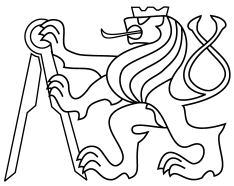
CENTER FOR
MACHINE PERCEPTION



CZECH TECHNICAL
UNIVERSITY

REPRINT

# Multiview 3D Tracking with an Incrementally Constructed 3D Model

Karel Zimmermann, Tomáš Svoboda and Jiří Matas

zimmerk@cmp.felk.cvut.cz

Karel Zimmermann, Tomáš Svoboda and Jiří Matas, *Multiview 3D Tracking with an Incrementally Constructed 3D Model*,3rd International Symposium on 3D Data Processing, Visualization and Transmission, Chapel Hill, USA, 2006

Available at
ftp://cmp.felk.cvut.cz/pub/cmp/articles/zimmerk/zimmerk-3dpvt06.pdf

Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University
Technická 2, 166 27 Prague 6, Czech Republic
fax +420 2 2435 7385, phone +420 2 2435 7637, www: http://cmp.felk.cvut.cz

# Multiview 3D Tracking with an Incrementally Constructed 3D Model

Karel Zimmermann[1], Tomáš Svoboda[1,2] and Jiří Matas[1]

[1]: Center for Machine Perception
Czech Technical University
Prague, Czech Republic

[2]: Center for Applied Cybernetics
Czech Technical University
Prague, Czech Republic

## Abstract

*We propose a multiview tracking method for rigid objects. Assuming that a part of the object is visible in at least two cameras, a partial 3D model is reconstructed in terms of a collection of small 3D planar patches of arbitrary topology. The 3D representation, recovered fully automatically, allows to formulate tracking as gradient minimization in pose (translation, rotation) space. As the object moves, the 3D model is incrementally updated. A virtuous circle emerges: tracking enables composition of the partial 3D model; the 3D model facilitates and robustifies the multiview tracking.*

*We demonstrate experimentally that the interleaved track-and-reconstruct approach successfully tracks a 360 degrees turn-around and a wide range of motions. Monocular tracking is also possible after the model is constructed. Using more cameras, however, significantly increases stability in critical poses and moves. We demonstrate how to exploit the 3D model to increases stability in the presence of uneven and/or changing illumination.*

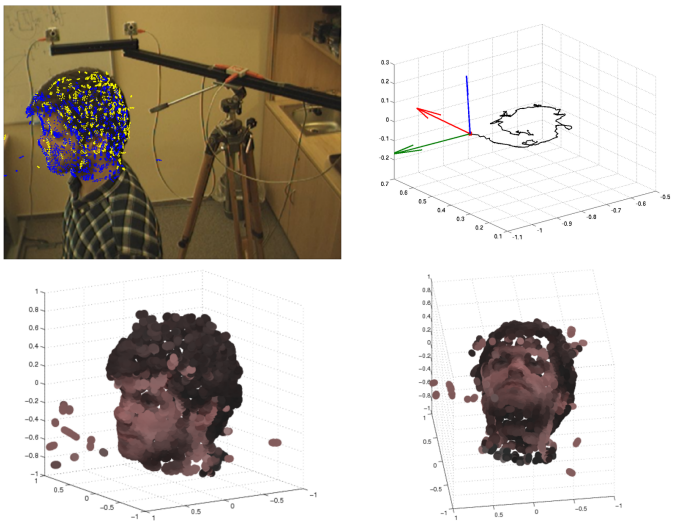


**Figure 1. Interleaved model contruction and tracking: Camera image with reprojected model, trajectory of the head and two different views of the automatically constructed model.**

## 1 Introduction

Existing multiview approaches mostly represent objects as blobs. Blob representation assumes that the appearance of an object does not significantly change when the object rotates. Global object position is sought and the methods do not attempt to recover the *orientation* of the object [3, 9].

Most *model-based tracking* methods use 3D models prepared offline. An overview of such methods was recently published by Lepetit et al. [7]. Vacchetti et al. [16] propose a tracker based on matching with keyframes. The method demonstrates impressive results on out-of-plane rotation data. Still, it cannot track complete turn of the object and needs offline manual selection of keyframes which are essential for its stability. Muñoz et al. [10] suggest a method that track even deformable objects. Their model is composed of small textured planar patches, a set of shape bases, and a set of texture bases. The tracking procedure needs a reference image and optimizes over local shape deformations, colour/texture changes and overall motion. Results on real data show successful tracking only of small variations in object pose and negligible local deformations.

Several approaches build elaborated 3D models from multiple views. The methods rely heavily on carefully constructed and expensive setup and require special scene arrangement since they are based on scene/object segmentation [1, 5, 8, 17]. Würmlin et al. [17] propose dynamic 3D point samples for streaming 3D video. This point based representation somehow resembles our model. However, the method does not track object and needs many cameras and very precise pixel-wise motion segmentation.

We propose a combined method that tracks objects in 3D and constructs a point based appearance model simultaneously. The primary interest is object tracking and detection. The model is rather simple, a set of 3D points associated with 3D orientation and albedo. Despite its simplicity, the model is rich enough for recovering orientation of the object. The tracking can follow a complete 360 degree turn of object. Rothganger et al. [12] also compose a 3D model from small planar patches. The patches are reconstructed from multiview correspondences. Objects are photographed an object from several viewpoints, corresponding image patches are found by affine covariant feature matching. Finally, patches are reconstructed in 3D. In fact, it would be possible to use this model in our tracking. Any complete off-line built model [11] could be used, too.

Cobzas and Jagersand [2] propose a monocular, registration-based, 3D camera tracking of the planar 3D patches. The 3D planar patches are estimated from tracks. Although the formulation of the tracking resembles our method, there are several differences. The patch based model is initialized at the beginning of the sequence (in about 100 frames) by using a standard 2D patch based tracker. Then the algorithm switches to tracking and refine the model using 3D model-based tracking. Cobzas et al. estimate camera pose, assuming a rigid scene. Unlike our method which models illumination changes, Cobzas et al. assume constant illumination and intensity of observed points. Our method builds the model from the very beginning of the sequence. Tracked objects change their position and orientation w.r.t. to light sources. In this case, constant pixel intensities cannot be assumed even for Lambertian surfaces and our method reflects this.

## 2 3D tracking

An object $O$ is modelled as a triplet $(X, \alpha, N)$ where $X$ is a set of 3D points, $\alpha : X \rightarrow \mathcal{R}$ assigns albedo and $N : X \rightarrow \mathcal{S}^2$ a normal to each point $\mathbf{x} \in X$, where $\mathcal{S}^2$ is a sphere. During tracking, intensity $T(\mathbf{x})$ of point $\mathbf{x}$ in a given frame is predicted from its albedo $\alpha(\mathbf{x})$ and an estimated illumination as detailed in section 3.

Assuming rigidity, the motion of points $\mathbf{x} \in X$ between two time instances $t_1$ and $t_2$ is

$$\mathbf{x}^{t_2} = \mathtt{R}\mathbf{x}^{t_1} + \mathbf{d},$$

where $\mathtt{R}$ represents rotation and $\mathbf{d}$ translation. When the rotation is small [4] (e.g. between two consecutive video frames), the motion equation simplifies to

$$\mathbf{x}^{t} = (\mathtt{I} + \mathtt{D})\mathbf{x}^{t-1} + \mathbf{d}, \tag{1}$$

where the rotation matrix $\mathtt{R}$ is replaced by an antisymmetric matrix $\mathtt{D}$ and an identity matrix $\mathtt{I}$. Matrix $\mathtt{D}$ is defined by

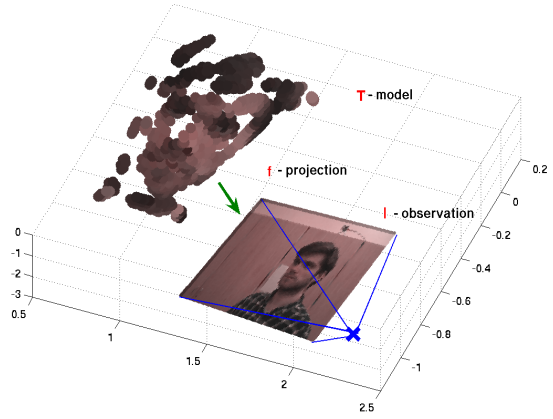



**Figure 2. Model (template) $T$ is projected by projection function $f$ and compared to the current observation $I$.**

three parameters $\mathbf{u} = [D_1, D_2, D_3]^T$;

$$\mathtt{D} = \begin{bmatrix} 0 & D_3 & -D_2 \\ -D_3 & 0 & D_1 \\ D_2 & -D_1 & 0 \end{bmatrix}.$$

Tracking in 3D is defined as the process of finding motion parameters $\mathtt{D}, \mathbf{d}$ minimizing the following image dissimilarity

$$\sum_{\mathbf{x} \in X} \left[ T\left(\mathbf{x}^{t-1}\right) - I\Big(f\left(\mathbf{x}^{t}\right)\Big) \right]^2, \tag{2}$$

where $I : \mathcal{R}^2 \rightarrow \mathcal{R}$ assigns intensity to each pixel, $T : X \rightarrow \mathcal{R}$ assigns intensity to each 3D point. The projection function $f : \mathcal{R}^3 \rightarrow \mathcal{R}^2$ maps 3D points to image coordinates and depends on internal and external parameters of the camera, see Appendix A for details.

Substituting from equation (1) for $\mathbf{x}^t$ in the dissimilarity function (2) and simplifying notation by setting $\mathbf{x}^{t-1} = \mathbf{x}$, a cost function in six unknowns is obtained

$$J(\mathbf{u}, \mathbf{d}) = \sum \left[ T\left(\mathbf{x}\right) - I\Big(f\left(\mathbf{x} + \mathtt{D}\mathbf{x} + \mathbf{d}\right)\Big) \right]^2, \tag{3}$$

where the sum is over all $\mathbf{x} \in X$ as in (2); starting from (3) the summation range is omitted for brevity. We seek motion parameters $\mathbf{u}$ and $\mathbf{d}$ that minimize dissimilarity $J(\mathbf{u}, \mathbf{d})$. At the minimum, the partial derivatives with respect to all variables must be zero:

$$\frac{\partial J(\mathbf{u}, \mathbf{d})}{\partial \mathbf{d}} = \mathbf{0}, \quad \frac{\partial J(\mathbf{u}, \mathbf{d})}{\partial \mathbf{u}} = \mathbf{0},$$

which yields the following two vector equations

$$\sum \Big[ T(\mathbf{x}) - I\big(f(\mathbf{x} + \mathtt{D}\mathbf{x} + \mathbf{d})\big) \Big] \frac{\partial I\big(f(\mathbf{x} + \mathtt{D}\mathbf{x} + \mathbf{d})\big)}{\partial \mathbf{d}} = \mathbf{0}, \tag{4}$$

$$\sum \Big[T(\mathbf{x}) - I\big(f(\mathbf{x}+\mathtt{D}\mathbf{x}+\mathbf{d})\big)\Big] \frac{\partial I\big(f(\mathbf{x}+\mathtt{D}\mathbf{x}+\mathbf{d})\big)}{\partial \mathbf{u}} = \mathbf{0}, \quad (5)$$

There is no closed-form solution for $(\mathbf{u}, \mathbf{d})$. We therefore apply Newton-Raphson minimization, approximating $I\big(f(\mathbf{x}+\mathtt{D}\mathbf{x}+\mathbf{d})\big)$ by its first-order Taylor expansion

$$I\big(f(\mathbf{x}+\mathtt{D}\mathbf{x}+\mathbf{d})\big) \approx I\big(f(\mathbf{x})\big) + \mathbf{g}^T(\mathtt{D}\mathbf{x}+\mathbf{d}), \quad (6)$$

where

$$\mathbf{g}^T = I'^T\big(f(\mathbf{x})\big) f'(\mathbf{x}); \quad (7)$$

$I' : \mathcal{R}^2 \to \mathcal{R}^2$ is the gradient of image $I$ and $f' : \mathcal{R}^3 \to \mathcal{R}^{2\times 3}$ is the Jacobian of the projection function $f$.

Differentiating the linear approximation (6) leads to

$$\frac{\partial I\big(f(\mathbf{x}+\mathtt{D}\mathbf{x}+\mathbf{d})\big)}{\partial \mathbf{d}} \approx \mathbf{g}, \quad (8)$$

$$\frac{\partial I\big(f(\mathbf{x}+\mathtt{D}\mathbf{x}+\mathbf{d})\big)}{\partial \mathbf{u}} \approx \frac{\partial \mathbf{g}^T \mathtt{D}\mathbf{x}}{\partial \mathbf{u}}. \quad (9)$$

Applying the approximations (8), (9), equations (4), (5) are simplified to

$$\sum \big[T(\mathbf{x}) - I\big(f(\mathbf{x})\big) - \mathbf{g}^T\mathtt{D}\mathbf{x} - \mathbf{g}^T\mathbf{d}\big]\mathbf{g} = \mathbf{0}, \quad (10)$$

$$\sum \big[T(\mathbf{x}) - I\big(f(\mathbf{x})\big) - \mathbf{g}^T\mathtt{D}\mathbf{x} - \mathbf{g}^T\mathbf{d}\big]\frac{\partial \mathbf{g}^T\mathtt{D}\mathbf{x}}{\partial \mathbf{u}} = \mathbf{0}, \quad (11)$$

Simple algebraic manipulations confirms that the following two identities hold

$$\mathbf{g}^T\mathtt{D}\mathbf{x} = (\mathbf{g}\times\mathbf{x})^T\mathbf{u},$$

$$\frac{\partial \mathbf{g}^T\mathtt{D}\mathbf{x}}{\partial \mathbf{u}} = (\mathbf{g}\times\mathbf{x}),$$

where $\times$ is the cross product. Equations (11) and (10) can be compactly represented as a system of six linear equations $\mathtt{A}$.

$$\mathtt{A}\begin{bmatrix}\mathbf{u}\\ \mathbf{d}\end{bmatrix} = \mathbf{b}, \quad (12)$$

where

$$\mathtt{A} = \sum \begin{bmatrix} (\mathbf{g}\times\mathbf{x})(\mathbf{g}\times\mathbf{x})^T & (\mathbf{g}\times\mathbf{x})\mathbf{g}^T \\ \mathbf{g}(\mathbf{g}\times\mathbf{x})^T & \mathbf{g}\mathbf{g}^T \end{bmatrix}, \quad (13)$$

$$\mathbf{b} = \sum \big[T(\mathbf{x}) - I\big(f(\mathbf{x})\big)\big] \begin{bmatrix} (\mathbf{g}\times\mathbf{x}) \\ \mathbf{g} \end{bmatrix}. \quad (14)$$

Assuming regular $\mathtt{A}$, the solution approximately minimizing equation $J(\mathbf{u}, \mathbf{d})$ is

$$\begin{bmatrix}\mathbf{u}\\ \mathbf{d}\end{bmatrix} = \mathtt{A}^{-1}\mathbf{b}. \quad (15)$$

The $6 \times 6$ matrix $\mathtt{A}$ consists of four $3 \times 3$ sub-matrices and is block-wise symmetric. Unknown motion parameters $\mathbf{d}$, $\mathbf{u}$ are both $3 \times 1$ column vectors and $\mathbf{b}$ is a $6 \times 1$ column vector.

At least six points are required for $\mathrm{rank}(\mathtt{A}) = 6$. In practice, many more points are visible. If the object is weakly textured back-projected image derivatives $\mathbf{g}$ may get close to zero and matrix $\mathtt{A}$ becomes nearly singular. Texture properties needed for reliable tracking of the object are discussed in [14]. Unlike [14], we optimize over the whole object not just over a small patch.

Newton-Raphson iterations are carried out until convergence or a maximum number of steps $N$. Experiments showed the process converged usually in $8 - 10$ iterations. Convergence may require more iterations when the motion is fast, so $N$ was set to 20.

The tracking method was derived for an intensity image and single camera. Extension to RGB tracking is straightforward. The single sum in solution (13,14) is replaced by summations over all visible points, cameras and all RGB channels.

## 3 Compensation of Illumination

Intensity recorded during model acquisition depends, besides the object shape and reflectance, on light sources. We treat the intensity as albedo. As the object moves, the set of light sources visible from a point and their photometric angles change. When modeling these effects we assume:

- cast shadows can be ignored,
- the light sources are distant,
- no specular reflectance.

Under these assumptions, intensities of all points with identical normals will be scaled by a common matrix (for grayscale images only scalar is considered). We adopted a simple method for estimation of the matrix, which performed well in experiments. The method clusters the points $X$ into $n$ groups $G_1, \ldots, G_n$ according to their normals and compensates the illumination of $i$-th cluster in each optimization step (15) by a color correction matrix

$$\mathtt{E}_i^* = \arg\min_{\mathtt{E}_i} \sum_{\mathbf{x}\in G_i} \|\mathtt{E}_i I(f(\mathbf{x})) - T(\mathbf{x})\|_2^2. \quad (16)$$

Let us denote

$$F(\mathtt{E}_i) = \sum_{\mathbf{x}\in G_i} \|\mathtt{E}_i I(f(\mathbf{x})) - T(\mathbf{x})\|_2^2 =$$

$$\sum_{\mathbf{x}\in G_i} I^T(f(\mathbf{x}))\mathtt{E}_i^T\mathtt{E}_i I(f(\mathbf{x})) - 2T^T(\mathbf{x})\mathtt{E}_i I(f(\mathbf{x})) + T^T(\mathbf{x})T(\mathbf{x}),$$

then minimization yields the following matrix equation

$$\frac{\partial F(\mathtt{E}_i)}{\partial \mathtt{E}_i} = \sum_{\mathbf{x}\in G_i} -2T(\mathbf{x})I^T(f(\mathbf{x}))+2\mathtt{E}_i^* I(f(\mathbf{x}))I^T(f(\mathbf{x})) = \mathtt{0} \tag{17}$$

and the least square solution is

$$\mathtt{E}_i^* = \left[\sum_{\mathbf{x}\in G_i} I(f(\mathbf{x}))I^T(f(\mathbf{x}))\right]^{-1} \sum_{\mathbf{x}\in G_i} T(\mathbf{x})I^T(f(\mathbf{x})). \tag{18}$$

## 4 Tracking-Modeling Algorithm

A minimal configuration able to build the model must include at least one stereo pair. For tracking, a single camera is sufficient.

If no model is available from a previous tracking-modeling session, the processing starts with a stereo-based reconstruction [6] of the visible part of the object. Albedo of each point is determined from the average of intensities at its projections onto images used for 3D reconstruction. The reconstructed points are clustered and replaced by points on fish-scales [13]. Fish-scales are small oriented planar patches obtained by local clustering of the cloud of points. Small clusters of points are replaced by ellipses with half-axes corresponding to the two main eigenvectors of their covariance matrix. The third eigenvector defines the surface normal. Note that, computation of fish-scale representation is much simpler then a complete surface triangulation. Still the fish-scales are experimentally shown to be sufficient representation for 3D tracking. Knowledge of surface orientation at each points allows:

- Efficient visibility calculations for convex objects.
- Compensation of illumination effects.

Once the partial model is known, it can be used for pose estimation. If observed motion in the image indicates that a part of the image moves consistently with points currently in the model, stereo is invoked again and newly reconstructed patches are merged into the model. The complete algorithm is summarized in Figure 3.

Note, that the system never knows when the model is completed, because another consistently moving rigid part of the object can appear later. The system only detects that no reconstruction is currently needed.

1. Capture images
2. If needed, invoke **stereo reconstruction** and merge it to the model.
3. **Estimate the pose** of the object by iterating least square solution (15).
4. **Update matrices** $\mathtt{E}_1, \ldots, \mathtt{E}_n$ and for all $i$ and each $\mathbf{x} \in G_i$ recompute object intensity $T(\mathbf{x}) \leftarrow \mathtt{E}_i T(\mathbf{x})$. **goto 1**.

**Figure 3. Tracking-Modeling Algorithm**

## 5 Experiments

The sequences were captured in an office. We used four firewire cameras with resolution of $640 \times 480$ pixels connected to Linux operated computers. The acquisition was TCP/IP synchronized and the setup was calibrated. The total cost of the setup (without computers) is less than 500 dollars and calibration is easy since a free software for automatic (self)calibration exists [15].

Two different sequences were used. In the *human* sequence, a person makes a variety of motions. The individual walks around, shakes and tilts his head. The camera setup consists of two narrow-baseline cameras for stereo reconstruction and two other cameras spanning approximately a half-circle.

The *book* sequence poses slightly different challenges. The book is a relatively thin object and in some poses the dominant planes (front and back cover) are invisible. The camera setup consists of three cameras located near each other. Two of them are used for stereo, all of them are used for tracking. The model of the book is incrementally constructed from a stereo pair and tracked in all cameras.

Objects are tracked successfully in both sequences and their shapes are correctly reconstructed. We performed experiments to assess the accuracy and robustness of multiview and monocular tracking. Section 5.1 shows that the accuracy of multiview tracking is sufficient for incremental model construction without additional alignment. Section 5.2 compares monocular and polynocular tracking. We show that monocular tracking often estimates poses which are incorrect but look correct in the tracking camera. Robustness is tested in section 5.3 on the book sequence where the tracking survives even in frames where dominant planes are absent. Experiments showing illumination compensation are described in section 5.4. Tracking speed is considered in section 5.5. Experiments in sections 5.4,5.5 are conducted with illumination compensation.

In Figures 3-5, projections of visible points are depicted in blue and invisible in yellow. Readers are encouraged to zoom-in the Figures in the electronic version of the document and watch the accompanying video sequences.

### 5.1 Interleaved Tracking and Model Construction

The first experiment demonstrates the interleaved operation of tracking and model construction. The process starts with a partial reconstruction in the first frame, see the leftmost column of Figure 4. The tracker is initialized using this partial model. As the human is turning Fig.4(b), the model, is augmented by adding further partial reconstructions Figs. 4(c,d). Once the 360 turn is finished, the model is complete and further reconstruction are not required.

The 3D model is only a side product of the tracking. Its visual appearance cannot match models created with specialized stereo algorithms or visual-hull based algorithms.

### 5.2 Monocular Model-Based 3D Tracking

In the case of monocular tracking, a 3D model and its initial position are considered to be known in advance (e.g. we use the model from previous experiment). The head was successfully tracked over 630 frames, despite the fact that both 3D translation and out-of-plane rotation were present in the sequence. Tracking results are shown in Figure 5. In images from the tracking camera, the projected model poses seem correct. However, since only a single camera was used, the recovered 3D position is inaccurate, see row 2 in Figure 5. Naturally, the more cameras are used for the optimization, the more accurate 3D pose becomes. Results from the same sequence with the object tracked by all cameras are depicted in the last row of Figure 5.

### 5.3 Robustness against Critical Poses

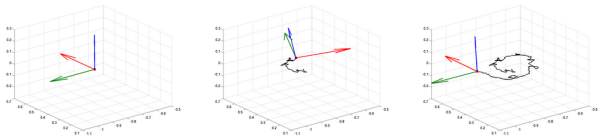
A thin object like the book used in the experiment may easily appear in poses which are inherently challenging for a tracking algorithm. If only the back is visible, the tracking may get unstable. Even during multiview tracking it may happen that most of the object is visible only in a single camera. We call such poses critical.

In a critical pose, the book has to be tracked virtually from the single view. The position of the model does not correspond to the projection in the cameras where only a small fraction of the book is observable. After the object leaves critical pose, the model converges to the true position, see Figure 6.

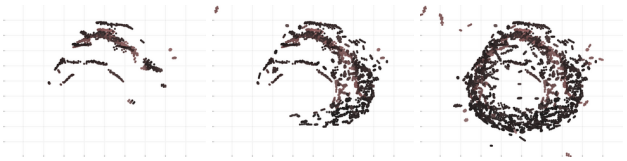
### 5.4 Compensation of Illuminance Effects

The model points are clustered in 14 equally distributed clusters according to their normals. Each cluster is associated with illuminance constant $E_i$ which changes during the tracking to best fit the observed data.

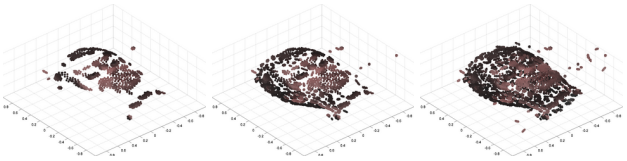
a) Multiview tracking; blue are visible, yellow invisible (occluded) projections



b) Corresponding poses and path recorded



c) Incremental construction of the model, as seen from top



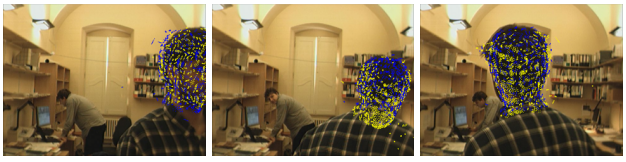
d) Incremental construction of the model, an random view

**Figure 4. Incremental model construction from partial 3D reconstructions and registered by 3D tracking. Rows 1-3: Different views with projected model. Row 4: Position and orientation in 3D space. Rows 5-6: incrementally constructed 3D model. Columns correspond to frames 1, 100 and 310.**

Figure 8-left shows a view with a projected model. Gray levels of particular fish-scales correspond to the values of illuminance constants. Higher values corresponds to the recently illuminated points and vice versa. One can see that in this case light sources were located on the left side of the object which corresponds to the reality.

*Tracking camera*, in monocular tracking, this is the only one used for optimization. Results of monocular tracking projected

*Monocular tracking* results as projected to a camera which approximately orthogonal to the tracking one.



*Polynocular tracking*. The same camera as above. Note the essentially more consistent 3D pose.

**Figure 5. Comparison of monocular (rows 1-2) and polynocular (row 3) tracking. Monocular: Row 1: view from the tracking camera, Row 2: observing camera (shows that, accuracy in orthogonal direction is low). Polynocular: Row 3: The same camera with the projected model from multiview tracking.**

The office has several light sources placed on opposite walls and oriented to the irregularly arched ceiling. Corresponding changes of the illuminance constant $E_6$ during 360 turn are shown at Figure 8-right. Two significant changes during the turn corresponding the light sources are clearly visible. The function of illumination changes is not smooth because during the turn, fish-scales visibility in particular cameras changes and in different times different sets of fish-scales are used for the compensation of illumination effects. Another reason is local inaccuracy of tracking caused by image discretization. Tracking trajectories as well as illumination changes could be smoothed using a motion model, but in our experiments only the output of the optimization is used.

### 5.5 Speed Evaluation

The speed of the algorithm shown in Figure 3 was tested on the sequence introduced in the first two experiments (i.e. 4 cameras, RGB images). Slightly-optimized implementation in Matlab runs cca 1.8 s/frame on an AMD-64b linux running machine. We experimentally show that the tracking of the same sequences in graysacle is successful as well as in RGB. Since one of the most important property of the tracking is the framerate, we increase it 3 many times by considering only grayscale model/sequence.

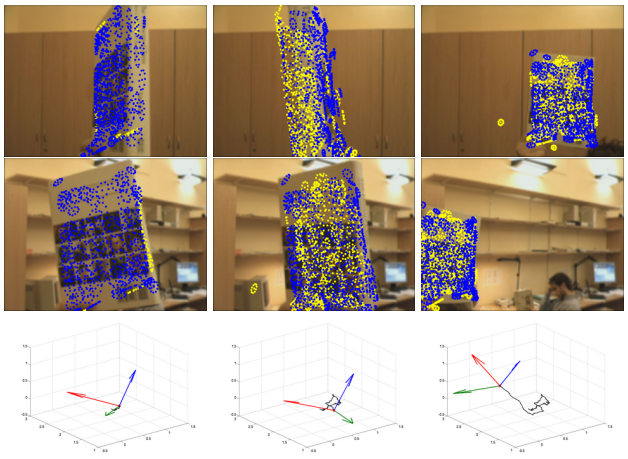


**Figure 6. Book tracking: Rows 1-2: different cameras with projected model, row 3: shows position and orientation in 3D space, columns correspond to frames 55, 205 and 265. The second column shows the book in a critical position where dominant plane is visible only in one camera.**

Tracking of grayscale sequence takes approximately 800 ms/frame. Typically, multiple cameras are connected to different computers. Hence, all the contributions to the $\mathtt{A}, \mathbf{b}$ from equation (13,14) can be computed independently on the particular computers. Using such a system, a frame rate of 5 frames per second can be achieved with the current Matlab implementation.

## 6 Conclusions

We proposed a fully automatic approach of multiview/monocular 3D object tracking interleaved with incremental model construction. Neither model nor initialization are needed to be known in advance. We formulated tracking as a gradient based method minimizing dissimilarity of the observe image and projected 3D point intensities. We showed that the fish-scale 3D model [13] is accurate enough to support stable 3D tracking.

We experimentally demonstrated that the proposed interleaved approach, successfully tracks a complete 360 turn and a wide range of motion without a need for pre-prepared

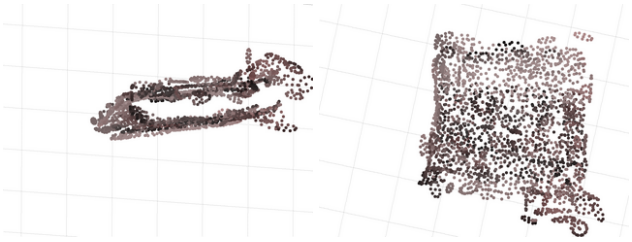


**Figure 7. Book Model: Different views of the book model. Small non-planarity in one corner is the reconstructed hand.**

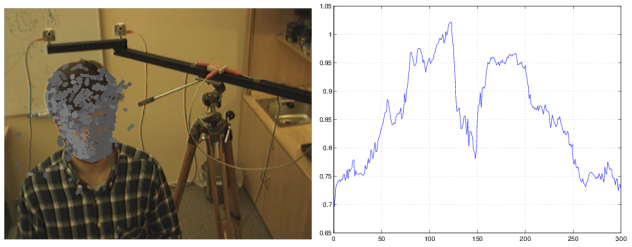


**Figure 8. Left: The image with projected model. Colors correspond to the computed illuminance $\mathtt{E}_i$ of each particular cluster. Right: Values of $\mathtt{E}_6$ during the the 360 turn.**

3D model. A 3D model is delivered as a side product. We demonstrated the robustness of our method on a sequence with a thin object where the dominant plane was often tracked only from one view.

We showed that monocular tracking is possible if the model is available. The model projection to the tracking camera often looks correct, projections to other cameras reveals 3D inaccuracies. Still, monocular tracking can provide results acceptable for some applications. Using more cameras significantly increases stability and accuracy in critical poses and moves. Exact 3D pose may be necessary in many application ranging from virtual reality, human computer interfaces to visual surveillance.

## Acknowledgement

Karel Zimmermann was supported by The Czech Academy of Sciences under project 1ET101210407. Tomáš Svoboda was supported by The Czech Ministry of Education under project 1M0567. Jiří Matas was supported by The European Commission under project IST-004176. Partial support of EU project Dirac FP6-IST-027787 and The STINT under project Dur IG2003-2 062 is also acknowledged.

## Appendix A

A 3D point $\mathbf{x}$ is projected to 2D image (pixel) coordinates $\mathbf{p}$ as

$$\begin{bmatrix} \lambda\mathbf{p} \\ \lambda \end{bmatrix} = \mathtt{P} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix},$$

where $\mathtt{P}$ is $3 \times 4$ camera matrix [4] and $\lambda \in \mathcal{R}$. Let the camera matrix be parameterized as

$$\mathtt{P} = \begin{bmatrix} \mathbf{m}_1^T & t_1 \\ \mathbf{m}_2^T & t_2 \\ \mathbf{m}_3^T & t_3 \end{bmatrix} \tag{19}$$

the function $f : \mathcal{R}^3 \rightarrow \mathcal{R}^2$ projecting 3D point to the camera coordinates is

$$f(\mathbf{x}) = \begin{bmatrix} \frac{\mathbf{m}_1^T\mathbf{x}+t_1}{\mathbf{m}_3^T\mathbf{x}+t_3} \\ \frac{\mathbf{m}_2^T\mathbf{x}+t_2}{\mathbf{m}_3^T\mathbf{x}+t_3} \end{bmatrix}. \tag{20}$$

Differentiating $f$ with respect to $\mathbf{x}$ we obtain $f' : \mathcal{R}^3 \rightarrow \mathcal{R}^{2\times 3}$ Jacobian matrix function, which consists of elements

$$f'_{pq} = \frac{m_{pq}(\mathbf{m}_3^T\mathbf{x} + t_3) - m_{3q}(\mathbf{m}_1^T\mathbf{x} + t_p)}{(\mathbf{m}_3^T\mathbf{x} + t_3)^2} \tag{21}$$

where $m_{pq}, p = 1 \ldots 2, q = 1 \ldots 3$ is $q$-th elements of $\mathbf{m}_p^T$.

## References

[1] J. Carranza, C. Theobalt, M. Magnor, and H.-P. Seidel. Free-viewpoint video of human actors. *ACM Transaction on Computer Graphics*, 22(3), July 2003.

[2] D. Cobzas and M. Jagersand. 3D SSD tracking from uncalibrated video. In *Workshop on Spatial Coherence for Visual Motion Analysis (SCVMA), in conjunction with ECCV 2004*, 2004.

[3] F. Fleuret, R. Lengagne, and P. Fua. Fixed point probability field for complex occlusion handling. In *IEEE International Conference on Computer Vision*, 2005.

[4] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, Cambridge, UK, 2000.

[5] T. Kanade, P. Narayanan, and P. W. Rander. Virtualized reality: Concepts and early results. In *IEEE Workshop on the Representation of Visual Scenes*, pages 69–76, June 1995.

[6] J. Kostková and R. Šára. Stratified dense matching for stereopsis in complex scenes. In R. Harvey and J. A. Bangham, editors, *BMVC 2003: Proceedings of the 14th British Machine Vision Conference*, volume 1, pages 339–348, London, UK, September 2003. British Machine Vision Association.

[7] V. Lepetit and P. Fua. Monocular model-based 3D tracking of rigid objects. *Foundations and Trends in Computer Graphics and Vision*, 1(1):1–89, 2005.

[8] B. C. R. R. M. L. Matusik, W. and S. Gortler. Image–based visual hulls. In *Proceedings of ACM SIGGRAPH*, 2000.

[9] A. Mittal and L. S. Davis. M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene using region-based stereo. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *The seventh European Conference on Computer Vision, ECCV2002*, number 2350 in LNCS, pages 18–36. Springer, May 2002.

[10] E. Muñoz, J. Buenaposada, and L. Baumela. Efficient model-based 3D tracking of deformable objects. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 877–882, China, October 2005.

[11] D. Nistér. Automatic passive recovery of 3d from images and video. In *Second International Symposium on 3D Data Processing, Visualization and TRansmission (3DPVT04)*, 2004. Invited paper.

[12] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3D object modeling and recognition using affine-invariant patches and multi-view spatial constraints. In *International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 272–277, 2003.

[13] R. Šára and R. Bajcsy. Fish-scales: Representing fuzzy manifolds. In S. Chandran and U. Desai, editors, *Proc. 6th International Conference on Computer Vision*, pages 811–817, New Delhi, India, January 1998. IEEE Computer Society, Narosa Publishing House.

[14] J. Shi and C. Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593 – 600, 1994.

[15] T. Svoboda, D. Martinec, and T. Pajdla. A convenient multicamera self-calibration for virtual environments. *PRESENCE: Teleoperators and Virtual Environments*, 14(4):407–422, August 2005.

[16] L. Vacchetti, V. Lepetit, and P. Fua. Stable real-time 3D tracking using online and offline information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1385–1391, 2004.

[17] S. Würmlin, E. Lamboray, and M. Gross. 3D video fragments: Dynamic point samples for real-time free-viewpoint video. *Computers and Graphics*, 28(1):3–14, 2004.