Int J Comput Vis (2008) 78: 107–118
DOI 10.1007/s11263-007-0083-7

Mutual Information-Based 3D Object Tracking

Giorgio Panin · Alois Knoll

Received: 6 December 2006 / Accepted: 14 August 2007 / Published online: 10 October 2007

Abstract We propose a robust methodology for 3D model-based markerless tracking of textured objects in monocular image sequences. The technique is based on mutual information maximization, a widely known criterion for multimodal image registration, and employs an efficient multiresolution strategy in order to achieve robustness while keeping fast computational time, thus achieving near real-time performance for visual tracking of complex textured surfaces.

Keywords Surface-image alignment · Mutual information · Nonlinear optimization · B-Spline interpolation · Multiresolution · 3D tracking · Template matching

1 Introduction

The present work deals with a robust methodology for template-based markerless object tracking in 3D applications. In order to motivate our approach, we examine here related methodologies from the current available literature on the subject.

Several model-based techniques have been developed for the purpose of pose estimation in monocular sequences. A pose estimation methodology generally requires mapping a more or less large set of distinctive elements (features) from the model surface to the current image, which can be selected and stored in advance, or refined during the tracking process itself. To this respect, we make here a main distinction between keypoints (small-dimension, distinctive intensity patterns), lines and small geometrical shapes, which we call *local* features, as opposed to *global* features describing large areas of the object surface, up to the whole appearance directly mapped onto the 3D shape.

Concerning keypoint-based techniques, we can furtherly distinguish (Vacchetti and Lepetit 2004) between off-line detection and on-line features tracking: detection requires selecting a set of invariant keypoints, both from the current image and from one or more reference views of the object, and afterwards matching them pairwise; on-line tracking instead updates the current keypoint localization by using the previous frame result, updating as well the overall feature descriptor (appearance).

In the first case, in order to achieve robustness with respect to unpredictable light/shading situations, as well as noise and partial occlusions, invariant keypoints are carefully selected from the model surface, and robustly localized into the current image by using appropriate description and matching criteria. This approach has the advantage of providing an independent frame-by-frame pose estimation, which therefore does not suffer from error accumulation or drift problems; on the other hand, for the very same reason this methodology does not take any advantage from the previous estimation result, therefore showing both a lower precision and speed. One of the most popular keypoint detection techniques is currently the scale-invariant features transform (SIFT) (Lowe 2004); it runs in near real-time on common platforms (in our experiments we observe a detec-

Electronic supplementary material The online version of this article (http://dx.doi.org/10.1007/s11263-007-0083-7) contains supplementary material, which is available to authorized users.

G. Panin (✉) · A. Knoll
Chair for Robotics and Embedded Systems, Technical University of Munich, Boltzmannstrasse 3, 85748 Garching bei Muenchen, Germany
e-mail: panin@in.tum.de

A. Knoll
e-mail: knoll@in.tum.de

tion rate of about 2 fps), and it has been applied by the same Authors to 3D object tracking (Skrypnyk and Lowe 2004).

A main bottleneck for keypoint detection is the possible presence of false identifications (outliers), that likely lead to a less precise and stable pose estimation (*jitter* effect), even in presence of a few wrong detections. After obtaining candidate matchings, robust statistics methods like RANSAC (Fischler and Bolles 1981) try therefore to enforce *a posteriori* the 3D model geometric constraints in order to remove outliers, but they cannot guarantee a stable result unless the reliable features subset is large enough. M-estimators (Huber 1981) are another common choice for this problem, that amounts to reducing the influence of outliers, by selecting a proper robust cost function and the outlier threshold.

On-line keypoint tracking (Shi and Tomasi 1994) is instead based on local, frame-to-frame optimization, therefore obtaining a higher speed and a better localization, as long as the local appearance is not too quickly changing because of lighting, or the feature gets temporarily occluded; for this reason, an independent drift detection mechanism must be provided (Shi and Tomasi 1994).

In order to take advantage from both approaches, Vacchetti and Lepetit (2004) combines off-line and on-line information by matching local features both from the reference views and the previous frame, through a combined, robust least-squares optimization. This approach can obtain a better stability, precision and speed for different object models; nevertheless, the tradeoff between the two modalities may pose further implementation issues: off-line information needs generating and updating invariant reference keypoints, as the object viewpoint and lighting will change over time, while on the other hand on-line tracking requires correctly mapping new detected keypoints back to the model surface; these and other related issues have to be carefully considered for the overall system design, in order to avoid error accumulation and drift (see Vacchetti and Lepetit 2004 for more details).

We mention here some 3D face tracking applications developed using keypoint matching and tracking. The system proposed in Toyama (1998) employs a robust multi-layer fusion of different visual cues in the hierarchical framework IFA (Toyama and Hager 1996), proceeding from coarse to accurate visual modalities, and providing the result from the top-level tracker as output; in this work, simple template and feature point models are used at the higher levels. Natural keypoints have also been used in Gorodnichy et al. (2002) in order to obtain 3D face tracking in a stereo framework; the system requires off-line calibration of the stereo rig using multiple point correspondences under epipolar constraints, plus an initial learning of additional face features, for tracking a given user. In Xiang-Tian (2004), a 3D face model is fitted by matching features across subsequent frames, with an approach combining RANSAC and Particle Filters under frame-to-frame epipolar constraints.

A rather challenging situation for keypoint-based tracking happens when the surface has an overall distinctive texture pattern, but a few distinctive local keypoints. This is usually the case for face tracking, where the only reliable keypoints are usually found on the eyebrows, nostrils, eye and mouth corners, and a few other locations, whereas the overall textured shape as a whole could be well localized in space.

Another common difficulty arises when the object surface shows a significant curvature, since a basic assumption is that a local keypoint can be represented by a small, planar image patch, undergoing an approximately affine transformation without significant nonlinear distortion. For this reason, non-planar surfaces usually show less reliable keypoints from any given viewpoint, thus requiring more reference views for a reliable tracking (Lowe 2004).

By summarizing, we see how local features have the advantage of being small and allowing, with a careful implementation, an independent identification and tracking over image sequences; but they suffer from the robustness point of view, whenever the amount of reliable detections is insufficient—generally speaking, because of the little information on the object appearance contained into a single local pattern (*descriptor*).

A global template-based approach, instead, attempts to directly exploit the whole model information available, so that a more general class of textured objects can be tracked, and at the same time more precise and stable 3D localizations can be obtained (Hager and Belhumeur 1998; Black and Jepson 1996).

In order to estimate the pose, template-based techniques optimize on-line a similarity measure, which usually is a standard SSD (Sum of Squared Differences) between model and image correspondent color or intensity pixels at a given pose hypothesis (Baker and Matthews 2004). Template tracking therefore amounts to solve a single large, nonlinear LSE problem, which can as well be formulated in more or less robust ways using M-Estimators.

However, a further difficulty here arises because of possibly complex and unpredictable light shading patterns, which usually are coped with by combining multiple appearance models Cootes et al. (1998) and augmenting as well the object state-space.

Several face tracking applications in this area can be mentioned. In Matthews and Baker (2003), shape and appearance parameters are optimized at the same time under a 2D piece-wise affine deformation model; although the 3D head pose is not directly provided by the estimation algorithm, it can subsequently be estimated from the set of planar parameters, at the price of more complex computations involving an Extended Kalman Filter (Xiao et al. 2004). The work (Cascia et al. 1999), closer to ours, directly employs a full 3D template, with multiple lighting models and M-estimators for robust optimization; a full 6*dof* evaluation of pose parameters is also provided, and compared with ground-truth data obtained through a magnetic sensor device. The head shape of a given user, together with the texture models taken under different light conditions, need to be off-line provided.

In this paper we propose a template tracking technique making use of efficient information-based optimization. The paper is organized as follows: Sect. 2 provides an outline and motivation of the present approach; Sect. 3 describes the geometric framework, and Sects. 4 and 5 the overall pose estimation algorithm; Sect. 6 reports experimental results, showing tracking performance over simulated and real sequences, together with ground-truth comparisons with other approaches. Conclusions and planned improvements over the current implementation are given at the end.

2 Motivation and Scope of the Present Work

The need for multiple appearance models can constitute a major drawback of template tracking, from one side requiring the user to provide off-line several training images, and from the other side needing to augment the state dimensionality in a complex way.

This complication is absent in keypoint-based techniques, where each local window approximately undergoes an overall brightness/contrast change, which can be coped with in simpler ways. For a large surface template this approximation is not sufficient anymore, because observed brightness changes can be distributed along the surface in a nonlinear and unpredictable way.

An alternative idea could be to on-line update the single appearance model at each frame, but (as also pointed out in Cascia et al. 1999) this procedure easily leads to error accumulation, and ultimately biases the tracker towards incorrect matching results.

It would of course be still desirable to use a few, possibly just one, reference appearance for tracking; towards this goal, therefore, standard SSD cannot be used as similarity measure, as well as robust versions like as M-Estimators, which can well deal with individual outliers but not with an overall shading variation.

A more general similarity measure can be provided by the Normalized Cross-Correlation (NCC) index (Duda and Hart 1973; Gonzalez and Woods 2006), which is employed with good results for keypoint-based pose estimation (Vacchetti and Lepetit 2004) and pattern recognition (Brunelli and Poggio 1993). However, NCC still assumes the relationship between model and underlying image appearance to be of a linear nature, which is only a locally valid model.

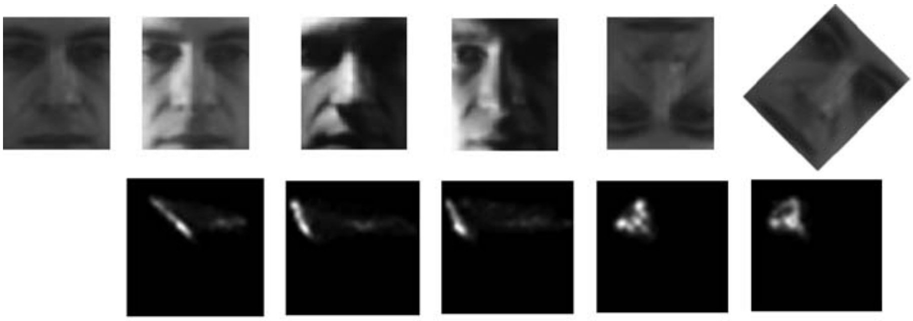
Nevertheless, under a few assumptions about the surface reflectance properties (absence of strong specularities or transparent areas), when model and image are correctly aligned, the corresponding intensity patterns show a relationship, i.e. a statistical *dependence*, also under different light conditions; this dependence becomes clearly weaker, or completely absent, for a misaligned pose. This fact can be observed by looking at the sparseness of the co-occurrence matrix of corresponding pixel intensities, (Fig. 1) which represents the joint probability histogram.

From a statistical point of view, the amount of dependence between two variables can be effectively and reliably measured by using Mutual Information (Cover and Thomas 1991). Maximizing MI between grey-level images is a powerful and widely acknowledged principle for multi-modal alignment in the medical imaging field, since the seminal work (Wells et al. 1996) and the almost contemporary one (Maes et al. 1997). The same principle has been subsequently applied in a more general machine learning context (Principe et al. 1999).

As the example shows, using mutual information as consistency measure allows accommodating a more general model of variation between expected and observed light shading; the same can be observed in presence of noise and partial occlusions (outliers). This holds the potential of a more robust and stable trajectory tracking, whether the stability is considered with respect to any of the mentioned factors. And at the same time, complexity of model building can be kept to a minimal extent, in the ideal case using a single reference texture under an almost arbitrary environment light.

A well-known bottleneck of MI maximization can be a higher computational complexity, which requires a careful balance of model and image resolution, sample size, memory and timing requirements, and the choice of a suitable multi-dimensional optimization algorithm. As the medical image registration literature shows (Pluim et al. 2003), MI has a big impact in fields where speed is a relatively secondary issue, while robustness and precision are critical requirements.

Towards a fast and reliable visual pose estimation, choice of the optimization method can be very important and problem-dependent; in particular, for our purpose of 3D-2D alignment, simple derivative-free methods (Nelder and Mead 1965) or first-order gradient descent (Wells et al. 1996) are not well-suited, since the function level sets of any similarity function show a rather variable behavior along different directions in 6-pose space (e.g. depth vs. planar roto-translations). In the LSE literature, this problem is normally approached by using a robust Levenberg–Marquardt strategy (Marquardt 1963), with a first-order approximation of the Hessian matrix given by the Gauss-Newton matrix (Baker and Matthews 2004). An extension of this strategy to MI for medical image registration has been proposed in Thevenaz and Unser (2000).



SSD	40.5226	58.6129	66.5071	27.3357	25.3888
NCC	0.4365	0.3746	0.0540	-0.2037	0.2817
MI	0.6942	0.4384	0.3591	0.2466	0.1616

Fig. 1 A comparison between three similarity measures for template matching in presence of light and shading variations. *Top row*: a face template taken from (Hager and Belhumeur 1998) (*left*) with a set of different views under light and pose variations; *Second row*: gray level co-occurrence matrices between template and images (joint histograms); *Bottom*: corresponding SSD, NCC and Mutual Information similarity measures. Standard SSD is not robust to light changes, while NCC can only model linear relationships between template and image; a stronger dependence, although non linear, can be observed when the pose is correct also under different lighting, for which MI shows to be a more reliable and smooth similarity index

Taking inspiration from this idea, in the present work we propose a smooth MI maximization method for template tracking in video sequences. For this purpose, we employ multiresolution with Gaussian filtering, fast B-Spline image interpolation (Unser et al. 1993), fuzzy histograms, and Levenberg–Marquardt optimization.

In this framework, quadratic B-Splines are used both for smooth image interpolation and kernel-based histogram binning (Sect. 4); this choice, as also pointed out in Unser (1999), has several advantages over other kernel-based techniques for many image processing and registration problems. First, a B-Spline kernel satisfies the *partition of unity* condition (Thevenaz and Unser 2000), that ensures independence of the constant template distribution on the pose parameters, when computed by marginalization from the joint histogram (see (23)); several other desirable properties from approximation and sampling theory are satisfied as well by B-Splines (Unser et al. 1993). Second, at the image level, computation of spline interpolation coefficients can be very efficiently done through recursive filtering (Unser 1999; Unser et al. 1993). Third, when using a pyramidal multiresolution approach, spline coefficients for the whole pyramid can as well be efficiently derived from the base image coefficients (Unser et al. 1993). And finally, differentiability of the kernel provides exact Jacobian matrices, which are obtained with almost the same computational complexity of the MI function itself.

With this methodology we obtain a precise, robust and relatively fast MI optimization over the 6-pose parameters in full projective space, for template-based object tracking tasks.

3 Template Modeling Framework

3.1 World Geometry Representation

We express the rigid transformation between camera and object coordinate frames with the homogeneous (4×4) transformation matrix T

$$T = \begin{bmatrix} R & \theta_T \\ \mathbf{0} & 1 \end{bmatrix} \tag{1}$$

with R the rotation matrix, and $\theta_T = [X, Y, Z]^T$ the translation vector. 3D rotations are expressed in terms of *XYZ* Euler angles

$$\begin{aligned} &\theta_R = [\alpha, \beta, \gamma]^T, \\ &R(\theta_R) = R_x(\alpha) R_y(\beta) R_z(\gamma) \end{aligned} \tag{2}$$

and the overall object pose is given by the 6-vector θ

$$\theta = [\theta_R, \theta_T]. \tag{3}$$

In order to avoid representation singularities, we refer the transformation matrix T_t at time t to the last estimated value $T_{\mathrm{ref}} = T_{t-1}$, so that

$$T_t(\theta) = \delta T(\theta) T_{t-1} \tag{4}$$

where $\delta T(\theta)$ is computed according to (1). A body point in homogeneous coordinates ${}^b\bar{\mathbf{x}}$ therefore transforms to camera coordinates ${}^c\bar{\mathbf{x}}$ according to

$${}^c\bar{\mathbf{x}} = \delta T(\theta) T^b_{t-1} \bar{\mathbf{x}}. \tag{5}$$

Concerning intrinsic camera parameters, we adopt a simple pinhole model with focal length F, so that points in camera space ${}^{c}\mathbf{x} = [x_{1c}, x_{2c}, x_{3c}]^T$ project to the screen $\mathbf{y} = [y_1, y_2]^T$ as

$$y_1 = F\frac{x_{1c}}{x_{3c}} + \frac{r_{y1}}{2}; \qquad y_2 = -F\frac{x_{2c}}{x_{3c}} + \frac{r_{y2}}{2} \tag{6}$$

with r_{y1}, r_{y2} the horizontal and vertical image resolution. By assuming camera calibration to be off-line performed, the overall body-to-screen mapping at time t can be indicated with

$$\mathbf{y} = f(\mathbf{x}, \theta, T_{t-1}) \tag{7}$$

which constitutes a nonlinear, 3D projective warp.

3.2 Model Initialization

The object model consists of a 3D CAD mesh and one or more reference views. The views are taken from real or rendered images of the object, at known poses; in order to simplify the description, we refer to a single reference view θ_{ref}, which provides the base texture.

In order to obtain a multi-resolution template, the texture image is blurred with R Gaussian filters of increasing size, and the resulting images are used for texture mapping.

Afterwards, in order to sample surface points for tracking, an off-screen rendering window is created, under perspective projection with the same focal length F and resolution of the camera, where the multiresolution model is rendered with standard z-buffering. By knowing the model pose θ_{ref}, the depth of each visible pixel is then back-projected in 3D space, and a large set of N model points $\mathbf{x}_n$ is collected, together with their corresponding intensity values $\mathbf{u}_n \equiv [u_{n,1}, ..., u_{n,R}]^T$ at each resolution r.

This set constitutes the global template, in the following denoted by

$$M \equiv \{(\mathbf{x}_1, \mathbf{u}_1), \dots, (\mathbf{x}_N, \mathbf{u}_N)\}. \tag{8}$$

4 On-Line Mutual Information and Derivatives Computation

As already mentioned in Sect. 2, in order to obtain a completely analytical formulation for the similarity function and its derivatives, we use B-Splines for different purposes:

- at the image level, in order to obtain a smooth interpolation and differentiation of grey values at non-integer coordinates (x, y)
- at the statistical level (the joint intensity histogram), in order to obtain a smooth dependence with respect to the corresponding model-image intensity pairs (u, v).

The latter technique, also termed *fuzzy binning*, amounts to distributing the contribution of a given pair (u, v) between neighboring cells, according to the distance of the sample point from the respective cell center. For both purposes, uniform quadratic splines have been employed, providing sufficient first-order differentiability for the Levenberg–Marquardt optimization (Sect. 5).

The computation of Mutual Information, and its first derivatives with respect to the pose parameters, proceeds as follows.

4.1 Collecting the Intensity Sample

During on-line tracking, from the incoming camera image I a corresponding multiresolution set $I_1, \dots, I_R$ is computed, by using the same Gaussian filters employed for the reference template.

Afterwards, 2D interpolation of each resolution is performed

$$v_r(x, y) = \sum_j \sum_i c_r[i, j] B(x - i) B(y - j) \tag{9}$$

with B the quadratic spline basis function.

The coefficient matrices c_r, with the same size of the image, result from the interpolation constraints

$$v_r(h, k) = I_r[h, k], \qquad h, k = 0, 1, 2, \dots. \tag{10}$$

Efficient computation of the c_r coefficients is obtained here through the B-Spline filtering technique (Unser et al. 1993). By working at a given resolution for both template and image, for sake of clarity in the following we will drop the subscript r.

The representation (9) is given by smooth and differentiable polynomial kernels, so that both sub-pixel intensity values and spatial derivatives can be analytically evaluated with the same computational effort

$$\frac{\partial v}{\partial x} = \sum_j \sum_i c[i, j] \frac{dB}{dx}(x - i) B(y - j), \tag{11}$$

$$\frac{\partial v}{\partial y} = \sum_j \sum_i c[i, j] B(x - i) \frac{dB}{dy}(y - j). \tag{12}$$

At a given pose hypothesis θ, visible model points from the set (8) $(\mathbf{x}_1, \dots, \mathbf{x}_N)$ are then computed by using again the z-buffering visibility test. In our multiresolution approach, the model set M is also subsampled at higher resolutions, by taking a uniformly distributed subset of visible surface.[1]

[1] In the present implementation, instead of computing a sub-sampled image pyramid, the model sample M is subsampled while keeping a

Selected model points are then projected onto the screen through the warp function (7)

$$\mathbf{y}_n = f(\mathbf{x}_n, \theta). \tag{13}$$

The (2 × 6) transformation Jacobian is computed

$$J_f(\mathbf{x}, \theta) \equiv \left.\frac{\partial f}{\partial \theta}\right|_{(\mathbf{x},\theta)} \tag{14}$$

and underlying image values v_n and gradients J_v are evaluated at each projected coordinate $\mathbf{y}_n$

$$\nabla v_n \equiv \left[\frac{\partial v_n}{\partial x}, \frac{\partial v_n}{\partial y}\right] \tag{15}$$

so that the gradient of underlying image intensity w.r.t. pose parameters θ is given by

$$J_{v,n} = \nabla v_n J_f(\mathbf{x}_n, \theta). \tag{16}$$

The obtained set of intensity pairs and gradients $(u_n, v_n, J_{v,n})$ is used in order to evaluate Mutual Information and its derivatives.

4.2 Building the Joint Histogram and its Gradient

In order to obtain the 2D joint histogram of grey levels, a fuzzy binning procedure is performed, by using again the separable quadratic spline kernel. For this purpose, the cell (c_u, c_v) which the point (u, v) belongs to, is computed as

$$c_u = \left\lfloor u\frac{N_c}{256}\right\rfloor; \qquad c_v = \left\lfloor v\frac{N_c}{256}\right\rfloor \tag{17}$$

with N_c the number of cells per dimension. Individual cell and neighborhood contributions from the sample pair are given by

$$P(c_u + i, c_v + j) = P(c_u + i, c_v + j) + b_{u,i} b_{v,j} \tag{18}$$

with $b_{u,i}, b_{v,i}$ the kernel values

$$\begin{aligned} b_{u,i} &= B(w(u) + i), \\ w(u) &= \left| u\frac{N_c}{256} - c_u \right|. \end{aligned} \tag{19}$$

Since the basis function has a compact support of 3 units, each model point contributes to 9 overall neighboring cells $(i, j = -1, 0, 1)$.

The contribution to the joint histogram gradient is then given by

$$\begin{aligned} &\frac{\partial}{\partial\theta} P(c_u + i, c_v + j) \\ &\quad = \frac{\partial}{\partial\theta} P(c_u + i, c_v + j) + b_{u,i} \frac{\partial b_{v,j}}{\partial v} J_{v,n} \end{aligned} \tag{20}$$

and both the joint histogram and its gradient are normalized by the sum $\sum\sum P(c_u, c_v)$, thus obtaining an estimate of the intensity distribution

$$P(c_u, c_v), \frac{\partial}{\partial\theta} P(c_u, c_v) \tag{21}$$

stored into $(N_c \times N_c)$ and $(N_c \times N_c \times 6)$ arrays, respectively.

From this representation, marginal distributions and gradients are simply obtained by summation over rows and columns

$$\begin{aligned} P_u(c_u) &= \sum_{c_v} P(c_u, c_v), \\ P_v(c_v) &= \sum_{c_u} P(c_u, c_v), \\ \frac{\partial}{\partial\theta} P_v(c_v) &= \sum_{c_u} \frac{\partial}{\partial\theta} P(c_u, c_v). \end{aligned} \tag{22}$$

As already mentioned in the Introduction, using uniform B-Splines for fuzzy binning and interpolation ensures to cancel the partial derivative of marginal template distribution

$$\frac{\partial}{\partial\theta} P_u(c_u) = 0 \tag{23}$$

which is the expected requirement for a constant model template.

4.3 Computing MI and Derivatives

Mutual Information is obtained by using (21) and (22) as

$$MI = \sum_{c_u}\sum_{c_v} P(c_u, c_v) \log \frac{P(c_u, c_v)}{P_u(c_u) P_v(c_v)} \tag{24}$$

and its gradient is given by

$$\frac{\partial MI}{\partial\theta} = \sum_{c_u}\sum_{c_v} \left.\frac{\partial P}{\partial\theta}\right|_{(c_u,c_v)} \log \frac{P(c_u, c_v)}{P_v(c_v)}. \tag{25}$$

The appropriate first-order approximation to the Hessian matrix for the LM algorithm (Thevenaz and Unser 2000), is finally given by

$$\begin{aligned} \frac{\partial^2 MI}{\partial\theta^2} &= \sum_{c_u}\sum_{c_v} \left[\frac{1}{P} \frac{\partial P}{\partial\theta}^T \frac{\partial P}{\partial\theta}\right]_{(c_u,c_v)} \\ &\quad - \sum_{c_v} \left[\frac{1}{P_v} \frac{\partial P_v}{\partial\theta}^T \frac{\partial P_v}{\partial\theta}\right]_{c_v}. \end{aligned} \tag{26}$$

constant image size. If spline pyramids were used (Unser et al. 1993), a further speedup could be obtained by computing only the first resolution coefficients, and processing the other resolutions directly in spline-space; however, for a small set R this improvement has shown not to be necessary.

Algorithm 1 Pose estimation through maximization of MI

```
Input: [Geometry+Texture] 3D model (see (8)); Input grey-scale image $I$; initial pose guess $\theta_0$
Initialize: Get $R$ resolution images by Gaussian filtering; Get B-Spline coefficient matrices for each resolution; Compute
    the visible points subset from the set (see (8)), at the initial pose $\theta_0$
Main Loop: Perform a Levenberg–Marquardt optimization for each resolution
for $r = R, \dots, 1$ do
  Compute the initial $MI$ (24), gradient $\mathbf{g}$ (25) and Hessian matrix $H$ (26) at pose $\theta_0$
  while Convergence criteria not met do
    $eval := eval + 1$
    Compute the Newton update $\Delta\theta = (H + \lambda\,\mathrm{diag}(H))^{-1}\mathbf{g}$
    Try out the new parameters $MI(\theta - \Delta\theta)$
    if MI is increasing then
        Accept parameter update $\theta \leftarrow \theta - \Delta\theta$
        Compute new gradient and Hessian matrix
        Decrease $\lambda$: $\lambda = \min(\lambda/10, 10^{-6})$
        Check convergence criteria (see (27))
    else
        Reject new parameters and increase $\lambda$: $\lambda = \max(10\lambda, 10^{6})$
    end if
  end while
end for
Output: Estimated object pose $\theta$
```

5 Estimation of 3D Object Pose

In order to optimize MI starting from an initial parameter guess θ_0, a Levenberg–Marquardt loop (Algorithm 1) is performed for each model and image resolution, starting from the coarsest one R. For this purpose the MI, gradient and Hessian matrix of Sect. 4 are evaluated at different pose hypotheses θ.

The visible subset of model points (8) is evaluated only at the beginning of each optimization loop (at pose θ_0), for a small expected viewpoint variation during local search. The constancy of model sample ensures at the same time a better stability of the LM algorithm, as well as a significant computation speedup.

Each optimization loop exits when one or more of the following conditions are met

$$\begin{aligned}
&\mathrm{abs}(\Delta I) < \tau_f,\\
&\|\Delta\theta\| < \tau_\theta,\\
&\mathrm{eval} > \mathrm{eval}_{\max},\\
&\lambda \geq 10^6
\end{aligned} \tag{27}$$

with ΔI, $\Delta\theta$ the function and parameter increments respectively, $eval$ the cumulative number of function and derivative evaluations, and τ_f, τ_θ two suitable tolerance values.

Optimization parameters are also specified in advance: the initial Levenberg–Marquardt coefficient λ, the subsample rate, and the convergence parameters (27). The output pose θ is then used as start value for the next optimization loop.

By working with different resolutions, we efficiently avoid possible local optima by first optimizing a smooth, wide and mono-modal similarity function in the first loop, while increasing precision for finer resolutions, thus guaranteeing at the same time a better precision, speed and a larger convergence region for θ.

In Fig. 2, an example of single-frame 4-resolution estimation is shown, along with the Gaussian filter width σ for each resolution. The overall number of MI and derivative evaluations needed for this example is around 100 although, as in most of the cases, already after the first two LM loops (roughly half of the total number of function evaluations) the estimated pose converges to the final result.

The full pose estimation algorithm is synthesized in Algorithm 1.

6 Experimental Results

We describe here results obtained by applying our technique to simulated and real object tracking applications, compared to invariant keypoints matching, and standard LSE template tracking.

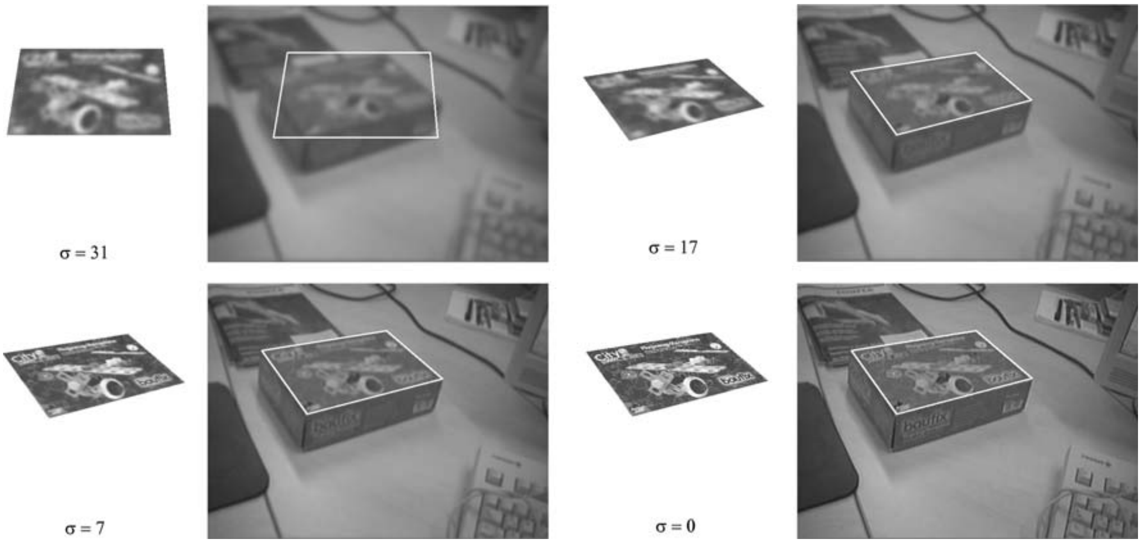


Fig. 2 Single frame multiresolution matching. σ is the standard deviation of the Gaussian filter employed for each resolution

Table 1 RMS errors of successfully tracked frames for the simulated sequence

	X-Y-Z orientation [deg]			X-Y-Z position [mm]		
SIFT	1.4316	1.4118	0.6986	0.9716	0.9715	5.0797
LSE	0.7809	0.9457	0.1898	0.7218	0.3976	5.1820
MI	0.9582	0.6650	0.2251	0.5419	0.4611	2.7205

As keypoint detection technique, we refer to the SIFT implementation from the *ERSP* Vision Library.[2] Implementation of LSE template tracking is instead obtained by replacing the MI function evaluation (Sect. 4) with the SSD gradient and Gauss–Newton matrix (as in the Lucas–Kanade algorithm Baker and Matthews 2004; Matthews and Baker 2003).

6.1 Simulated Sequence with Ground Truth Available

As a first experiment, we run the tracking algorithms on a virtual rendered sequence, with ground truth available. In this experiment, the 3D textured model of a toy box (Fig. 2) has been rendered using OpenGL onto a real background sequence; in order to simulate lighting effects, reflectance properties have been given to the model surface, together with virtual point light sources (Fig. 3).

The model template (8) for this planar object consists of roughly $N = 100{,}000$ collected visible points, each one with $R = 4$ resolution intensity values.

Ground truth motion for this sequence has been generated by using a 2nd order AR model (White Noise Acceleration). Apart from light shading effects, the virtual sequence yet does not contain any external shadow, noise or object occlusions, as well as camera distortion effects, therefore providing relatively good tracking conditions.

In Fig. 4 we can see the output position and orientation errors for each modality; in particular, orientation errors are evaluated by using the equivalent axis-angle representation $(\mathbf{v}, \theta)$, and the three components of the error vector $\mathbf{w} = \theta\mathbf{v}$ are displayed; for this experiment, average *rms* values for orientation and translation errors are given in Table 1. Missing detections have not been considered while computing these values. As it can be seen, template tracking methods achieve a generally higher precision over frame-by-frame keypoint detection, already for a noise and occlusion-free simulated sequence.

An important aspect concerns the number of estimation failures for both methods: over the 1000 frames of the sequence, 13 failure cases have been recorded for SIFT, 16 for SSD and no one for MI tracking. In particular, most SIFT and LSE failures have been observed for views with strong perspective distortion, where local keypoints are hardly detectable, and where at the same time the surface appearance shows a very different shading pattern from the reference template.

Another important point of comparison between LSE and MI optimization concerns the single-frame overall number of function and gradient evaluations. Figure 5 shows the number of evaluations for the two cases, throughout the sequence.

While a single evaluation of Mutual Information is computationally more expensive than SSD, as we can see the average number needed is much higher and less regular for SSD, so that finally the frame rate is slower and less predictable in the second case.

[2] http://www.evolution.com/products/ersp/.

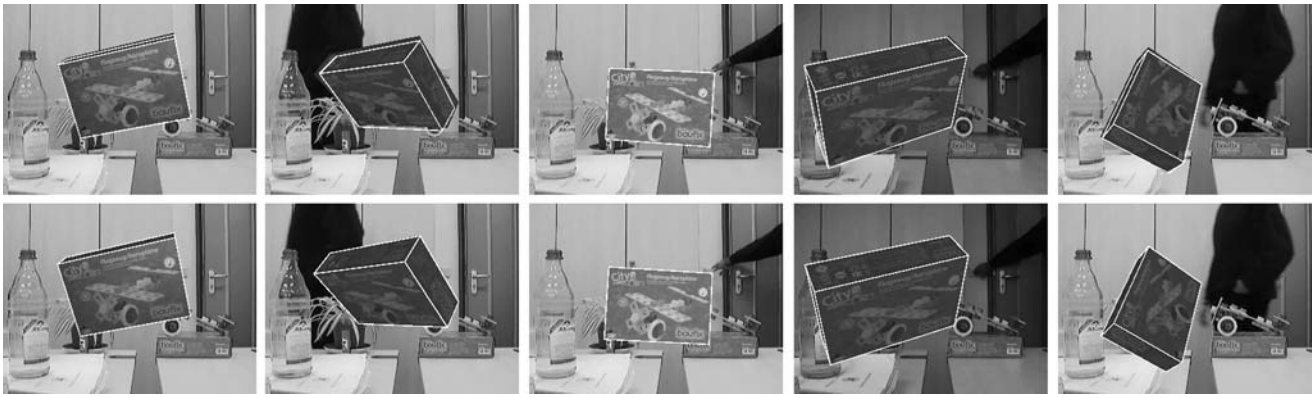


Fig. 3 Simulated object tracking with light and shading changes. *Top row*: result of keypoint-based estimation; *bottom row*: result of MI template tracker

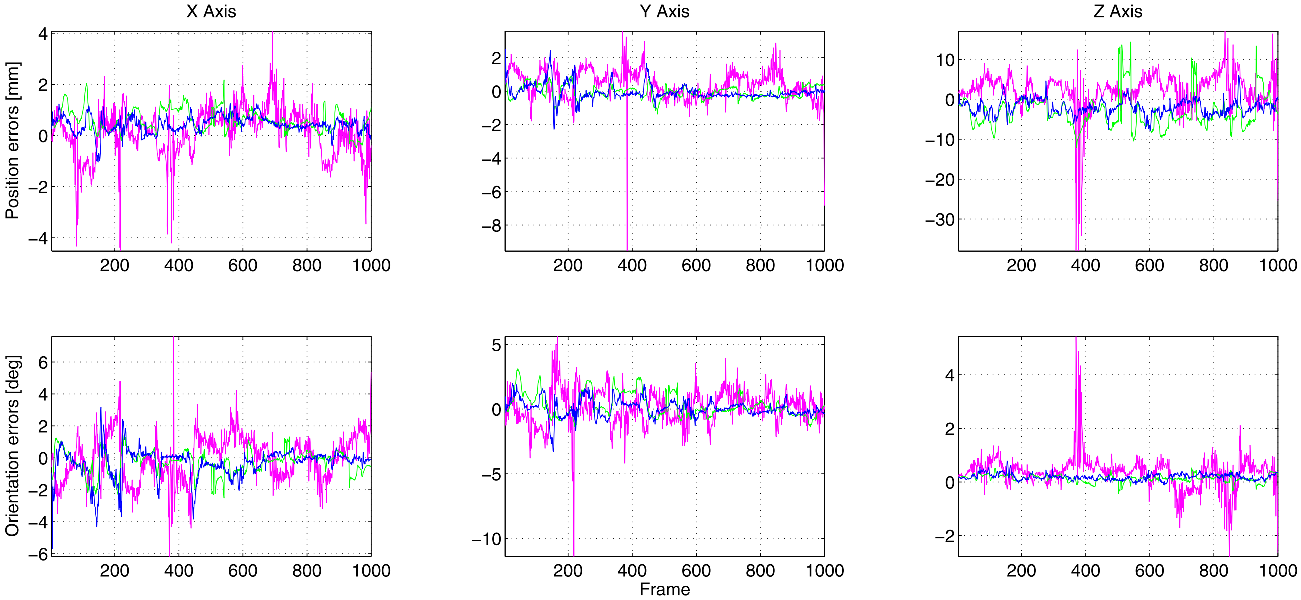


Fig. 4 (Color online) Position and orientation errors for the simulated experiment of Fig. 3, with respect to the available ground truth, over the 6 roto-translational degrees of freedom. Failed pose estimations are not shown. *Red line*: SIFT; *Green*: LSE; *Blue*: MI

The reason for the observed difficulty in the optimization process is twofold. On one side, by working with full non-planar 3D templates, we use the full nonlinear projective warp (7), which generally make the estimation problem more ill-conditioned with respect to other transformations such as a planar homography, or the piece-wise affine warp of (Matthews and Baker 2003). On the other side, concerning the LSE estimator the situation is made worse by the fact that we employ a unique template, without any appearance update, which as we have seen can give an unpredictable and less reliable behavior of the cost function in presence of a variable light shading pattern.

The average frame-rate for MI optimization, on the same Hw/Sw platform (Intel Xeon, 3.15 GHz), is between 1–2 frames/sec, whereas keypoint detection shows a frame rate between 2–3 fps.

6.2 Experiments on Real Sequences

Subsequently, we tested the same algorithms on real sequences involving the same object. The first experiment still involves the toy box, for which the same template model has been used; the sequence has been recorded by using a common webcam, with standard off-line calibration and without any compensation for nonlinear image distortion or motion blurring effects.

For this case ground truth is not available, so that precision of results can only be visually evaluated, by looking at superimposed object model and image edges at estimated

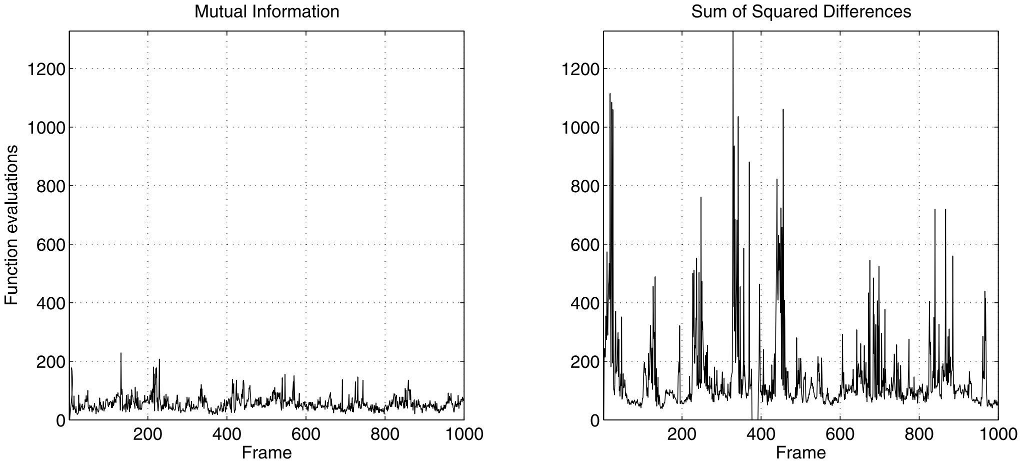


Fig. 5 Computational complexity of our MI algorithm vs. a standard LSE tracker, referred to the simulated toy box sequence. Shown are numbers of function evaluations per frame for the two similarity measures. Between frame 377 and frame 392 the LSE tracker lost the object, therefore the number of function evaluation is not shown

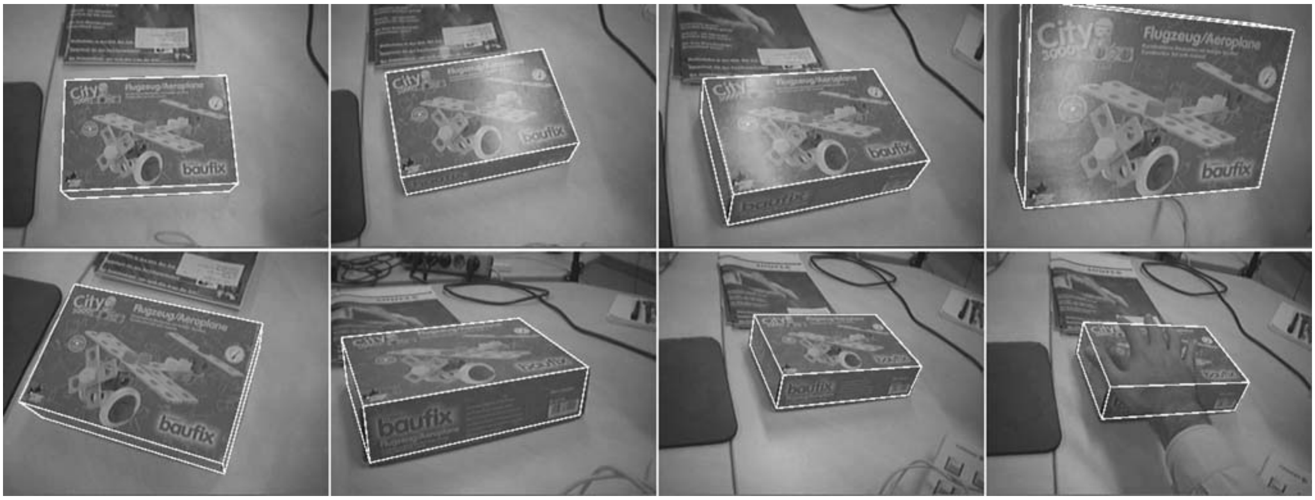


Fig. 6 Robust alignment in presence of partial occlusions, perspective and lighting effects

poses, as well as the motion throughout the sequence; Fig. 6 shows some frames with the results of the MI tracker, that again showed both a higher stability and precision, in particular in presence of partial occlusions, significant light and shading variations, and poses with strong perspective distortion.

Also in this case, the number of estimation failures has been different: 81 for SIFT vs. 17 for MI, over the full 532 frames of the sequence. The LSE tracker definitely failed pose estimation after frame 452, where perspective distortion, a higher distance, and partial occlusion and shading effects posed too difficult conditions using a single appearance model, whereas MI robustly kept the estimation accuracy (bottom right frame).

By considering a more complex model, we subsequently tested the mentioned approach for 3D face tracking applications. For this experiment, a generic 3D head model has been off-line adapted from two photos (front and profile) of the subject by using a modeling procedure similar to (Park et al. 2004); from the same photos the texture has been mapped as well.

The template model (8) in this case has been obtained through the procedure described in Section III-B. A long car-driving sequence has been recorded, and a few frames are shown in Fig. 7.

Since the model surface for this case is non-planar, a z-buffer visibility test is performed at every frame t, in order to select the from the full template the visible points at last estimated pose θ_{t-1}, before the optimization process.

In this case, the SIFT detector has shown to be insufficient for tracking, because of too few detected keypoints from any view, therefore comparisons are not given. Figure 7

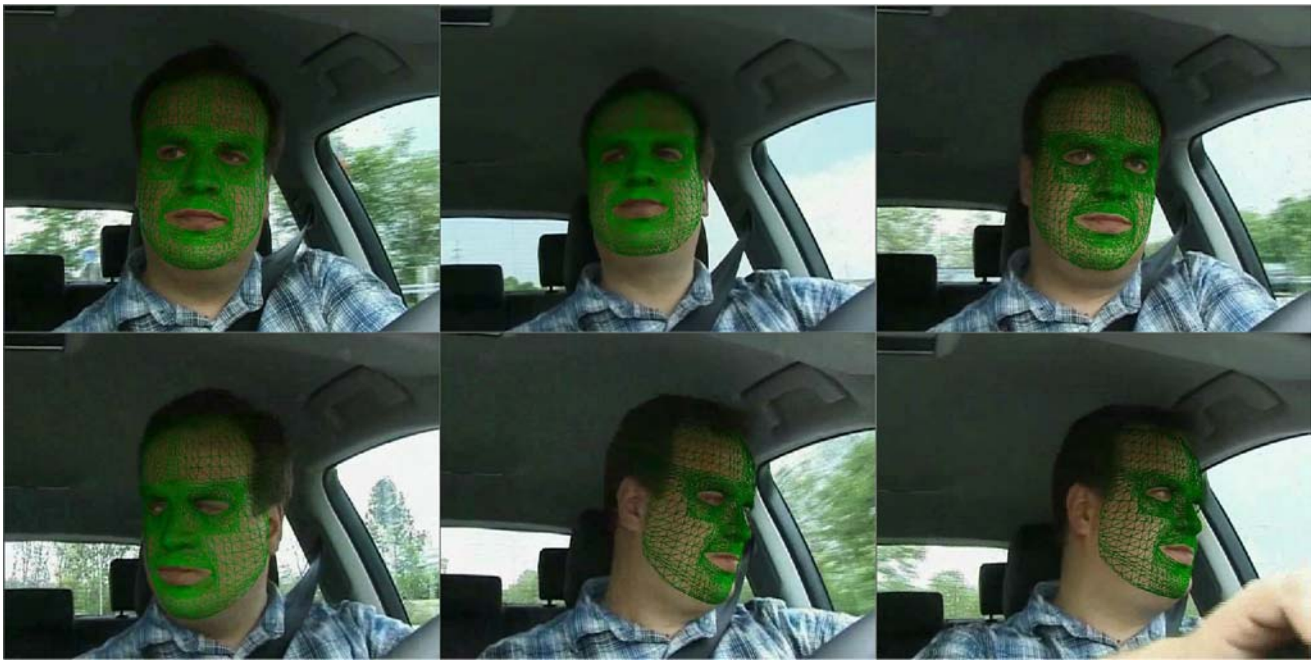


Fig. 7 Stable 3D head tracking in a car-driving environment

shows the result of MI with the superimposed wireframe mesh, exhibiting almost the same tracking performances observed for the toy box case; initialization is provided here by a standard face detection algorithm (Viola and Jones 2001).

7 Conclusions

We presented a robust algorithm for template-based object tracking in image sequences using Mutual Information optimization and a single appearance template. The proposed methodology can be applied to a variety of 3D textured objects, with a generic 3D shape and a distinctive overall appearance, including face models, and a large class of 3D objects as well.

The present implementation can be improved for real-time purposes in several ways. A first idea is using GPU-based computations for rendering template views, computing visible points and on-board optimizing MI as well; this will be possible through the available new generation graphic cards supporting algebra computations, e.g. the *CUDA*[3] architecture. An important improvement concerns the case of planar surfaces under a linearized warp function, for which an inverse-compositional parameter update approach akin to (Matthews and Baker 2003), could be developed by moving most of derivative computations for MI from the image to the template side.

An important note concerns re-initialization of tracking, both at the beginning and in case of loss. Since template tracking is based on a frame-to-frame local search, tracking can be lost for fast object motions, and therefore a global detection method is needed (e.g. the same SIFT methodology).

The choice of a SIFT-based initialization for the above described experiments is of course not meant for providing comparison with local keypoints, since the algorithm is called here only occasionally, in a loss situation which hopefully should happen as seldom as possible. In fact, for initialization purposes any other detection method over the whole image can in principle be used; however, we consider a discussion over initialization issues to be outside the scope of the present work, which focuses on the frame-to-frame tracking methodology, whereas of course it is of a primary importance for a tracking system design.

Planned work in the direction of a complete tracking system, therefore, includes a global, derivative-free MI-based search of the object template in a large pose space, for example using a particle-based method (Kennedy and Eberhart 1995) or Genetic Algorithms (Goldberg 1989).

[3] http://developer.nvidia.com/object/cuda.html

References

Baker, S., & Matthews, I. (2004). Lucas–Kanade 20 years on: a unifying framework. *International Journal of Computer Vision*, *56*(3), 221–255.

Black, M. J., & Jepson, A. D. (1996). Eigentracking: robust matching and tracking of articulated objects using a view-based representation. In *European conference on computer vision* (Vol. 1, pp. 329–342).

Brunelli, R., & Poggio, T. (1993). Face recognition: features versus templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *15*(10), 1042–1052.

Cascia, M., Sclaroff, S., & Athitsos, V. (1999). *Fast, reliable head tracking under varying illumination: an approach based on registration of texture-mapped 3d models*.

Cootes, T. F., Edwards, G. J., & Taylor, C. J. (1998). Active appearance models. *Lecture Notes in Computer Science*, *1407*, 484–498.

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory*. New York: Wiley.

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis*. New York: Wiley.

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, *24*(6), 381–395.

Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Reading: Addison–Wesley.

Gonzalez, R. C., & Woods, R. E. (2006). *Digital image processing* (3rd ed.). Upper Saddle River: Prentice-Hall.

Gorodnichy, D., Malik, S., & Roth, G. (2002). Affordable 3d face tracking using projective vision. In *International conference on vision interfaces* (pp. 383–390).

Hager, G. D., & Belhumeur, P. N. (1998). Efficient region tracking with parametric models of geometry and illumination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(10), 1025–1039.

Huber, P. (1981). *Robust statistics*. New York: Wiley.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks* (Vol. 4, pp. 1942–1948).

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Lu, L., Dai, X.-T., & Hager, G. (2004). A particle filter without dynamics for robust 3d face tracking. In *Proceedings of the 2004 conference on computer vision and pattern recognition workshop (CVPRW'04)* (Vol. 5, p. 70). Washington: IEEE Computer Society.

Maes, F., Collignon, A., Vandermeulen, D., Marchal, G., & Suetens, P. (1997). Multimodality image registration by maximization of mutual information. *IEEE Transactions on Medical Imaging*, *16*(2), 187–198.

Marquardt, D. W. (1963). An algorithm for least-squares estimation of nonlinear parameters. *j-J-SIAM*, *11*(2), 431–441.

Matthews, I., & Baker, S. (2003). *Active appearance models revisited* (Technical Report CMU-RI-TR-03-02). Robotics Institute, Carnegie Mellon University.

Nelder, J., & Mead, R. (1965). A simplex method for function minimization. *Computer Journal*, *7*, 308–313.

Park, I. K., Zhang, H., Vezhnevets, V., & Choh, H.-K. (2004). Image-based photorealistic 3-d face modeling. In *International conference on automatic face and gesture recognition* (pp. 49–56).

Pluim, J. P. W., Maintz, J. B. A., & Viergever, M. A. (2003). Mutual-information-based registration of medical images: a survey. *IEEE Transactions on Medical Imaging*, *22*(8), 986–1004.

Principe, J., Xu, D., & Fisher, J. (1999). Information theoretic learning. In S. Haykin (Ed.), *Unsupervised adaptive filtering*. New York: Wiley.

Shi, J., & Tomasi, C. (1994). Good features to track. In *IEEE conference on computer vision and pattern recognition (CVPR'94)*, Seattle, June 1994.

Skrypnyk, I., & Lowe, D. G. (2004). Scene modelling, recognition and tracking with invariant image features. In *ISMAR '04: proceedings of the third IEEE and ACM international symposium on mixed and augmented reality (ISMAR'04)* (pp. 110–119), Washington, DC, USA. Los Alamitos: IEEE Computer Society.

Thevenaz, P., & Unser, M. (2000). Optimization of mutual information for multiresolution image registration. *IEEE Transactions on Image Processing*, *9*(12), 2083–2099.

Toyama, K. (1998). Look, ma—no hands!' hands-free cursor control with real-time 3d face tracking. In *Proceedings of the workshop on perceptual using interfaces (PUI'98)* (pp. 49–54), San Francisco.

Toyama, K., & Hager, G. (1996). Incremental focus of attention for robust visual tracking. *International Journal on Computer Vision*, *35*(1), 45–63.

Unser, M. (1999). Splines: a perfect fit for signal and image processing. *IEEE Signal Processing Magazine*, *16*(6), 22–38. IEEE Signal Processing Society's 2000 magazine award.

Unser, M., Aldroubi, A., & Eden, M. (1993). B-spline signal processing: part I: theory. *IEEE Transactions on Signal Processing*, *41*(2), 821–833.

Unser, M., Aldroubi, A., & Eden, M. (1993). B-spline signal processing, part II: efficient design and applications. *IEEE Transactions on Signal Processing*, *41*(2), 834–848.

Unser, M., Aldroubi, A., & Eden, M. (1993). The L_2-polynomial spline pyramid. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *15*(4), 364–379.

Vacchetti, L., & Lepetit, V. (2004). Stable real-time 3d tracking using online and offline information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(10), 1385–1391.

Viola, P. A., & Jones, M. J. (2001). Robust real-time face detection. In *International conference on computer vision* (p. 747).

Wells, W., Viola, P., Atsumi, H., Nakajima, S., & Kikinis, R. (1996). *Multi-modal volume registration by maximization of mutual information*.

Xiao, J., Baker, S., Matthews, I., & Kanade, T. (2004). Real-time combined 2d + 3d active appearance models. In *CVPR* (pp. 535–542).