Model-Based Hand Tracking with Texture, Shading and Self-occlusions

Martin de La Gorce[1] - Nikos Paragios[1,2] - David J. Fleet[3]

1. Laboratoire MAS, Ecole Centrale de Paris
2. Equipe GALEN, INRIA Saclay - Ile-de-France, Orsay, France
3. Department of Computer Science, University of Toronto, Canada

Abstract

A novel model-based approach to 3D hand tracking from monocular video is presented. The 3D hand pose, the hand texture and the illuminant are dynamically estimated through minimization of an objective function. Derived from an inverse problem formulation, the objective function enables explicit use of texture temporal continuity and shading information, while handling important self-occlusions and time-varying illumination. The minimization is done efficiently using a quasi-Newton method, for which we propose a rigorous derivation of the objective function gradient. Particular attention is given to terms related to the change of visibility near self-occlusion boundaries that are neglected in existing formulations. In doing so we introduce new occlusion forces and show that using all gradient terms greatly improves the performance of the method. Experimental results demonstrate the potential of the formulation.

1. Introduction

Hand gestures play a fundamental role in inter-human communication. Their use in human-machine interaction requires a hand motion tracking system. Data gloves are commonly used as input devices but are expensive and the wires may inhibit free movements. As an alternative, vision-based tracking is the most natural, non-intrusive form of hand pose tracking. However, building a fast and effective vision-based hand pose tracker is challenging. This is due to the high dimensionality of the pose space, the ambiguities due to occlusion, the lack of visible surface texture and the significant appearance variations due to shading. Monocular hand tracking is even more difficult due to depth ambiguities.

Two complementary approaches have been suggested in the literature for monocular hand tracking. *Discriminative* methods aim to recover hand pose from a single frame through classification or regression techniques. [15, 16, 1]. 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. Due to the high dimensionality of the space spanned by possible hand poses, it is not possible to perform dense sampling. As a consequence these methods are well suited for rough initialization or recognition of a limited set of predefined poses.

Generative methods use a 3D articulated hand model whose projection is registered to the observed image[8, 14, 20, 6, 21]. The model projection is synthesized on-line and the registration of the model to the observed image can be done using a local search method. Those methods are good candidates for continuous tracking over consecutive frames with small or predictable inter-frame displacements. A variety of cues such as the distance to edges, segmented silhouettes [12, 23] or optical flow [11] can be considered for the registration. The method proposed in [19] aims to combine both approaches but does not use on-line synthesis and thus allows only limited refinement of the pose.

One of the key problems in monocular hand tracking concerns the existence of depth ambiguities when estimating hand pose. Of course, we aim to reduce ambiguity where possible by exploiting image information. One potential source of information derives from the obvious dependence of the image size of each part on depth. Unfortunately image size measurements have limited utility because projected size also depends on relative pose as well as 3D. Thus the estimation of depth from image size is sensitive to errors in the geometry of the 3D hand model. Shading and motion provide two further depth cues, however they have not been used extensively in the context of articulated tracking (but see [11, 2]). In particular, the use of shading cues requires an accurate model of surface shape, which can be difficult to obtain. Second, the lack of significant surface markings (i.e., texture) on hands means that optical flow estimation is often under-constrained. A further barrier to the use of shading and optical flow is caused by the large number of depth discontinuities that commonly occur as parts of the hand occlude one another. Flow esti-

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

mation is unreliable near occlusion boundaries, as is shape from shading.

This paper introduces a new analysis-by-synthesis formulation of the hand tracking problem that allows one to exploit both shading and texture information while handling self-occlusion. Once our parametric hand model is defined and the synthesis process is specified, the problem amounts to estimating the hand pose parameters that generate a synthetic image that is the most similar to the observed image. Our similarity measure, referred to as the *objective function*, is a single term that comprises the sum of residual errors in the image domain. The use of a fine triangulated model allows for a good shading model. By explicitly modeling texture of the hand, we obtain a method that naturally captures the related information without the need to add new ad-hoc terms to the objective function.

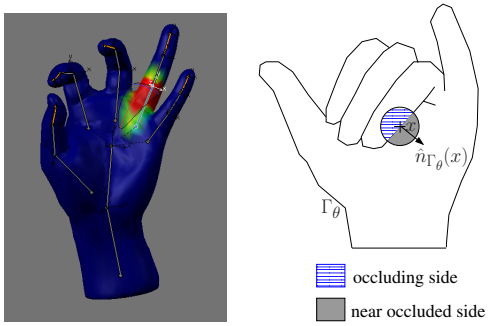
During the tracking process we determine, for each frame, the hand pose and illumination parameters by minimizing the objective function. The texture is updated after convergence in each frame and remains static while fitting the model to the next frame. In contrast with the optical flow approach proposed in [11], our objective function does not intrinsically assume limits on the range of displacements; it allows consequently large displacements and discontinuities. The optimal configuration is determined through a quasi-Newton descent using the exact gradient of the objective function. The adequate treatment of discontinuities along occlusion in the image domain, while deriving the functional gradient, provide the definition of new terms, that we refer to as *occlusion forces*. We test our method on challenging sequences involving large self-occlusion and out-of-plane rotations.

2. Image Synthesis and Objective Function

2.1. Synthesis

The analysis-by-synthesis approach we adopt requires the definition of the forward image synthesis process given a 3D hand pose, a surface texture and an illuminant. The formulation is derived from well-known *computer animation* and *computer graphics* concepts.

Following [4], we model the hand surface by a three dimensional, closed and orientable triangulated surface. The surface mesh consists of 1000 facets (Fig. 1.a). It is deformed according to pose changes of an underlying articulated skeleton using the pose space deformation technique [10]. The skeleton comprises 17 bones with 22 degrees of freedom (DOF). Each DOF corresponds to an articulation angle whose range is bounded to avoid unrealistic poses. The pose is fully determined by a vector θ that comprises the 22 articulation parameters and the 6 parameters that specify the global position and orientation of the palm with respect to the camera's coordinate frame.



(a) (b)

Figure 1. (a) The skinned hand model (b) The occlusion support Γ_θ and the local decomposition of the image domain near an occlusion point

We expect the sizes of the palm and fingers to change from one person to the next. Therefore, we defined 51 additional scaling parameters (3 per bone), called *morphological parameters*, to control those morphological variations. Those parameters are estimated during the calibration process explained in Section 4.1.

We assume a Lambertian reflectance model for the hand. This may be implemented in two ways. The first associates an RGB triplet to each vertex of the surface mesh, from which one can linearly interpolate over each mesh facet. This approach is conceptually simple but computationally inefficient as it requires many faces to obtain detailed reflectance information. The second approach, widely used in computer graphics, involves mapping an RGB reflectance (texture) image onto the surface. This technique guarantees the preservation of detail with a reasonably small number of faces.

In contrast with previous methods in the field of computer vision that use textured models (such as the morphable model [3]), our pose estimation formulation (Sec. 3) requires the texture reflectance to be continuously defined over the entire surface (i.e., seamless texture mapping). Using bilinear interpolation of the discretized texture we ensure continuity of the reflectance properties within each face. Because the hand is a closed surface, there is no simple way to ensure texture continuity over the entire surface. Following [18] and [9], we use patch-based texture mapping in conjunction with some linear equality constraints on the RGB values of the texture pixels lying along edges of patches. We refer the reader to [18, 9] for more details. We will denote by T the reflectance texture image and by $\mathbb{T}$ the linear subspace of valid textures i.e the textures whose RGB values satisfy the linear constraints that ensure reflectance continuity over the entire surface.

Since most hands have relatively little texture, shading has a major impact on hand appearance. We therefore incorporate illumination and shading (using the Gouraud shading algorithm) in our synthesis model. The illuminant is specified by a 4D vector denoted by L, comprising three ele-

ments for a directional component, and one for an ambient component. The irradiance at each vertex of the surface mesh is obtained by the scalar product between the homogenized surface normal at the vertex and this 4D vector. The irradiance across each face is then obtained through linear interpolation. Multiplying the reflectance and the irradiance yields the appearance for any point on the surface.

The synthetic image intensities for each point $\mathbf{x}$ in the 2D image plane are obtained in two steps. First, as in ray-tracing, we determine the first intersection between the triangulated surface mesh and the half-line (or ray) starting at the camera center and passing through $\mathbf{x}$. Second, the appearance of this intersection point is computed given the illuminant and the information at the vertices of the intersected face. If no intersection exists then the image is determined by the background. In practice, the image is computed on a discrete grid and the image synthesis can be done efficiently using the triangle rasterization technique in combination with a depth buffer.

If we also assume that we have the background model (e.g., an image for a stationary camera), then we have a fully specified synthesis process for the image of a hand. Let $i(\theta, L, T)$ denote the synthetic image comprising the hand and the background, for a given pose θ, texture T and illuminant L. We next consider the estimation problem.

2.2. Objective Function

Our main task is to recover the pose parameters θ, the texture T and illuminant L that produce a synthesized image that best matches the observed one, denoted by $i_{obs}()$. To specify the objective function we need to define a discrepancy measure between the synthesized and observed images. Here we obtain the discrepancy measure by summing the residual errors in the image domain, $\Omega \subset \mathbb{R}^2$; i.e.,

$$E(\theta, L, T) = \int_{\Omega} \rho\big(i(\theta, L, T, \mathbf{x}) - i_{obs}(\mathbf{x})\big) d\mathbf{x} \quad (1)$$

where ρ is either the classical squared error function (L_2 norm) or a robust error function such as the Huber function used in [17].

When defining the image synthesis process, the background image is assumed to be known. If the background is not static, we can relax this constraint by assuming that some color distribution d_{bk} associated to the background is available. We separate the integration domain Ω into the hypothesized hand and background region. In the background region we replace the term $\rho\big(i(\theta, L, T, \mathbf{x}) - i_{obs}(\mathbf{x})\big)$ by $-log\big(d_{bk}(i_{obs}(x))\big)$.

Our discrepancy measure presents certain advantages in comparison with more sophisticated ones that combine heterogeneous cues such as optical flow, silhouettes, or chamfer distances between detected and synthetic edges. First, this measure avoids the tuning of weights associated with different cues that is often problematic both in practice and in theory. This is due to the fact that a weighted sum of errors from different cues implicitly assumes (usually incorrectly) that errors in different cues are independent. Second, this measure avoids early decisions about the relevance of edges through thresholding, about the area of the silhouette by segmentation, or about the position of discontinuities in the optical flow. Third, this measure is a continuous function θ, while measures based on distances between edges like the symmetric chamfer distance usually cause discontinuities when an edge of one finger is suddenly occluded by another finger.

By a change of integration domain, the summation of the residual error within the hypothesized hand region can be re-expressed as a continuous integral over the visible part of the surface. It can then be approximated by a finite weighted sum over centers of all visible faces. Much of the literature on 3D deformable models take this approach, and assume the visibility of each face to be a binary state that can be obtained from a depth buffer (e.g., see [3]). Unfortunately, such a discretization introduces discontinuities in the approximate functional when θ varies. When the surface moves or deforms, the binary visibility state of a face near self-collusion is likely to switch between 0 and 1. This will cause a discontinuity in the sum of residuals. Such discontinuities are undesirable if one hopes to use gradient-based optimization methods. In order to preserve continuity of the discretized functional with respect to θ, the visibility state should not be binary. Rather, it should be real-valued between zero and one, and should behave smoothly as the surface becomes occluded or unoccluded. In practice, this appears rather cumbersome to implement and the derivation of the functional gradient may be complicated. To address this continuity problem, in contrast with much of the literature on 3D deformable models, we keep the formulation in the continuous image domain when deriving the expression of functional gradient.

In order to estimate the pose θ and the lighting condition L for each new incoming frame, or to update the texture T, we look for the lowest potential of the objective function. During tracking we either minimize the functional with respect to θ and L in order to register the model onto a new frame, or minimize with respect to T to update the texture once the model has been fitted. This alternate minimization can be interpreted as form of coordinate descent limited to a single iteration per frame. We will first consider the problem of estimating θ and L given a new frame as it constitutes the core of the tracking problem.

3. Pose and Lighting Estimation

The simultaneous estimation of the pose parameters θ and the illuminant L is a challenging non-linear and non-convex optimization problem. The dimensionality of θ

therefore makes global optimization unattainable and we resort to an efficient quasi-Newton, iterative local optimization. This method requires the analytical computation of the functional gradient with respect to θ and L. Because of the discontinuities along occlusions, the derivation of the functional gradient with respect to θ constitutes the main challenge of our approach and is therefore the focus of the next section.

3.1. Gradient w.r.t. Pose and Lighting

In the synthesis process we carefully designed the reflectance and the irradiance to be continuous on the hand surface (using Gouraud shading and patch-based texture mapping with bilinear interpolation). We further assume the background image to be known and continuous. We consider the image domain Ω to be a continuous subset of $\mathbb{R}^2$. So the synthesized image is spatially continuous almost everywhere, with the exception of the set of points corresponding to occlusion and self-occlusion boundaries; i.e., those points where the hand starts occluding the background or other parts of the hand (see Fig. 1.b).

Let Γ_θ be the set of boundary points. One can interpret Γ_θ to be the support of depth discontinuities in the image domain. Because we are working with a triangulated surface mesh, Γ_θ can be decomposed into a number of linear segments. More precisely, Γ_θ corresponds to the union of the projections of all visible portions of edges of the triangulated surface that separate a front-facing face from a back-facing face. For any point $\mathbf{x}$ along Γ_θ, the corresponding edge projection locally separates the image domain into two subregions of $\Omega\backslash\Gamma_\theta$, which we refer to as the *near-occluded* side and the *occluding* side.

We will denote by $\bar{V}_k$ the projection of the k^{th} vertex of the hand mesh in the image. For any point $\mathbf{x}$ on the self-occlusion boundary Γ_θ, there exist two vertices indexed by $m_\mathbf{x}$ and $n_\mathbf{x}$ such that $\mathbf{x}$ belongs to the image segment joining the projected vertices $\bar{V}_{m_\mathbf{x}}$ and $\bar{V}_{n_\mathbf{x}}$. We define $t_\mathbf{x} \in [0,1]$ to be the scalar value such that $\mathbf{x} = (1-t_\mathbf{x})\bar{V}_{m_\mathbf{x}} + t_\mathbf{x}\bar{V}_{n_\mathbf{x}}$. We also obtain the 2D normal vector to the corresponding edge segment by:

$$\hat{n}_{\Gamma_\theta}(\mathbf{x}) = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \frac{\bar{V}_{n_\mathbf{x}}(\theta) - \bar{V}_{m_\mathbf{x}}(\theta)}{\|\bar{V}_{n_\mathbf{x}} - \bar{V}_{m_\mathbf{x}}\|} \tag{2}$$

Here we assume $n_\mathbf{x}$ and $m_\mathbf{x}$ to be ordered such that the 2D normals are oriented outward, toward the *near-occluded* side (see Fig. 1.b). For a given pose parameter, θ_j in θ, we can define the 2D derivative of the curve Γ_θ

$$v_j(\mathbf{x}) = (1-t_\mathbf{x})\frac{\partial \bar{V}_{m_\mathbf{x}}}{\partial \theta_j} + t_\mathbf{x}\frac{\partial \bar{V}_{n_\mathbf{x}}}{\partial \theta_j} \tag{3}$$

Along Γ_θ, the synthetic image $i()$ is discontinuous both with respect to $\mathbf{x}$ and θ. Thus $\frac{\partial i(\theta,\mathbf{x})}{\partial \theta_j}$ is not defined along Γ_θ. To deal with such discontinuities, we introduce a new image $i^+(\theta,\mathbf{x})$ that extends $i()$ by continuity in Γ_θ, while replicating the content in $\Omega\backslash\Gamma_\theta$ of $i()$ from the *near-occluded* side. By forming i^+, we are recovering the RGB values of the *occluded faces* in the vicinity of the occluding boundary points $\mathbf{x}$. This can be done using a infinite sequence of points that approaches $\mathbf{x}$ from the *near-occluded* side:

$$i^+(\theta,\mathbf{x}) = \lim_{k\to\infty} \big(i(\theta,\, \mathbf{x}+\hat{n}_{\Gamma_\theta}(\mathbf{x})/k)\big) \tag{4}$$

One can also interpret the RGB color of $i^+(\theta,\mathbf{x})$ along Γ_θ as the appearance of the second intersection point of the surface S_θ along the ray starting at the camera center and passing through $\mathbf{x}$.

When the parameter θ_j is increased with an infinitesimal step $d\theta_j$, the contour of occlusion Γ_θ around a point $\mathbf{x} \in \Gamma_\theta$ moves with an infinitesimal step $v_i(\mathbf{x})d\theta$. The image intensities in the occluded vicinity of $\mathbf{x}$ vary in a discontinuous manner; i.e., they jump between $i^+(\theta,\mathbf{x})$ and $i(\theta,\mathbf{x})$. The residual error at $\mathbf{x}$ therefore jumps between $\rho\big(i^+(\theta,\mathbf{x}) - i_{obs}(\mathbf{x})\big)$ and $\rho\big(i(\theta,\mathbf{x}) - i_{obs}(\mathbf{x})\big)$. However, because the surface area where this *jump* occurs is also infinitesimal and proportional to $\big(v_j(\mathbf{x}).\hat{n}_{\Gamma_\theta}(\mathbf{x})\big)d\theta_j$, this induces only an infinitesimal change in the objective functional after integration on the continuous image domain Ω. One can therefore formally derive the exact functional gradient $\nabla_\theta E \equiv (\frac{\partial E}{\partial \theta_1}, \ldots, \frac{\partial E}{\partial \theta_n})$ by decomposing the image support Γ_θ into two components as follows:

$$\begin{aligned} \frac{\partial E}{\partial \theta_j} = &\int_{\Gamma^\theta} f_{oc}(\mathbf{x}) v_i(\mathbf{x}) d\mathbf{x} \\ &+ \int_{\Omega\backslash\Gamma_\theta} \big[\rho'\big(i(\theta,\mathbf{x}) - i_{obs}(\mathbf{x})\big)\frac{\partial i(\theta,\mathbf{x})}{\partial \theta_j}\big] d\mathbf{x} \end{aligned} \tag{5}$$

where f_{oc}, referred to as the *occlusion force*, is defined by

$$\begin{aligned} f_{oc} :&\Gamma_\theta \to \mathbb{R}^2 \\ f_{oc}(\mathbf{x}) =&\big[\rho\big(i^+(\theta,\mathbf{x}) - i_{obs}(\mathbf{x})\big) \\ &- \rho\big(i(\theta,\mathbf{x}) - i_{obs}(\mathbf{x})\big)\big]\hat{n}_{\Gamma_\theta}(\mathbf{x}) \end{aligned} \tag{6}$$

These *occlusion forces* account for the displacement of occlusion and self-occlusion contours when θ varies. These forces are novel while bearing certain similarities with the forces obtained in the context of 2D active regions [13]. The force directions are normal to the curve, and their norms are proportional to the difference of the local cost at each side of the curve. This similarity with 2D active regions derives from the fact that we kept the image domain Ω continuous while computing the functional gradient.

Because the surface is triangulated, Γ_θ can be decomposed into a set of image line segments and we can rewrite expressions similar to the ones reported in [22] for active polygons. The analytical derivation of $\partial i(\theta,\mathbf{x})/\partial\theta_j$, as

well as $\partial E/\partial L$, simply requires application of the chain rule to the synthesis process. Using a backward ordering of derivative multiplications (called adjoint coding in the algorithm differentiation literature [7]), we obtain an inexpensive evaluation of the objective function gradient. The computational cost of evaluating the objective function and its gradient are comparable.

When deriving the functional gradient we assumed the background image to be known. If the background is not static, we replace the term $\rho\big(i^+(\theta,\mathbf{x}) - i_{obs}(\mathbf{x})\big)$ by $-log\big(d_{bk}(i_{obs}(x))\big)$ in the expression of occlusion forces along all edges corresponding to background occlusion (i.e the hand silhouette). This approach has successfully handled videos with moving backgrounds (see Fig. 5).

3.2. Model Registration

During the tracking process, the model is registered to each new frame by minimizing the objective function E with respect to the pose θ and the illuminant L. Because we can analytically specify the gradient of the objective function, the non-linear minimization of (1) is done efficiently using a sequential quadratic programming method (SQP) [5] with a adapted Broyden-Fletcher-Goldfarb-Shanno (BFGS) approximation of the Hessian. This method allows us to combine the well-known BFGS quasi-Newton method with the linear constraints that limit the range of the articulation angles of the hand pose. We slightly adapted the BFGS update method to take advantage of the partial independence of separated fingers. With such independence, some sparseness of the Hessian can be exploited. We adapted the BFGS update method to take advantage of this structure, which increased the convergence rate by a factor ranging from 3 to 6.

Because our optimization method is local, we need to initialize the search. To obtain reasonable starting points we first perform a one-dimensional search on the line that linearly extrapolates the two previous poses. Each local minima obtained during this 1D search is used as a starting point for our SQP method in the full pose space. The number of starting points found is usually just one, but when there are several local minima, the results of the SQP method are compared and the best solution is naturally chosen.

Once the model has been fit to a new image frame using this optimization method, we update the texture model that is used for registration with the next frame.

3.3. Texture Update

Various methods for mapping images onto a static 3D surface exist. Perhaps the simplest method involves, for each 3D surface point, (1) computing its projected coordinates in the image plane, (2) checking its visibility by comparing its depth with the depth buffer at those image coordinates, and (3) if the point is visible, interpolating image intensities at those coordinates and then recovering the reflectance by dividing the interpolated intensity by the model irradiance at that point. This approach is not suitable for our formulation for several reasons. First, we need to interpolate values in hidden parts of the hand, as those regions may become visible in the next frame. Second, we need to update the texture robustly to avoid progressive contamination by the background color, or self-contamination between different parts of the hand.

Here, we formulate texture estimation as the minimization of the same objective functional as that used for tracking, in combination with a smoothness regularization term. That is, for texture T we minimize

$$E_{texture}(T) = E(\theta, l, T) + \beta E_{sm}(T)\ . \tag{7}$$

For simplicity the smoothness measure is defined to be the sum of squared differences between adjacent pixels in the texture (texels). That is,

$$E_{sm}(T) = \sum_i \sum_{j\in\mathcal{N}_T(i)} \|T_i - T_j\|^2 \tag{8}$$

where $\mathcal{N}_T(i)$ represent the neighborhood of the texel i. If one chooses ρ to be the Huber function or a truncated quadratic function in (1), then the minimization of the function $E_{texture}(T)$ can be done efficiently using a conventional, iteratively reweighed least-square approach. The truncated quadratic function gave the best results.

Due to the smoothness term, the color is diffused to the texels that do not contribute to the image i, either because they are associated to a part that is hidden or because of texture aliasing artifacts. To improve robustness we also remove pixels near the occlusion boundary from the integration domain Ω when computing the term $E(\theta, l, T)$, and we bound the difference between the texture in the first frame and the subsequent texture estimates.

Cast shadows are not modeled in our approach, and are therefore back-projected into the texture as any other color information. Introducing cast shadows in our continuous optimization framework is not straightforward as it would require computation of related terms in the functional gradient. Shadows would then constitute an additional source of information to guide the registration of the model. Nevertheless, despite lack of cast shadows in our model, our results show adequate, robust tracking.

4. Experimental Results

4.1. Initialization

The hand pose tracker requires a reliable initial guess in the first frame. Estimating the hand pose in a single frame without a strong prior of the hand pose is challenging. In our case the morphological parameters are also estimated in

the first frame. However, since the pose estimation problem is ill-posed itself, adding these additional parameters makes the process even more difficult. One could attempt to use a discriminative method to obtain a rough initialization (e.g., [15]). However, since this is outside the scope of our approach at present, we assume prior information about the initial hand pose.

In particular, the hand is assumed to be parallel to the image plane at initialization and linear constraints were defined on relative length of the parts within each finger. Furthermore, since we do not yet have a texture estimate in the first frame, we suppose the hand color to be constant across the surface. With this assumption the appearance of the hand is largely due to shading. The three RGB values of the hand color, along with the hand pose, the morphological parameters and the illuminant are estimated simultaneously using the quasi-Newton method. We suppose that the background image or its histogram are provided by the user. Once the morphological parameters are estimated in the first frame, they remain fixed for the remainder of the image sequence.

4.2. Tracking

We test our tracking algorithm on various image sequences. In the first sequence (Fig. 2) each finger bends in sequential order, eventually occluding parts of other fingers and the palm. The cluttered background image is static, and was obtained from a frame where the hand was not visible. The resolution of each frame is 640 by 480 pixels.

To illustrate the improvements provided by self-occlusion forces, we could simply remove the occlusion-forces while computing the functional gradient. The effect is dramatic as the resulting algorithm is unable track any displacement. The comparisons with the "conventional" sum-on-surface approach outlined in Section 2.2 is more relevant. This alternative approach involves summing errors on 3D points that remain fixed on the triangulated surface throughout the iterations. Those points are uniformly distributed on the hand surface and their binary visibility is computed at each step of the quasi-Newton minimization process. To account for the change of summation domain (from image to the surface), the error associated to each point is weighted by the inverse of the ratio of surfaces between the 3D triangular face and its projection. The errors associated with the background are also taken into account, in order to remain consistent with the initial cost function. Furthermore, this will prohibit the model from shrinking in the image domain by increasing its distance to the camera. We kept occlusion forces between the hand and the background to account for variations of the background visibility while computing the functional gradient. The functional computed in both approaches would ideally be equal and their difference is bounded by the error induced by the dis-

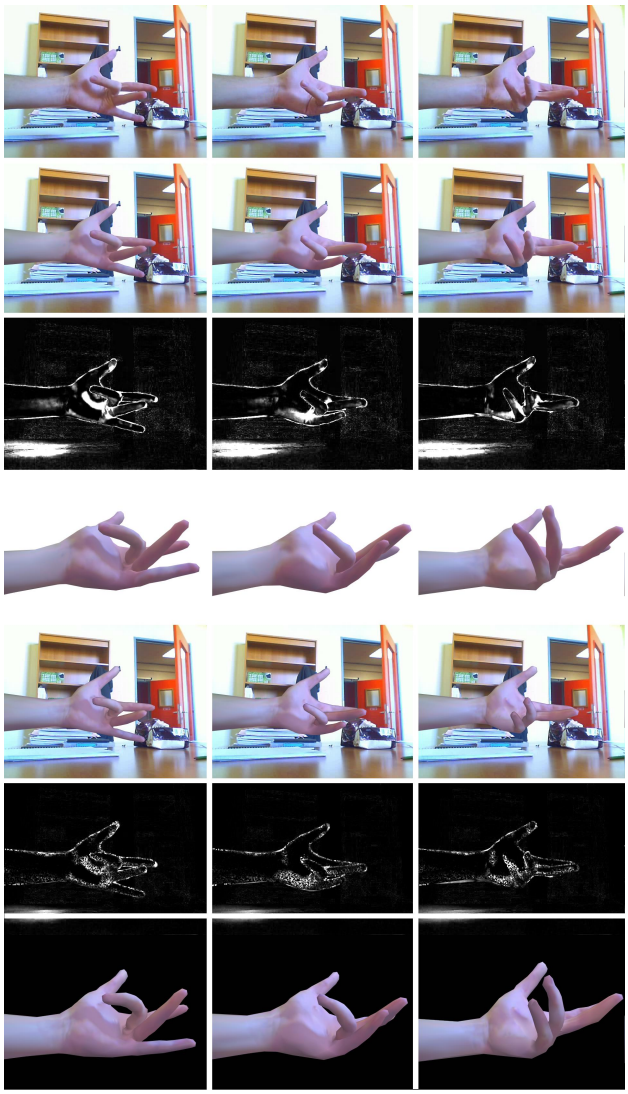


Figure 2. First sequence illustrating improvement due to self-occlusion forces. Each row corresponds in order to : the observed image, the final synthetic image, the final residual image, the synthetic side view with 45°, the final synthetic image with residual summed on surface, the residual for visible points on the surface, the synthetic side view

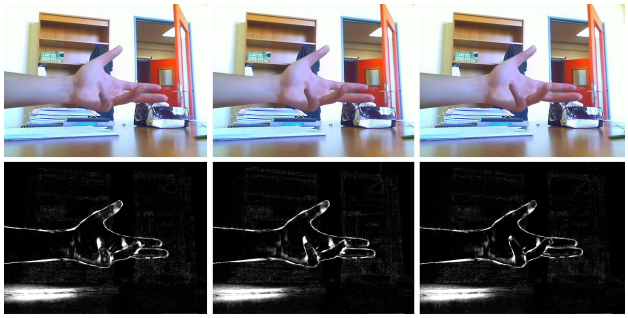


Figure 3. Recovery of the failure mode of the sum-on-surface method (section 4.2) by our method with occlusion force

Figure 4. Second sequence. Each row corresponds in order to : the observed image, the final synthetic image with limited pose space, the final residual image, the synthetic side view with an angle of $45°$, the final synthetic image with full pose space, the residual image, the synthetic side view

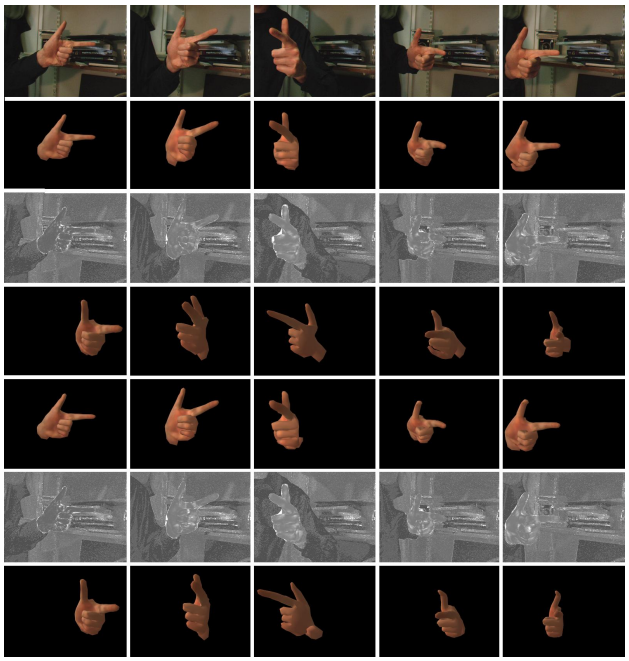


Figure 5. Third sequence. Each row corresponds in order to : the observed image, the final synthetic image with limited pose space, the final residual image, the synthetic side view with an angle of $45°$, the final synthetic image with full pose space, the residual image, the synthetic side view

cretization of integrals. For both methods we limited the number of iterations to 100.

This alternative approach produces overall good results (Fig. 2 rows 5-7) but fails to recover the precise location of the finger extremities when they bend and occlude the palm. This is most significant in the third column where a large portion of the little finger is missing. Our approach (rows 2-5) compares favorably, yielding accurate tracking through the entire sequence. The alternative approach fails because the hand/background silhouette is not particularly informative about the position of fingertips when fingers are bending. The residual error is mostly localized near the outside extremity of the synthesized finger, and self-occlusion forces are necessary to pull the finger toward this region.

We further validate our approach by choosing the erroneous estimated hand pose in the third column as an initialization of the new tracking method in this paper that uses the occlusion forces (Fig. 3). After a few frames, the hand pose is properly recovered. This illustrates the eventual inability of the alternative approach to converge to a local minima of the cost function as a consequence of its poor treatment of occlusions.

The second and third sequences (Fig. 4 and Fig. 5) were provided by the authors of the tree-based Monte Carlo method that constitutes the state-of-the-art in monocular hand tracking [19]. Both sequences have a resolution of 320 by 240 pixels. In the second sequence the hand is closing and opening while rotating. In the third sequence the index finger is pointing and the hand rotates in a rigid manner. Both sequences present important self-occlusion and large inter-frame displacements. The background in the third sequence is not static and the associated cost has been expressed using a histogram (see Sec. 2.2). For computational reasons, the results presented in [19] 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 (respectively rows 2 and 3). To do so, linear inequalities were defined between pairs or triplet 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. We did not update the texture for those sequences after the first frame. As one would expect the results are better when the pose space is reduced. Nevertheless, the results obtained with the full pose space are still satisfying. The loss of accuracy during tracking in the second and third sequences, in comparison with the first sequence, can be attributed to two key factors. The inter frame movement in the second and third sequences is large. This challenges our local search approach as a starting point for the minimization (i.e., it has

to be predicted from previous estimated poses). Also, in the second and third sequences the fingers are often touching each other. This challenges our method because collision avoidance has not been incorporated to prohibit parts from penetrating one another in our optimization framework.

5. Discussion

This paper describes a novel approach to the recovery of geometric and photometric pose parameters of a 3D model from monocular image sequences. Our approach introduces a complete model where geometry is integrated with texture (scene + object), shading, lighting, and with the proper handling of occlusions. These parameters are determined through a variational formulation where a rigorous mathematical model is considered to deal with discontinuities of the cost function and provide the complete and correct objective function gradient. To demonstrate the potential of our approach, we considered hand pose estimation using a dynamic articulated model with 28 degrees of freedom, where texture is learned/updated from the images.

Collision handling between parts of the articulated object will be addressed in the near future in order to prevent implausible pose estimates. Modeling cast shadows could improve results as it would introduce additional information during the registration process. Furthermore, exploring the temporal dynamics (currently only to determine an initial guess) would be a useful way to address convergence to local minima. Last, but not least, imposing prior knowledge on plausible configurations through the definition of prior manifolds is an interesting but rather challenging direction from theoretical and technical perspectives.

References

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

[2] A. Balan, M. Black, H. Haussecker, and L. Sigal. Shining a light on human pose: On shadows, shading and the estimation of pose and shape. In *ICCV*, pages 1–8, 2007.

[3] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Siggraph*, pages 187–194, 1999.

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

[5] A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust-region methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

[6] M. de la Gorce and N. Paragios. Monocular hand pose estimation using variable metric gradient-descent. In *BMVC*, page III:1269, 2006.

[7] A. Griewank. Evaluating derivatives: principles and techniques of algorithmic differentiation. 2000.

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

[9] C. Hernández. *Stereo and Silhouette Fusion for 3D Object Modeling from Uncalibrated Images Under Circular Motion*. PhD thesis, ENST, May 2004.

[10] J. P. Lewis, M. Cordner, and N. Fong. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *Siggraph*, pages 165–172, 2000.

[11] 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.

[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] N. Paragios and R. Deriche. Geodesic active regions for supervised texture segmentation. In *ICCV (2)*, pages II:926–932, 1999.

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

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

[16] N. Shimada. Real-time 3-d hand posture estimation based on 2-d appearance retrieval using monocular camera. In *RATFG*, page 23, 2001.

[17] H. Sidenbladh, M. J. Black, and D. J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *ECCV*, pages II:702–718, 2000.

[18] M. Soucy, G. Godin, and M. Rioux. A texture-mapping approach for the compression of colored 3d triangulations. *The Visual Computer*, 12(10):503–514, 1996.

[19] B. Stenger. *Model-Based Hand Tracking Using A Hierarchical Bayesian Filter*. PhD thesis, University of Cambridge, Cambridge, UK, March 2004.

[20] B. Stenger, P. R. S. Mendonça, and R. Cipolla. Model-based hand tracking using an unscented kalman filter. In *BMVC*, pages I:63–72, 2001.

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

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

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