 363–372

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

A variational approach to monocular hand-pose estimation

Martin de La Gorce *, Nikos Paragios

MAS Laboratory, Ecole Centrale de Paris, Grande Voie des Vignes, 92 295 Chatenay-Malabry, France

ARTICLE INFO

Article history:
Received 10 April 2008
Accepted 4 September 2009
Available online 15 September 2009

Keywords:
Hand tracking
Hand articulations
Variable metric
Hand-pose estimation
Particle filter
Generative models
Active polygons

ABSTRACT

In this paper, we propose a model-based approach to recover 3D hand pose from 2D images. To this end, we describe the hand structure using a compact 3D articulated model and reformulate pose estimation as a binary image segmentation problem aiming to separate the hand from the background. We propose generative models for hand and background pixels leading to a log-likelihood objective function which aims to enclose hand-like pixels within the projected silhouette of the 3D model while excluding background-like pixels. Segmentation and hand-pose estimation are jointly addressed through the minimization of a single likelihood function. Pose is determined through gradient descent in the hand parameter space of such an area-based objective function. Furthermore, we propose a new constrained variable metric gradient descent to speed up convergence and finally the so called smart particle filter to deal with occlusions and local minima through multiple hypotheses. 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, and could lead to automatic sign-language recognition [1]. Data gloves are commonly used as input devices but are expensive and may inhibit free movements. Vision-based tracking in monocular video streams provides the most natural, non-invasive form of hand motion capture. However the design of such an accurate and fast system is a difficult task and has been an active research 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 [2–6] rely on various assumptions and can be roughly classified into: (i) discriminative methods and (ii) generative methods.

Discriminative approaches (also referred to as view-based methods) approximate the inverse mapping from image to hand parameters [2–4,7–9] through classification or regression techniques. The classifier is constructed from a database that is either generated off-line with a synthetic model or acquired by a camera from a small set of poses. These methods are usually fast and do not require user-aided initialization. However, due to the high dimensionality of the hand pose space, it is not possible to perform dense sampling. Thus the estimated hand pose are not very precise. These methods are well suited for detection and rough pose estimation or recognition of a limited set of predefined poses. Although some discriminative methods have been applied to the multi-view setting [7], most discriminative methods assume that the hand is observed from a unique point of view which makes accurate tracking often unattainable, as no depth information is available.

Generative methods (also referred as model-based methods) use a 2D or 3D articulated hand model whose projection is fitted to the observed image. The fitting is generally done through minimization of a cost function based on extracted cues such as edges in the image [5,10,11], segmented silhouettes [12,13] or using a sum of errors on patches [6,14]. As an alternative to the explicit minimization formulation, the fitting process is sometimes formulated using generalized forces that are derived from cues extracted in the observed image and that are integrated using recursive dynamic model [15]. The two approaches are similar when the forces are derived as gradient of some cost function (referred as *energy* by analogy to mechanics) whose decrease is tested at each iteration of the fitting process. When the forces are not derived as gradient [15], it is often difficult to control the improvement of the fitting through iterations and to ensure its convergence.

The energy to be minimized is sometimes decomposed into a factor-graph [6]. Pose parameters are independently considered for each part of the hand, by defining an energy term for each part. Additional terms ensure that the mechanical constraints between adjacent parts are not violated. The minimization (or inference) is performed using message-passing algorithms that take advantage of the energy factorization to recover good minima (global if the graph is a tree and the variables are discrete). However some restrictive approximations/assumptions have to be done in order

* Corresponding author.
E-mail addresses: martin.de-la-gorce@ecp.fr (M. de La Gorce), nikos.paragios@ecp.fr (N. Paragios).

1077-3142/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2009.09.004

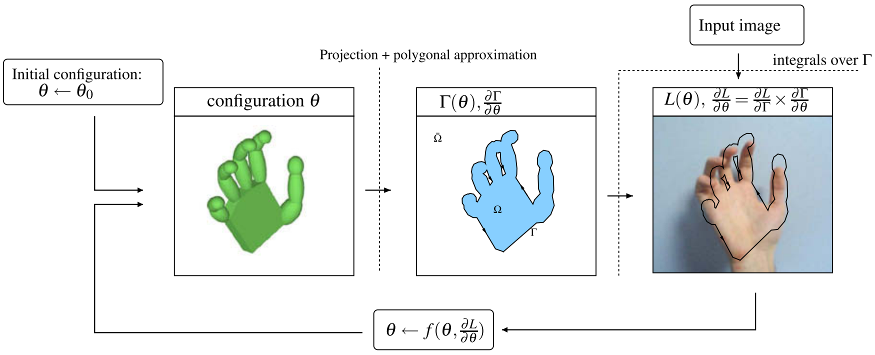

Input image
integrals over Γ
Projection + polygonal approximation
Initial configuration: $\theta \leftarrow \theta_0$
configuration θ
$\Gamma(\theta), \frac{\partial\Gamma}{\partial\theta}$
$\bar{\Omega}$
Ω
Γ
$L(\theta), \frac{\partial L}{\partial\theta} = \frac{\partial L}{\partial\Gamma} \times \frac{\partial\Gamma}{\partial\theta}$
$\theta \leftarrow f(\theta, \frac{\partial L}{\partial\theta})$

Fig. 1. Overview of the hand pose refinement through iterative minimization.

to obtain a factorized form of the energy (in particular while modeling occlusions), while efficient message passing for high dimensional continuous variables is still a research topic. Generative methods that use 3D hand models are suited whenever the hand is recorded from multiple viewpoints and often stereo-matching is possible. A possible approach [16] is to match the hand model surface to the 3D surface or point-cloud obtained from a regular stereo-matching algorithm. However, one can question the accuracy of the estimated depth map obtained by stereo-matching due to the absence of strong texture on hands.

In this paper we propose a model-based Bayesian inference method to recover 3D hand pose from monocular images. The hand is described by an articulated model with constraints limiting the range of each degree of freedom. To evaluate the likelihood of a plausible candidate 3D configuration, we synthesize the corresponding hand silhouette projection in the image plane and measure its likelihood given a generative model for the background and hand skin pixels. The hand pose is iteratively refined through minimization of the negative log-likelihood. One can see a graphical overview of the iterative minimization method for hand pose refinement in (Fig. 1). We propose a new variable metric constrained gradient descent in order to improve the convergence rate. Furthermore, we combine our approach with the so called smart particle filter [17] that combines a multiple hypothesis testing approach with a local search in order to improve robustness of the method to local minima. Our method bears some concept similarities with previous variational approaches [5,15,18,19] which have considered segmented hand silhouette or edges as image cues. However those methods rely on a segmentation of the hand silhouette or edge detection that may be inaccurate, especially with a cluttered background, due to the lack of strong shape constraints. In our method, the segmentation and hand-pose estimation are unified through the minimization of a single likelihood function using gradient descent which is novel and improves overall robustness. In contrast to [5,20] 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]. In contrast with [15] that formulate the fitting process using generalized forces that are integrated using forward recursive dynamic model, our fitting process is explicitly formulated as an energy minimization and thus decrease of the energy (and convergence) can be easily ensured using step length selection during the fitting procedure. The knowledge of the energy gradient allows the use of more advanced optimization methods and significantly speeds up local the search in the parameter space.

The remainder of this paper is organized in the following manner: 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 described in Section 4. Finally, Section 5 presents experimental results followed by a discussion and a conclusion.

2. Hand model and constraints

2.1. Parametrization

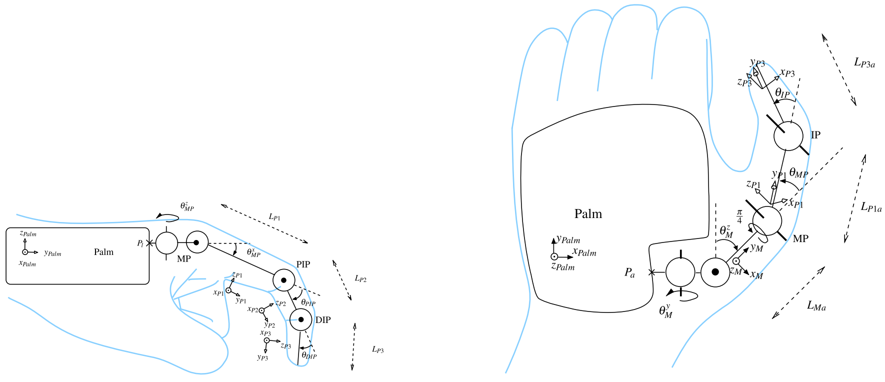
The hand is modeled with an articulated kinematic tree composed of 17 solid components inducing 28 degrees of freedom (DOF). Any possible configuration is described by a vector $\theta \in \mathbb{R}^{28}$. Each finger except the thumb is modeled with the same kinematic chain with 4 DOF. Both the distal interphalangeal joint (DIP), between the second and third (distal) phalanges, and the proximal interphalangeal joint (PIP), between the first and the second phalanges, are modeled with 1 DOF. The metacarpophalangeal (MP) joint, connecting fingers to the palm, is modeled as a saddle joint with 2 DOFs. Despite the fact that the thumb have been generally modeled with 5 DOF with orthogonal axes, we consider a 4 DOF model for the thumb with non-orthogonal joint axes. The wrist position is related to the palm position through two rotation angles. The hand model consists of 22 internal DOF represented by a parameter vector $[\theta_{int}]$. In order to describe the position of the palm with respect to the absolute frame, we consider 6 external DOF (3D translation/rotation) represented by a parameter vector $[\theta_{ext}]$. This leads to a complete model with 28 DOF $[\theta \equiv [\theta_{ext}, \theta_{int}]] \in \mathbb{R}^{28}$.

In order to write down the equations describing the kinematic chains, we define local coordinate systems $(x_i, y_i, z_i, o_i) \in {\mathbb{R}^3}^4$ on each of the 17 solid parts that compose the hand (Fig. 2). Each of this coordinate systems is represented using a 4 by 4 matrix that maps the solid coordinates of each point to its corresponding world coordinates using a simple matrix multiplication. We denote $R_x(\theta)$ the 4 by 4 matrix associated to the the rotation around the axis x with the angle θ. We define rotations matrices R_y, R_z, and the translations matrices T_x, T_y, T_z, T_{xy} in a similar manner.

Such a model should respect kinematic constraints to avoid unrealistic hand configurations during tracking. These conditions improve the performance of the tracker since they aim to impose a natural behavior on the model and reduce the search space.

2.2. Kinematics constraints

Bone, muscle, and tendon structures lead to a number of natural constraints that should be embedded in the hand model. Analysis of such constraints is essential in order to avoid unrealistic hand configurations during tracking but also to reduce the search space. Previous work [22,23] has introduced two types of constraints: the static and the dynamic ones. Static constraints are independent of the hand's pose. They include joint angle limits calculated for all



$$K_{P1l} = K_{palm} \times Txy(p_l) \times Rz(\theta^z_{MPl}) \times Rx(\theta^x_{MPl})$$

$$K_{P2l} = K_{P1l} \times Ty(L_{P1l}) \times Rx(\theta_{PIPl})$$

$$K_{P3l} = K_{P2l} \times Ty(L_{P2l}) \times Rx(\theta_{DIPl})$$

$$K_{Ma} = K_{palm} \times Txy(p_a) \times Ry(\theta^y_M) \times Rz(\theta^z_M)$$

$$K_{P1a} = K_{Ma} \times Ty(L_{Ma}) \times Ry(-\pi/4) \times Rz(\theta_{MP})$$

$$K_{P3a} = K_{P1a} \times Ty(L_{P1a}) \times Rz(\theta_{IPa})$$

Fig. 2. Left hand index kinematic chain and left hand thumb kinematic chain.

possible hand configurations and are primarily derived from the hand's bone structure. Dynamic constraints are angle limits that depend on the other joints for some specific configuration. They are derived from the tendon structure within the hand.

From an optimization point of view, static constraints correspond to bounds on a single angle while dynamic constraints correspond to equalities or inequalities that involve a combination of angles (linear or non-linear). The set of linear inequality constraints on the parameter vector θ can be expressed by enforcing $A\theta \leqslant b$ where A is a single sparse matrix and b is a vector. These inequalities define a convex polyhedron in $\mathbb{R}^{28}$ as the feasibility region. We could define many inequality constraints. However a small number of identified constraints are enough to avoid unrealistic configurations as shown in [22].

For each finger except the thumb we can experimentally observe that in any configuration we have

$$(\theta^x_{MP}, \theta_{PIP}, \theta_{DIP}) \in [0, 100^\circ]^3, \quad \theta^z_{MP} \in [-15^\circ, 15^\circ] \tag{1}$$

The limits on the thumb angles are:

$$\theta_{IP} \in [0^\circ, 110^\circ], \theta_{MP} \in [0^\circ, 80^\circ]$$
$$\theta^z_M \in [-15^\circ, 80^\circ],\ \theta^y_M \in [-30^\circ, 130^\circ]$$

The pose of the wrist respect to the arm is specified by two rotation angles θ^x_W and θ^z_W with limit constraints:

$$\theta^x_W \in [-80, 80],\ \theta^z_W \in [20]$$

For each finger it is nearly impossible to move the DIP joint without moving the PIP joint. Such a dependency has been modeled in [22] by $3\theta_{DIP} - 2\theta_{PIP} = 0$. However we notice that when the hand is clenched into a fist ($\theta_{MP} = \pi/2$ and $\theta_{PIP} = \pi/2$) it is easy to have the DIP joints extended ($\theta_{DIP} = 0$), which clearly violate the linear relation given above. Therefore we choose to replace this constraint by two linear inequalities:

$$3\theta_{DIP} - 2\theta_{PIP} \leqslant 0, \quad 3\theta_{DIP} - 2\theta_{PIP} \geqslant -2\theta^x_{MP} \tag{2}$$

However, since isolated flexion of a finger is restricted by accompanying tension in the palmar interdigital ligament, such flexion might cause flexion of the adjacent fingers. In the same way, a single finger's extension is limited by the flexion of others. Such conditions can be modeled by a set of linear inequalities:

$$\theta^x_{MPb} \leqslant \theta^x_{MPc} + 25^\circ \quad \theta^x_{MPb} \geqslant \theta^x_{MPc} - 55^\circ$$
$$\theta^x_{MPc} \leqslant inf(\theta^x_{MPb} + 55^\circ, \theta^x_{MPd} + 20^\circ)$$
$$\theta^x_{MPc} \geqslant sup(\theta^x_{MPb} - 25^\circ, \theta^x_{MPd} - 45^\circ)$$
$$\theta^x_{MPd} \leqslant inf(\theta^x_{MPc} + 45^\circ, \theta^x_{MPe} + 50^\circ)$$
$$\theta^x_{MPd} \geqslant sup(\theta^x_{MPc} - 20^\circ, \theta^x_{MPe} - 45^\circ)$$
$$\theta^x_{MPe} \leqslant \theta^x_{MPd} + 45^\circ \quad \theta^x_{MPe} \geqslant \theta^x_{MPd} - 50^\circ$$

All those constraints should be taken into account during the model fitting process. This is detailed in the optimization Section 4.3.

3. Hand silhouette computation and its differentiation with respect to the model parameters

Different hand surface models have been proposed in the literature. The skin surface has been modeled using a triangulated mesh [17], a simplex mesh [10] or a small set of simple primitives (conics and convex polyhedra) [5,18,20]. We choose the latter mainly because the use of meshes would lead to a silhouette position that is not differentiable with respect to the parameters θ and would hinder the local search (see [24]).

We consider a model that uses an ellipsoid for each phalange, a polyhedron for the palm and deformable polyhedra for the skin between fingers. Such a model is a good compromise between speed and accuracy for the silhouette computation that is critical in our approach. The parameters of each ellipsoid and the polyhedron are determined in a supervised calibration stage.

The hand silhouette computation is carried out in three main steps:

- calculation of the coordinate system matrix K_i associated to each part
- projection of each primitive followed polygonal approximations of the projections leading to the set of polygons P in the image plane
- computation of the silhouette described by a complex polygon Q.

Our method is based on gradient descent minimization; therefore it requires the calculation of the first-order variation of the silhouette with respect to the parameter vector θ. This differentiation is carried out for each of the three steps of the silhouette computation.

3.1. Ellipsoid projection and differentiation

The perspective projection in the image plane of each ellipsoid, corresponding to a phalanx of the hand, is an ellipse and therefore can be described by an equation $[x, y, 1]^t C[x, y, 1] = 0$. Computing the 3 by 3 matrix C and the Jacobian $\partial C/\partial\theta$ is a trivial task [20] and will not be detailed here. We aim to differentiate the ellipse contour with respect to θ. In order to simplify the derivation, we approximate the ellipse with an N-edge polygon. To this end, we diagonalize the C matrix:

$$C = VDV^t \quad \text{with} \quad VV^t = I, \quad D = diag([d_1, d_2, d_3]) \tag{3}$$

We order the eigen-values such that $d_1 \geqslant d_2 \geqslant d_3$, we define a and b two scalars as follows:

$$a = \sqrt{-d_3/d_1}, \quad b = \sqrt{-d_3/d_2} \tag{4}$$

Then one can approximate the ellipse with the polygon $P_i = \{p_n^i\}$:

$$p_n^i = [x_n/z_n, y_n/z_n]^t \tag{5}$$

with

$$[x_n, y_n, z_n]^t = V \times [a\cos(2\pi n/N), b\sin(2\pi n/N), 1]^t \tag{6}$$

Note that the resulting polygon is convex. The projections can now be differentiated for the purpose of the local optimization, which requires the estimation of:

$$\frac{\partial P_n}{\partial\theta_i} = \frac{\partial P_n}{\partial C}\frac{\partial C}{\partial\theta_i} \tag{7}$$

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

$$(C - d_k I)V_{1:3,k} = [0, 0, 0]^t \tag{8}$$

In order to calculate the derivative of the eigen-values with respect to C_{ij}, one use the fact that $\frac{dC}{\partial C_{ij}} = e_i e_j^t$ with (e_i) $i \in \{1, 2, 3\}$ being the canonical basis of $\mathbb{R}^3$ and differentiate Eq. (8):

$$\left(e_i e_j^t - \frac{\partial d_k}{\partial C_{ij}}\right)V_{1:3,k} + (C - d_k I)\frac{dV_{1:3,k}}{\partial C_{ij}} = [0, 0, 0]^t \tag{9}$$

We consider $V_{1:3,k}^t \times$ (9):

$$V_{1:3,k}^t\left(e_i e_j^t - \frac{\partial d_k}{\partial C_{ij}}\right)V_{1:3,k} = -V_{1:3,k}^t(C - d_k I)\frac{\partial V_{1:3,k}}{\partial C_{ij}} \tag{10}$$

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

$$\frac{\partial d_k}{\partial C_{ij}} = \frac{V_{1:3,k}^t e_i e_j^t V_{1:3,k}}{V_{1:3,k}^t V_{1:3,k}} = V_{ik}V_{jk} \tag{11}$$

The differentiation of the constraint $\|V_{1:3,k}\| = 1$ leads to

$$\frac{\partial V_{1:3,k}}{\partial C_{ij}}^t V_{1:3,k} = 0 \tag{12}$$

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

$$\begin{bmatrix} C - d_k I \\ (V_{1:3,k})^t \end{bmatrix}\frac{\partial V_{1:3,k}}{\partial C_{ij}} = \begin{bmatrix} \frac{\partial d_k}{\partial C_{ij}}V_{1:3,k} - e_i V_{jk} \\ 0 \end{bmatrix} \tag{13}$$

Then, $\frac{\partial p_n}{\partial C_{ij}}$ can be obtained from $\frac{\partial d_k}{\partial C_{ij}}$ and $\frac{dV_{1:3,k}}{\partial C_{ij}}$ using Eqs. (4) and (5).

3.2. Convex polyhedron projection and differentiation

The case of polyhedron projection is simpler. Suppose the ith primitive is a polyhedron; then it can be described as the convex hull of a finite set of vertices $\{s_n^i\}_{n=1}^{N_i}$ in $\mathbb{R}^3$. The contour of the projection matches the 2D convex hull of the projected points. For a polyhedron that is associated with the frame K_j we can project its points onto the image plane according to:

$$\hat{s}_n^i = [x_n^i/z_n^i, y_n^i/z_n^i]^t$$

with $[x_n^i, y_n^i, z_n^i]^t = P \times K_j \times s_n^i$. The convex hull of the set of projected points $\{\hat{s}_n^i\}_{n=1}^{N_i}$ gives a convex polygon with vertices $\{p_n^i\} = \{\hat{s}_{h(n)}^i\} \subset \{\hat{s}_n^i\}_{n=1}^{N_i}$. The function h selects a subset of the projected points. We calculate the first-order derivative of its vertex with respect to θ using he chain rule. The differentiation is straightforward and not detailed here.

3.3. Silhouette computation

Once all primitives are projected onto 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 of polygons described by a list of vertices $P \equiv \{P_i = (p_1^i, \ldots, p_{n_i}^i)\}_{i=1,\ldots,N_p}$. Their first-order derivatives $\left[\frac{\partial p_k^i}{\partial\theta_j}\right]$ are also computed. Such vertices are listed counter-clockwise. Let $\bar{P}_i$ denote the area within the ith polygon. The silhouette $[\Gamma \equiv \partial\Omega]$ 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 that might have holes. Let us 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 and $Q_{2:N_q}$ describe 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 with linear time $(O(m+n))$ algorithms [25,26]. 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 the successive comparison of each pair of polygons. The first-order derivative of the vertex q_k^i with respect to θ, i.e. $\left[\frac{\partial q_k^i}{\partial\theta_j}\right]$ is also computed. In case q_k^i does not result from an intersection between two segments, its derivative is obtained from its corresponding vertex in P. Otherwise it is calculated from the first-order derivative of the extremities of both intersecting segments.

The next step consists of proposing an objective function that measures the consistency of the synthesized silhouette with the observed image.

4. Pose estimation

4.1. Observations models

In order to estimate the hand position through time (i.e. the parameter vector θ_t), we need to measure, for each frame, the consistency between the synthesized hand silhouette contour $\Gamma(\theta_t)$ and the observed image I_t. In the Bayesian framework, this means recovering the probability density $p(I_t|\theta_t)$ of getting the observed image I_t given a hand configuration θ_t. This probability is built from generative models for pixels of the hand, the background and other regions. Those regions are ordered in depth and may occlude each other, leading to a so called 2.1D sketch model [27] (2D regions plus depth ordering). Our model leads to a log-likelihood $L(X_t) = -log(p(I_t|\theta_t))$ that can be interpreted as a cost function with the lowest value corresponding to the hand configuration that is most consistent with the observed image. Despite our specific modeling choice on background and foreground separation our approach can be adapted to other well-known contour-based or area based segmentation functionals $f(\Gamma, I_t)$. This could be done by taking the Gibbs distribution $p(I_t|\theta_t) \propto exp(-f(\Gamma(\theta_t), I_t)/T)$ with T a temperature parameter. Our generative model supposes that the observation is made of four classes representing four different elements of the image: (1) the static background, (2) the skin, (3) foreground which might occlude the hand, and (4) parts of the body behind the hand. Unlike most silhouette matching methods [12,18], we do not suppose pre-segmentation of the image into those four classes in order to match the hand to a segmented silhouette. 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 [28] for each pixel is an excellent compromise between low complexity and fairly good approximation of stationary signals. This yields the following background log-likelihood:

$$f_{bk}(x) = -\log\left(\sum_i w^i_x \mathcal{N}(\mu^i_x, \Sigma^i_x)(I(x))\right) \tag{14}$$

We model the three other classes using a kernel-based approximation (Parzen windows) of the RGB histograms. Some minimal interaction is required from the user in order to recover an initial form of this non-parametric approximation. Current effort is made towards automatic recovery and update of these approximations. We denote d_{hd}, d_{for} and d_{bd} the respective approximated distributions on the RGB space. The histograms are thresholded such that no-zero probability is given to any color. In the absence of spatial inter-pixel dependencies within each part, we obtain the following observation log-likelihoods:

$$\begin{aligned} f_{hd}(x) &= -\log(d_{hd}(I(x))) \\ f_{bd}(x) &= -\log(d_{bd}(I(x))) \\ f_{for}(x) &= -\log(d_{for}(I(x))) \end{aligned} \tag{15}$$

We denote M_{for}, M_{hd}, M_{bd} and M_{bk}, respectively, for the characteristic function of the regions (including occluded parts) corresponding to foreground, hand, body, and background. Rather than require the regions form a partition of the image, we give a depth order: foreground, hand, body, and background. Each part may occlude deeper parts. The probability of observing I_t given M_{hd} is obtained by marginalization over the possible configurations of the masks M_{for} and M_{bd}. The background mask $M_{bk}(x)$ remains equal to one on the entire image support and is therefore not marginalized.

$$\begin{aligned} p(I_t|\theta_t) \equiv p(I_t|M_{hd}(\theta_t)) \propto \textstyle\int_{M_{for}} \int_{M_{bd}} p(I_t|M_{hd}, M_{for}, M_{bd}) \\ p(M_{for})p(M_{bd})dM_{bd}dM_{for} \end{aligned} \tag{16}$$

defining $\prec$ the depth order relation we get the ordering $for \prec hd \prec bd \prec bk$ and we can write the observation likelihood taking occlusions into account:

$$p(I_t|M_{hd}, M_{for}, M_{bd}) = \int_{image} \sum_i d_j(I_t(x))M_i \textstyle\prod_{j\prec i}(1-M_j)dx \tag{17}$$

In order to simplify the computation we approximate by maximizing rather than marginalizing over the other masks and choose an "improper uniform prior" for $p(M_{for})$ and $p(M_{bk})$. Maximizing the expression in Eq. (17) with respect to M_{for} and M_{bd} and taking the negative logarithm of the maximum leads to an approximate log-likelihood of the hand configuration given the observed image. This functional, also referred to as the data cost function, is finally expressed as:

$$\begin{aligned} L(\theta) &= \int_{\Omega(\theta)} min(f_{hd}, f_{for})(x)dx + \int_{\bar{\Omega}(\theta)} min(f_{bk}, f_{bd}, f_{for})(x)dx \\ &= \int_{\Omega(\theta)} f(x)dx + K \end{aligned} \tag{18}$$

with $f(x) = min(f_{hd}, f_{for}) - min(f_{bk}, f_{bd}, f_{for})(x)$ and $K = \int_{image} min(f_{hd}, f_{bd}, f_{for})(x)dx$. The value K does not depend on the synthesized silhouette and can be pre-calculated for a given frame. As we calculate (Eq. (18)) several times for a single frame, we can speed up the computation using image integrals and the Green's divergence theorem as proposed in [29]. The integrals over the area Ω are then reduced to an integral over the silhouette Γ thus reducing the computational complexity:

$$\int_{\Omega(\theta)} f(p)dp = \oint_{\Gamma=\partial\Omega} \langle F, N\rangle ds \tag{19}$$

where N denotes the outward unit normal to Γ, ds the Euclidean arc length element and $F = (F_u, F_v)$ is chosen so that $\nabla \cdot F = f$. We choose $F_u(x,y) = \int_{t=0}^{x} f(t,y)dt$ and $F_v = 0$. The lowest potential of this cost function with respect to the θ parameters refers to the most likely pose configuration given the image.

4.2. Gradient computation

The local maximization of $p(I_t|\theta_t)$ is equivalent to the minimization of the functional $L(\theta_t)$ (Eq. (18)). We aim to use the gradient information to speed up convergence. In order to compute the likelihood 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:

$$\begin{aligned} \frac{\partial L}{\partial q^i_k} = &J(q^i_k - q^i_{k-1}) \int_0^1 f((1-t)q^i_{k-1} + tq^i_k)tdt \\ &+ J(q^i_{k+1} - q^i_k) \int_0^1 f((1-t)q^i_{k+1} + tq^i_k)tdt \end{aligned} \tag{20}$$

with $J = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$.Denoting $l^i_k \equiv |q^i_k - q^i_{k+}|$ to be the lengths of the segment $\overline{q^i_k, q^i_{k+}}$, and n^i_k to be its outward unit normal vector, we get $J(q^i_{k+1} - q^i_k) = n^i_k l^i_k$. 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 $\partial L/\partial q^i_k$ can be interpreted as a data force acting on the silhouette vertex q^i_k. Combining the likelihood derivative with respect to the silhouette vertices and their derivative with respect to θ, we get:

$$\frac{\partial L(\theta)}{\partial \theta_j} = \sum_{i=1}^{n_q} \sum_{k=0}^{n_i} \frac{\partial L}{\partial q^i_k} \frac{\partial q^i_k}{\partial \theta_j} \tag{21}$$

Through this simple matrix multiplication, forces on the silhouette vertices are transcribed to forces on the hand parameters. We remind the reader that despite our specific modeling choice on background and foreground separation our approach can be adapted to other well-known contour and region based segmentation functionals. As far as they can be expressed in the form of area integrals, the gradient can be computed using Eq. (20).

4.3. Variable metric descent

Several methods exist in order to recover the lowest potential of a cost function such as $L(\theta)$. Due to the explicit calculation of the gradient, efficient methods like quasi-Newton could be used. However we consider a quadratic penalization on the step based on the chamfer distance between silhouettes rather than the approximation of the Hessian using the popular Broyden–Fletcher–Goldfarb–Shanno (BFGS) method. This choice yields faster convergence due to nonlinearity of the gradient. Our new optimization method can be assimilated to a variable metric gradient descent under linear constraints.

At each iteration of our method, the parameter vector is updated through minimization of a quadratic form that combines the linear approximation of the cost function (based on the gradient) with a quadratic penalization of the step:

$$\theta \leftarrow \theta + \underset{\{\Delta_\theta | A(\theta+\Delta_\theta) \leqslant b\}}{argmin} \left(\frac{dL}{d\theta}\Delta_\theta + \frac{1}{\rho}\Delta_\theta^t C_\theta \Delta_\theta \right) \tag{22}$$

The solution of (Eq. (22)) is obtained using a standard quadratic programming method and ρ is an adaptive coefficient controlling the step length. Indeed, without the linear constraints, the solution of (Eq. (22)) 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 of the predicted one.

$$L(\theta + \Delta_\theta) - L(\theta) \leqslant \frac{1}{2}\frac{dL}{d\theta}\Delta_\theta \tag{23}$$

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 (Eq. (22)) locally matches the quadratic chamfer distance between the silhouettes for small variations, i.e.:

$$D_{qc}(\Gamma(\theta), \Gamma(\theta + \Delta_\theta)) = \Delta_\theta^t C_\theta \Delta_\theta + o(\|\Delta_\theta\|^2) \tag{24}$$

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

$$D_{qc}(\Gamma_1, \Gamma_2) \equiv \int_{\Gamma_1} min_t(\Gamma_1(s) - \Gamma_2(t))^2 ds \tag{25}$$

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

$$\begin{aligned} D_{qc}(\Gamma(\theta), \Gamma(\theta+\Delta_\theta)) = \frac{1}{3}\sum_{i,k} l_k^i[(\Delta_{q_k^i} \cdot n_k^i)^2 + (\Delta_{q_{k+}^i} \cdot n_k^i)^2 \\ + (\Delta_{q_k^i} \cdot n_k^i \times \Delta_{q_{k+}^i} \cdot n_k^i)^2] + o(\|\Delta_\theta\|^2) \end{aligned} \tag{26}$$

with $\Delta_{q_k^i} = \sum_j \frac{\partial q_k^i}{\partial \theta_j}\Delta_{\theta_j}$. The second term on the right side of Eq. (26) is quadratic with respect to Δ_θ and therefore can be rewritten under the form of Eq. (22). This quadratic term penalizes large steps and leads to a natural scaling between rotations and translations. Directions with a small influence on the silhouette are penalized less than directions with a greater influence. Such a variable metric will improve performance over the standard quasi-Newton method (see [Fig. 8] in Section 5.1). However current effort is made to properly combine BFGS approximation of the Hessian with the proposed metric in order to gain efficiency. Such a local optimization method allows reaching a nearby minimum quite efficiently. However occlusions and depth ambiguities tend to create multiple local minima. Consequently, given the absence of temporal constraints in our approach, the method could fail after several frames. Such a limitation can be dealt with through multiple hypotheses testing. Particle filters are a common approach to implement such a framework.

4.4. Smart particle filtering

Particle filtering is proposed to tackle the problem of Bayesian estimation of the hand trajectory. Given the set of frames $I_0 : t = (I_0, \ldots, I_t)$ we aim to recover the hand position and speed $X_t = [\theta_t, \dot{\theta}_t]^t$. We aim to compute the probability density functions $p(X_t|I_{1:t})$ of present state X_t, based on observations from time 1 to time t $I_{1:t}$. This pdf can be computed in a recursive manner: given our previous pdf $p(X_{t-1}|I_{1:t-1})$ and a new observation I_t , the updated *a posteriori* pdf $p(X_t|I_{1:t})$ can be computed using Bayes' rule:

$$p(X_t|I_{1:t}) = \frac{p(I_t|X_t)p(X_t|I_{1:t-1})}{p(I_t|I_{1:t-1})} \tag{27}$$

with $P(I_t|X_t) = (1/Z_T)\exp(-L(\theta_t)/T)$, where Z_t is a normalizing factor and T a temperature parameter. We model the dynamic process X_t using a simple constant speed model:

$$P(X_{t+1}|X_t) = \frac{1}{Z}exp\left(-\frac{1}{2}(DX - X)^t\Sigma_d^{-1}(DX - X)\right) \quad \text{if } A\theta_{t+1} < b, 0 \text{ else} \tag{28}$$

where $D = \begin{bmatrix} 1 & \triangle t \\ 0 & 1 \end{bmatrix}$($\triangle t$ being the time interval between two successive frames), Z is a normalizing factor, and Σ_d is a covariance matrix which we defined manually (by taking a diagonal matrix). Note that this matrix could be estimated from training data using the method proposed in [30].

Because the process is markovian (i.e. $p(X_t|X_{t-1}) = p(X_t|X_{1:t-1})$ we obtain:

$$p(X_t|I_{1:t-1}) = \int p(X_t|X_{t-1})p(X_{t-1}|I_{t-1})dX_{t-1} \tag{29}$$

The normalization factor in Eq. (27) is given by:

$$p(I_t|I_{1:t-1}) = \int p(I_t|X_t)p(X_t|I_{1:t-1})dX_{t-1}$$

The recursive computation of the prior and the posterior pdf leads to the exact computation of the posterior density. Nevertheless, due to the dimension of the hand configuration space it is impossible to exactly compute the posterior pdf $p(X_t|I_{1:t})$, which must be approximated. Particle filters, which are sequential Monte-Carlo techniques, estimate the Bayesian posterior probability density function (pdf) with a set of samples. Sequential Monte-Carlo methods have been first introduced in [31]. For a more complete review of particle filters, one can refer to [32]. Particle filtering methods approximate the posterior pdf $p(X_t|I_{1:t})$ by a weighted sum of M Dirac distributions centered at $\{X_t^m, m = 1 \ldots M\}$ with the weights $\{w_t^m, m = 1 \ldots M\}$:

$$p(X_t|I_{1:t}) \approx \sum_{n=1}^{N} w_t^n \delta(X_t - X_t^n) \tag{30}$$

The set of pairs $\{X_t^n, w_t^n\}_{n=1}^N$ is called the weighted particle set. Each weight w_t^n reflects the importance of the sample X_t^n in the pdf. Different methods exist for updating the approximated posterior pdf, i.e. the set of particles each time a new frame is observed. One possible method is called Sequential Importance Sampling (SIS) and consists in two steps:

- Draw each new state X_t^m from the previous state X_{t-1}^m using a proposal distribution $q(X_t|X_{1:t}^m, I_t)$.
- The importance of each sample w_t^m is updated according to the fitness measure between the generated solution and the observed data.

$$w_t^m \propto w_{t-1}^m \frac{p(I_t|X_t^m)p(X_t^m|X_{t-1}^m)}{q(X_t^m|X_{t-1}^m, I_t)} \tag{31}$$

The algorithm's performance is dependent on the choice of the importance distribution. Despite the fact that it is not the optimal importance distribution, the transition prior can be used as the importance function because it is easy to draw samples from it. In order to keep a reasonable number of particles and avoid degeneracy of the algorithm due to situations where all but one of the importance weights is close to zero, one could resample the set

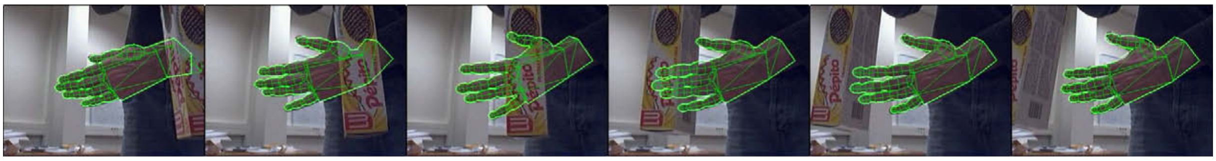


Fig. 3. The hand is clutched and extended without loosing track.

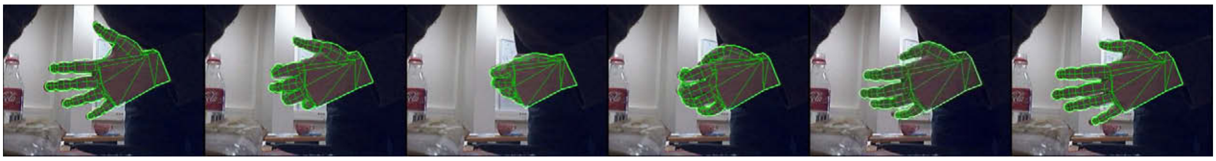


Fig. 4. Robustness to occlusion is demonstrated in this sequence.

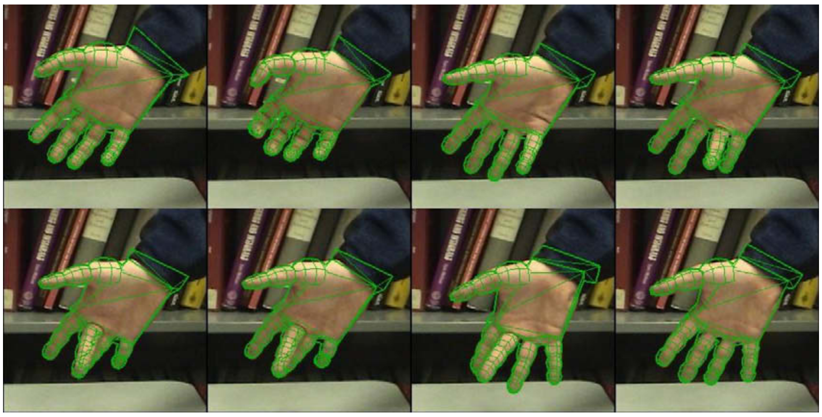


Fig. 5. Six frames from a tracking sequence provided by authors of [6] in which the hand makes grasping motions and individual finger movements. The results are qualitatively equivalent.

of particles after each pdf update. The approximated distribution with N particles of heterogeneous weights $\{X_t^n, w_t^n\}_{n=1}^N$ is resampled into an approximation with N particles of same weight $\{X_t^n, \frac{1}{N}\}_{n=1}^N$. The combination of SIS with resampling is called Sampling Importance Resampling (SIR).

Hand-pose estimation aims to recover parameters in a high-dimensional space. A huge number of particles should be considered in order to get a reasonable approximation of the posterior pdf. Unfortunately, this is not practical due to the heavy computational cost of evaluating $p(I_t|X_t)$ for a single particle. Therefore, a classical particle filter with a reasonable amount of particles is likely to fail to produce optimal results. In order to address this limitation while being able to deal to some extent with the presence of multiple local minima, we adopt the concept of "smart particle filter" [17]. Such an approach combines multiple hypotheses generation with local gradient descent. After propagating the particles, each particle is duplicated with half of the original weight for both particles and a local maximization (which in our case is of variable metric as was earlier explained) of $p(I_t|X_t^n)$ with respect to X_t^n is performed for each copy in order to refine the positions of the particles toward the maximum of closest mode. Because we change some particles' positions, the represented pdf $p(X_t|I_{1:t})$ would be altered if the weights were kept unchanged. The resulting new particle set is therefore re-weighted such that the original Bayesian distribution is not altered. This helps to get more particles near the center of each mode of the distribution and allows efficient particle filtering using far fewer samples. We refer the reader to the original paper [17] for a fully detailed explanation.

5. Discussion

5.1. Validation

In order to validate the proposed technique, several sequences with important variations of the hand configurations were considered. We also tested our method on two sequences provided by authors [5,6]. Experimental tracking results are shown in (Figs. 3–6). User-aided initial hand configuration and calibration is provided for the first frame. The color distributions of the hand and the foreground have been provided for the test using user-aided labeling in the first frame. Current efforts are made in order to get automatic recovery of such distributions. For the sequences (Figs. 3 and 4), we limited the number of particles to 30 and the number of iteration to 10, thus limiting to 300 the number of function evaluations per frame. For the sequences (Figs. 5 and 6), we used a single particle with 300 iterations. The runtime for a frame is proportional to the number of function evaluations and is about 3 min on a 3 GHz intel Xeon$^{\text{TM}}$cpu when using 300 evaluations (0.6 s per particle and per iteration). The code is fully written in Matlab and not vectorized. Note that it is common to obtain a speedup by at last two orders of magnitude by rewriting non-vectorized Matlab code in C. In the first sequence (Fig. 3), the hand is clutched and extended. In the second sequence (Fig. 4), robustness to occlusion is demonstrated as tracking does not fail when the foreground object occludes parts of the hand. In the third sequence (Fig. 5), the hand makes grasping motions and individual finger movements, as been provided by authors of [6]. We show the same frame as the ones shown in [6]. We obtain results that are

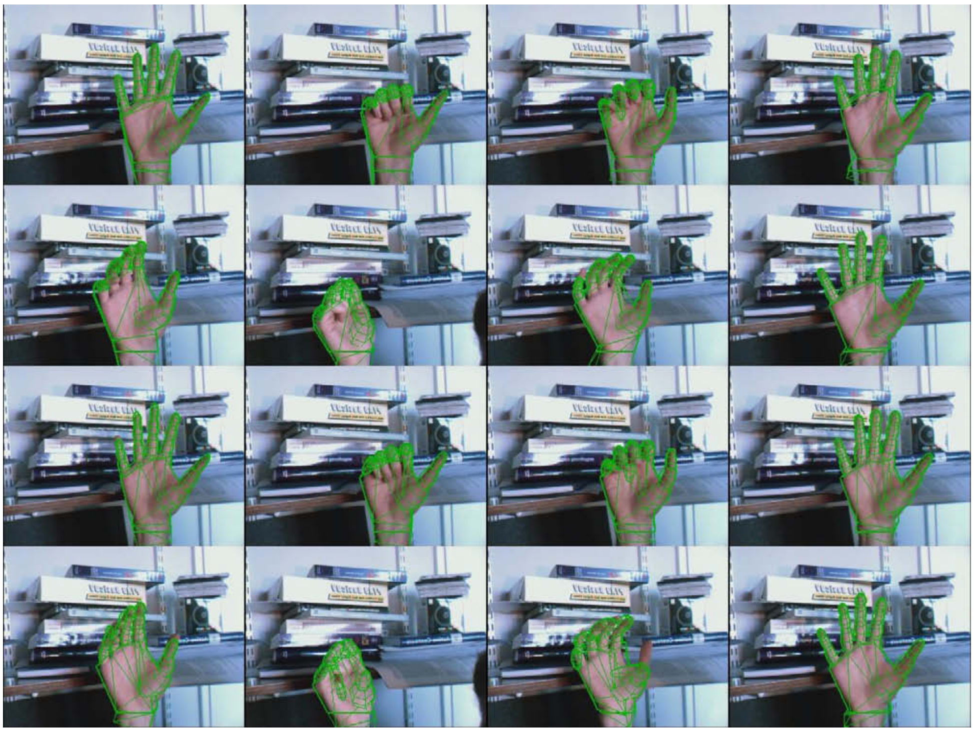


Fig. 6. Tracking results with a sequence provided by authors in [5] with a grasping movement of the hand. the first and second are obtained with some additional linear constraint between fingers angles. The third and fourth row are obtained without additional linear constraints.

qualitatively as good. Unfortunately, ground truth is not available in order to perform finer comparisons of the results.

In the fourth sequence (Fig. 6), the hand is closed and then opened, as been provided by authors of [5]. For computational reasons, the results presented in [5] were obtained with important reduction in the dimension of the hand pose-space, adapted to each sequence individually (8D movements for second sequence – 2 for articulation and 6 for global motion – and 6D rigid movement for third sequence). We tested our algorithm both with and without such reductions (with reduction rows 1 and 2, without reduction rows 3 and 4). To reduce the space of poses, linear inequalities were defined between pairs or triplets of angles. Inequalities were preferred to equalities because this limits the range of possible poses while locally keeping enough freedom of pose variation to make fine registration possible. As one would expect, the results are better when the pose space is reduced (see the first and third images from the left in the fourth row). A limitation of our approach appears while inspecting the results on these sequences. When the hand is closed, the position of the phalanges that fully project within the hand palm are not well estimated. This can be explained by that fact that these phalanges do not contribute to the synthetic hand/background boundary and thus their positions do not affect the objective function being minimized. In other words, our objective function does not allow us to capture information relative to edges of fingers that do not lie on the hand/background silhouette.

We compared the results with both the classic particle filter method (where no local search is performed to update the particle set) and a single hypothesis method where the quasi-newton local search is initialized with the best pose found in the previous frame. We limited the number of particles to 300 for the former and the number of iteration to 300 for the latter, thus obtaining the same overall number of function evaluations per frame.

Some selected frame are presented in (Fig. 7). The classical particle filter method fails after few iterations. The dimensionality of the pose space is too high for the particle filter to be stable with only 300 particles. The single-hypothesis method fails the first time the hand gets occluded by an object. This is due to the fact that the single-hypothesis tracker is unable to escape local minima.

We also compared our variable metric optimization method to standard optimization methods for minimization of non-linear function with linear constraints, using synthetic data. In Fig. 8 we illustrate the fact that our variable metric method yields faster convergence rates than the standard quasi-newton method based on a BFGS update of the Hessian approximate.

5.2. Summary and future work

In this paper we have proposed a novel optimization method for hand-pose estimation from monocular images. Our method is based on a hand model with 28 degrees of freedom for the articulations, while fingers refer to a succession of ellipsoids. Anatomical conditions are considered through constraints within the minimization process. Pose is estimated in a Bayesian manner through a particle filter combined with local optimization of the observation-likelihood function. In contrast to prior work, we estimate the gradient of the cost function with respect to the model parameters, and propose a new constrained variable metric gradient descent method, that improves convergence to the optimal hand parameters. The particle filter framework addresses the limitations of local optimization methods as it introduces multiple hypotheses in the process, eliminating the risk of convergence to local minima. Efficient and automatic initialization of the method is one of the most important limitations. The case of a mobile observer is also a natural extension of our method where more advanced tools for background–foreground separation are to be built. Introducing a more sophisticated model of the hand appearance with adaptive texture and shading is the most prominent direction to help to disambiguate configurations that lead to the same silhouette. Furthermore, modeling and understanding hand gestures as a succession

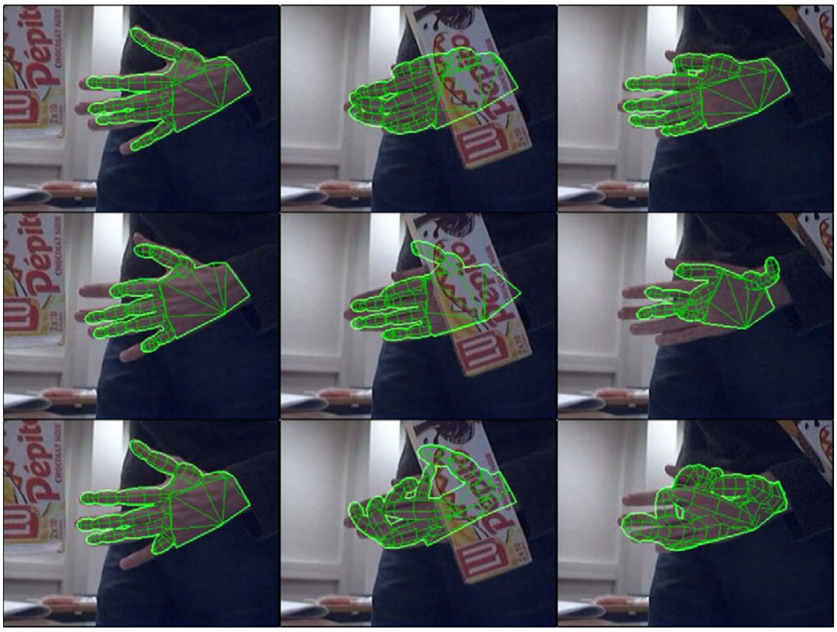


Fig. 7. Tracking results comparisons: each row corresponds, respectively, to the smart particle filter, the classical particle filter and the single hypothesis method.

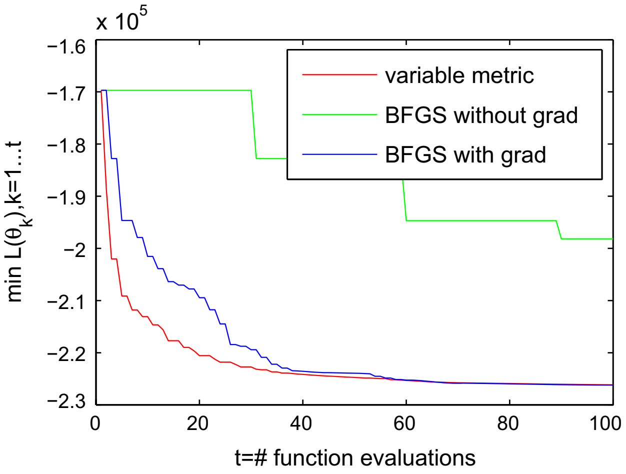


Fig. 8. Convergence rates: (i) variable metric, (ii) without gradient, and (iii) BFGS.

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. more efficient minimization techniques like Belief Propagation methods could further improve the performance of our method and help us dealing with the hand-silhouette projections ambiguities.

References

[1] C. Vogler and D. Metaxas, A framework for recognizing the simultaneous aspects of american sign language. In *CVIU*, volume 81, pages 358–384, 2001.

[2] V. Athitsos and S. Sclaroff, Estimating 3D hand pose from a cluttered image. In *CVPR*, pages II:432–439, 2003.

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

[4] N. Shimada, K. Kimura, and Y. Shirai. Real-time 3-d hand posture estimation based on 2-d appearance retrieval using monocular camera. In *RATFG*, pages 23, 2001.

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

[6] E. Sudderth, M. Mandel, W. Freeman, and A. Willsky. Visual hand tracking using nonparametric belief propagation. In *CVPR*, page 189, 2004.

[7] T.E. de Campos, D.W. Murray, Regression-based hand pose estimation from multiple cameras, CVPR (2006).

[8] A. Agarwal, B. Triggs, Recovering 3d human pose from monocular images, PAMI 28 (1) (2006) 44–58.

[9] Liefeng Bo, C. Sminchisescu, A. Kanaujia, and D. Metaxas. Fast algorithms for large scale conditional 3d prediction. *CVPR*, pages 1–8, 2008.

[10] T. Heap and D. Hogg. Towards 3D hand tracking using a deformable model. In *FG*, page 140, 1996.

[11] J. Rehg and T. Kanade, Visual tracking of high dof articulated structures: An application to human hand tracking. In *ECCV*, pages II:35–46, May 1994.

[12] H. Ouhaddi and P. Horain. 3D hand gesture tracking by model registration. In *Proc. International Workshop on Synthetic–Natural Hybrid Coding and 3D Imaging*, pages 70–73, 1999.

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

[14] J. Rehg and T. Kanade. Model-based tracking of self-occluding articulated objects. In *ICCV*, pages 612–617, 1995.

[15] S. Lu, D. Metaxas, D. Samaras, and J. Oliensis. Using multiple cues for hand tracking and model refinement. In *CVPR*, pages II: 443–450, 2003.

[16] Q. Delamarre and O. Faugeras. Finding pose of hand in video images: A stereo-based approach. In *FG*, page 585, 1998.

[17] M. Bray, E. Koller-Meier, L.J. Van Gool, and N. N. Schraudolph. 3d hand tracking by rapid stochastic gradient descent using a skinning model. *1st European Conference on Visual Media Production (CVMP)*, 2004.

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

[19] D. Knossow, *Paramétrage et Capture Multicaméras du Mouvement Humain*. Phd. manuscript, INPG, INRIA, 655 avenue de l'Europe, 38330 Montbonnot, April 2007.

[20] B. Stenger, P. R. S. Mendonça, and R. Cipolla. Model based 3D tracking of an articulated hand. In *CVPR*, pages II:310–315, 2001.

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

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

[23] J. Kuch and T. Huang, Vision based hand modeling and tracking for virtual teleconferencing and telecollaboration. In *ICCV*, page 666, 1995.

[24] S. Ilic and P.l Fua, Implicit meshes for modeling and reconstruction. In *CVPR*, page II:483, 2003.

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

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

[27] M. Nitzberg and D. Mumford, The 2.1-D sketch. In *ICCV*, pages 138–144, 1990.

[28] C. Stauffer and W. Grimson. Adaptive Background Mixture Models for Real-time Tracking. In *CVPR*, pages II:246–252, 1999.

[29] I. Jermyn and H. Ishikawa. Globally optimal regions and boundaries. In *ICCV*, pages II:904–910, 1999.

[30] Z. Ghahramani and G. Hinton. Parameter estimation for linear dynamical systems. Technical Report CRG-TR-96-2, Department of Computer Science, University of Toronto, 1996.

[31] N.J. Gordon, D.J. Salmond, A.F.M. Smith, Novel approach to Nonlinear/Non-Gaussian Bayesian State Estimation, Radar and Signal Processing, IEEE Proceedings F 140 (1993) 107–113.

[32] N. Gordon, A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking, IEEE Transactions on Signal Processing 50 (2002) 174–188.