A robust algorithm for 3D–motion estimation of a small surface patch of a 3D–object

Geovanni Martínez
Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung
Universität Hannover, Appelstraße 9A, D–30167 Hannover , F.R. Germany
Phone: ++49–511–762–5311, email: martinez@tnt.uni–hannover.de

Abstract

A robust algorithm for estimating 3D–motion of small surface patches of a 3D–object is described. The robustness is measured by the probability of convergency to correct parameters. The 3D motion is estimated by a gradient method, which uses *a local luminance error model, a motion compensated Kalman Filter* and *a selected neighboring surface around the patch*. This technique can be used for automatic modelling of flexibly connected rigid 3D object components considering mutual occlusion.

1. Introduction

To model 3D scenes from 2D image sequences, a model–based image analysis describes each moving object m by a model object with 3 sets of parameters defining its motion $\mathbf{A}^{(m)}$, shape $\mathbf{M}^{(m)}$, and color $\mathbf{S}^{(m)}$. The parameter sets depend on the type of motion model being applied. In this contribution, the motion model of rigid 3D objects is used [3]. According to this motion model, the motion parameters $\mathbf{A}^{(m)} = (T_x^{(m)}, T_y^{(m)}, T_z^{(m)}, R_x^{(m)}, R_y^{(m)}, R_z^{(m)})$ describe the motion of a rigid object in 3D space. $T_x^{(m)}$, $T_y^{(m)}$, and $T_z^{(m)}$ represent the translation and $R_x^{(m)}$, $R_y^{(m)}$, and $R_z^{(m)}$ the rotation. The shape of the 3D–model object is described by a mesh of triangles which is put up by a set of vertices referred as control points. The color parameters define the luminance and chrominance reflectance of the object surface and are taken from a real image by projection of the corresponding part of the real image onto each visible triangle of the mesh. Each image of a sequence can be reconstructed from the parameters of the model object by projecting the color parameters onto the image plane of the model camera.

In order to estimate the 3D motion parameters $\mathbf{A}^{(m)}$ of an object a basic algorithm was developed in [4]. However, due to the small surface of a surface patch, the probability of convergency to correct parameters will be low. For increasing the probability of convergency the number of the motion parameters to be estimated was reduced in [1]. However, this has the disadvantage that only three parameters $(T_x^{(m)}, T_y^{(m)}, R_z^{(m)})$ can be calculated.

This contribution presents an algorithm for estimating the 6 motion parameters of a small surface patch of a 3D–object based on a gradient method which tries to achieve a higher probability of convergency and accuracy than the algorithm described in [4]. In order to improve the probability of convergency the proposed technique uses a selected neighborhood around the surface patch and for improving the accuracy, it applies a local luminance error model and a motion compensated Kalman filter.

In section 2, the proposed 3D motion estimation algorithm is described. In section 3 and 4, first results on a synthetically generated image sequence and a real image sequence respectively are discussed.

2. The 3D motion estimation algorithm

Fig. 1 shows a rigid model object with a surface patch s of which the 3D–motion shall be estimated. The 3D motion is described by the parameters $\mathbf{A}^{(s)} = (T_x^{(s)}, T_y^{(s)}, T_z^{(s)}, R_x^{(s)}, R_y^{(s)}, R_z^{(s)})$ defining translation and rotation. The surface patch s represents a surface of a 3D model object and consists of N control points $\mathbf{P}_C^{(1)}, \mathbf{P}_C^{(2)}, \; \dots \; , \mathbf{P}_C^{(N)}$ and q triangles. An arbitrary point $\mathbf{P}^{(i)}$ on the surface of s is moved to its new position $\mathbf{P}'^{(i)}$ according to:

$$\mathbf{P}'^{(i)} = [\mathbf{R}_C^{(s)}] \cdot (\mathbf{P}^{(i)} - \mathbf{C}^{(s)}) + \mathbf{C}^{(s)} + \mathbf{T}^{(s)} \tag{2.1}$$

with the translation vector $\mathbf{T}^{(s)} = (T_x^{(s)}, T_y^{(s)}, T_z^{(s)})^T$, the surface patch center $\mathbf{C}^{(s)} = (C_x^{(s)}, C_y^{(s)}, C_z^{(s)}) = \frac{1}{N}\sum_{j=1}^{N} \mathbf{P}_C^{(j)}$ and the rotation matrix$[\mathbf{R}_C^{(s)}]$ which is defined by the rotation angles $R_x^{(s)}, R_y^{(s)}, R_z^{(s)}$, around x–, y– and z–axes with the rotation center $\mathbf{C}^{(s)}$.

To estimate the 3D motion of a surface patch it is assumed that differences between two successive images s_k and s_{k+1} are due to object motion only and that the shape of the object is known. For simplification, it is assumed that only one triangle builds a surface patch s. In order to estimate 3D–motion the proposed algorithm minimizes the mean square luminance difference between a projection of the triangle's luminance component onto the image plane of a model camera and the corresponding luminance of the current image $s_{k+1.}$ For that purpose a gradient method is applied. It uses one set of observation points from each triangle. Each observation point $\mathbf{O}_k^{(j)} = (\mathbf{P}_k^{(j)}, \mathbf{g}^{(j)}, I^{(j)})$ is located on the triangle surface and is described by its position

$\mathbf{P}_k^{(j)} = (P_x^{(j)}, P_y^{(j)}, P_z^{(j)})^T$, its luminance value $I^{(j)}$ and its spatial linear gradient $\mathbf{g}^{(j)} = (g_x^{(j)}, g_y^{(j)})^T$. The luminance and gradient are taken from the same image of which the color parameters of the 3D model object were derived. The measure for selecting observation points is a high spatial gradient. For each observation point, the luminance difference ΔI between image s_k and s_{k+1} is related to motion by the following linearized equation:

$$\begin{aligned}\Delta I = \; & F \cdot g_x/P_z \cdot T_x^{(s)} \\ & + F \cdot g_y/P_z \cdot T_y^{(s)} \\ & - [(P_x g_x + P_y g_y)F/P_z^2 + \Delta I/P_z] \cdot T_z^{(s)} \\ & - [[P_x g_x(P_y - C_y^{(s)}) + P_y g_y(P_y - C_y^{(s)}) + P_z g_y(P_z - C_z^{(s)})]F/P_z^2 + \Delta I/P_z(P_y - C_y^{(s)})] \cdot R_x^{(s)} \\ & + [[P_y g_y(P_x - C_x^{(s)}) + P_x g_x(P_x - C_x^{(s)}) + P_z g_x(P_z - C_z^{(s)})]F/P_z^2 + \Delta I/P_z(P_x - C_x^{(s)})] \cdot R_y^{(s)} \\ & - [g_x(P_y - C_y^{(s)}) - g_y(P_x - C_x^{(s)})]F/P_z \cdot R_z^{(s)}\end{aligned}$$

(2.2)

with the focal length of the camera F, the unknown motion parameters $\mathbf{A}^{(s)} = (T_x^{(s)}, T_y^{(s)}, T_z^{(s)}, R_x^{(s)}, R_y^{(s)}, R_z^{(s)})$ and the observation point $\mathbf{O}_k = (\mathbf{P}_k, \mathbf{g}, I)$ at position $\mathbf{P}_k = (P_x, P_y, P_z)^T$.

The expression (2.2) can be written as:

$$\Delta I = [\mathbf{H}] \cdot \mathbf{A}^{(s)T} \tag{2.3}$$

Using at least 6 observation points, we have a linear system of equations that can be solved by an iterative Newton method for least square error[3][4]. Here, in order to improve the accuracy of the estimates a motion compensated Kalman filter[2] using a local luminance error model is applied. In a motion compensated Kalman filter, a 3D motion estimation of the triangle is carried out before the Kalman filter is applied. This is achieved by solving the set of equations by an iterative Newton method for least square error. Then, to improve the accuracy of those estimates, a Kalman filter is applied using a local luminance error model I_{em} for each observation point $\mathbf{O} = (\mathbf{P}, \mathbf{g}, I)$. Equation (2.3) is extended for the Kalman Filter to the following equation:

$$\Delta I = [\mathbf{H}] \cdot \mathbf{A}^{(s)T} + I_e \tag{2.4}$$

I_e is the luminance error which is modelled by the luminance error model I_{em}. Only those observation points are used by the Kalman Filter where the local luminance error model was found to be valid. The luminance error model considers for each observation point both, the luminance error due to the shape error of the 3D–model object $\mathbf{\Delta P}$ and the luminance error due to camera noise. The shape error and the camera noise are statistically independent. As a first approach, the shape error $\mathbf{\Delta P}$ is modelled by a Gaussian stationary random process describing the shape error of each observation point in the x, y and z directions. These errors are assumed to be uncorrelated, with mean 0 and the same variance $\sigma_M{}^2$. In order to compute the luminance error due to the shape error $\mathbf{\Delta P}$, only its projection $\mathbf{\Delta p_g}$ in the direction of the luminance gradient on the image plane is considered[7]. Therefore, the shape error $\mathbf{\Delta P}$ is mapped to a vector $\mathbf{\Delta p}$ onto the image plane by a linear transformation of the model camera. Then, the vector $\mathbf{\Delta p}$ is projected onto the unit luminance gradient vector $\mathbf{g}^T/|\mathbf{g}|$ for getting $\mathbf{\Delta p_g}$. The resulting luminance error variance $\sigma_{Mg}{}^2$ can be written as:

$$\sigma_{Mg}{}^2 = f^2 \frac{\sigma_M{}^2}{P_z{}^4} \cdot [\; g_x{}^2(P_z{}^2 + P_x{}^2) + 2 g_x g_y P_x P_y + g_y{}^2(P_z{}^2 + P_y{}^2) \;] \tag{2.5}$$

The camera noise is supposed to be a Gaussian uncorrelated zero–mean noise with variance $\sigma_n{}^2$. Finally, the luminance error I_e is modelled by a Gaussian stationary random process with mean 0 and variance $\sigma_{I_{em}}{}^2 = \sigma_{Mg}{}^2 + \sigma_n{}^2$.

In order to improve the probability of convergency and consequently the robustness of the algorithm, the number of observation points to be used for the estimation is increased. For that purpose, both the triangle and its neighboring triangles are taken into account for 3D–motion estimation. However, increasing the number of considered observation points improves the robustness of this algorithm only if the observation points of the neighboring triangles and the observation points of the triangle itself follow the same motion, i.e. belong to the same object. In order to minimize the probability that neighboring triangles from different objects are chosen, for each triangle the neighboring triangles are selected by a 2D segmentation. The regions found by the 2D segmentation represent approximately the silhouettes of the objects in the image. Then, in order to estimate the 3D motion parameters of a triangle, a neighboring triangle will be used only if both triangles belong to the same region found by the 2D segmentation. In this contribution, the 2D segmentation is based on a segmentation of the displacement vector field inside of the silhouette of the 3D model object. The displacement vector field is calculated by hierarchical block matching[6] using the current image s_{k+1} and the previous image s_k. To obtain the segmentation a multithresholding technique[5] is applied. Finally, the obtained segmentation is improved by local analysis of the contours near the boundary of the different regions which were found.

3. Experimental results using a synthetically generated image sequence

To examine the probability of convergency and the accuracy, a synthetically generated image sequence is used. Each frame of this sequence was generated moving a 3D–model object one pixel in both x and y directions and projecting its color parameters onto the image plane of a model camera.

To evaluate the probability of convergency, the *maximal norm* of the error of the estimated translation parameters $\| m_T \|$ and the maximal norm of the error of the estimated rotation parameters $\| m_R \|$ are used:

$$\| m_T \| = \max(\ |T_x^{(s)} - \hat{T}_x|, \ |T_y^{(s)} - \hat{T}_y|, \ |T_z^{(s)} - \hat{T}_z| \ ) \tag{3.1}$$

$$\| m_R \| = \max(\ |R_x^{(s)} - \hat{R}_x|, \ |R_y^{(s)} - \hat{R}_y|, \ |R_z^{(s)} - \hat{R}_z| \ ) \tag{3.2}$$

Here $\mathbf{A} = (T_x^{(s)}, T_y^{(s)}, T_z^{(s)}, R_x^{(s)}, R_y^{(s)}, R_z^{(s)})$ are the true motion parameters which were used for sequence generation and $\hat{\mathbf{A}} = (\hat{T}_x, \hat{T}_y, \hat{T}_z, \hat{R}_x, \hat{R}_y, \hat{R}_z)$ are the estimated motion parameters. In order to examine the probability of convergency, it is assumed, that the range of convergency is reached if the values of $\| m_T \|$ and $\| m_R \|$ are smaller than the thresholds $th_T = 0.5$pel and $th_R = 0.5$grad respectively. Experimental results show that the probability of convergency is 0.8076 by the proposed algorithm *(algorithm 1)* and 0.1057 by the algorithm described in [4] *(algorithm 2)*.

To evaluate the accuracy the estimation error variance σ_{pe}^2 for each estimate is used. Experimental results for each parameter are shown in the table 3.1. The average of the estimation error variance of the translation parameters $\overline{\sigma}_{peT}^2$ was found to be 2.7694 pel^2 by the algorithm 1 and 25.8495 pel^2 by the algorithm 2. The average of the estimation error variance of the rotation parameters $\overline{\sigma}_{peR}^2$ was found to be 0.8416 grad^2 by the algorithm 1 and 3.7033 grad^2 by the algorithm 2. To evaluate the improvement of the accuracy by the motion compensated Kalman filter only, the filter can be switched off. Then, $\overline{\sigma}_{peT}^2$ and $\overline{\sigma}_{peR}^2$ are 3.2471 pel^2 and 1.045 grad^2 without Kalman filter respectively.

Since the criterion for estimating 3D–motion is based on the minimization of the mean square luminance difference (MSE), an improvement of the probability of convergency and of the accuracy can also be evaluated comparing the MSE after motion compensation for each surface patch of the 3D–model object:

$$\mathbf{G} = -10 \log \frac{MSE1}{MSE2} \tag{3.3}$$

MSE1 applies to the algorithm 1 and MSE2 to the algorithm 2. **G** represents the gain on MSE achieved by increasing the probability of convergency and the accuracy by the algorithm 1. Experiments using the first two images of the synthetic sequence show an average gain **G** of 10.2710 dB. Fig. 2 shows the MSE1 and MSE2 for each triangle of the 208 of the 3D–model object.

4. Experimental results using a real image sequence

Results using the 1st and the 2nd images of the real test sequence ”Claire” (CIF, 10Hz) show an average gain **G** of 4.4734 dB on MSE as a result of the improvement of the probability of convergency and the accuracy by the algorithm 1. Here the 3D model object was automatically generated[3]. Fig 3 shows the MSE1 and MSE2 for each of the 149 triangle of the 3D–model object.

5. Conclusions

In this contribution an algorithm for estimating 3D–motion of small surface patches of a 3D–object is proposed. First experiments using a synthetically generated image sequence show that the probability of convergency is greater than 0.8076, an average of the estimation error variance for the translation parameters $\overline{\sigma}_{peT}^2$ of 2.7694 pel^2 and an average of the estimation error variance for the rotation parameters $\overline{\sigma}_{peR}^2$ of 0.8416 grad^2. Comparing this algorithm with the algorithm described in [4], using the selected neighborhood around the surface patch increases the probability of convergency from 0.1057 to 0.8076, additionally improves $\overline{\sigma}_{peT}^2$ from 25.8495 pel^2 to 3.2471 pel^2 and $\overline{\sigma}_{peR}^2$ from 3.7033 grad^2 to 1.0451 grad^2. Using the local luminance error model together with the motion compensated Kalman filter improves $\overline{\sigma}_{peT}^2$ from 3.2471 pel^2 to 2.7694 pel^2 and $\overline{\sigma}_{peR}^2$ from 1.0451 grad^2 to 0.8416 grad^2. The average gain **G** on MSE after motion compensation by increasing the probability of convergency and the accuracy is found to be 10.2710 dB using the synthetically generated image sequence and 4.4734 dB using the real image sequence ”Claire” (CIF, 10Hz). In the future work the proposed algorithm shall be investigated if it can be used for automatic modelling of flexibly connected rigid 3D object components considering mutual occlusion for an object–based analysis–synthesis coder (OBASC)[3].

6. References

[1] H. Busch, ”Subdividing non rigid 3D objects into quasi rigid parts”, Proceedings of the 3rd IEE International Conference on Image Processing and its Application, University of Warwick, 1989.

[2] I. Sezan, R. Lagendijk, ”Motion analysis and image sequence processing”, Kluwer Academic Publishers, Massachusetts, USA, 1993.

[3] J. Ostermann, ”An analysis–synthesis coder based on moving rigid 3D–objects”, Signal Processing, Image Communication, Vol. 6, No 2, May. 1994.

[4] F. Kappei, C.–E. Liedtke, ” Modelling of a natural 3–D scene consisting of moving objects from a sequence of monocular TV images”, SPIE, vol.860, Cannes 1987.

[5] J.N. Kapur, P.K. Sahoo, ”A New Method for Grey–Level Picture Thresholding Using the Entropy of the Histogram”, Computer Vision, Graphics, and Image Processing 29, 273–285, 1985.

[6] M. Bierling, ”Displacement estimation by hierarchical blockmatching”, 3rd SPIE Symposium on Visual Communications and Image Processing, Cambridge, USA, pp. 942–951, November 1988.

[7] Ralf Berger, ”Stereoskopische Bewegungsschätzung unter Berücksichtigung einer fehlerbehafteten Objektgeometrie”, diploma thesis, University of Hannover, Germany, 1993.

True Values (grad, pel)	mean of the estimate algorithm 1	estimate error variance (grad^2, pel^2) algorithm 1	mean of the estimate algorithm 2	estimate error variance (grad^2, pel^2) algorithm 2
$R_x^{(s)} = 0.01$	0.0123	0.8745	0.4163	5.4063
$R_y^{(s)} = 0.01$	0.0056	1.5620	0.4227	4.1201
$R_z^{(s)} = 0.01$	0.0123	0.0884	0.0465	1.5836
$T_x^{(s)} = 1.0$	0.9539	1.8972	0.9099	5.1500
$T_y^{(s)} = 1.0$	1.0485	1.3036	2.2862	6.4887
$T_z^{(s)} = 0.01$	0.4171	5.1075	16.2321	65.9098

Table 3.1 Mean of the estimates and the estimate error variances by the proposed algorithm (algorithm 1) and the algorithm described in [4] (algorithm 2) using the first two images of the synthetically generated sequence.

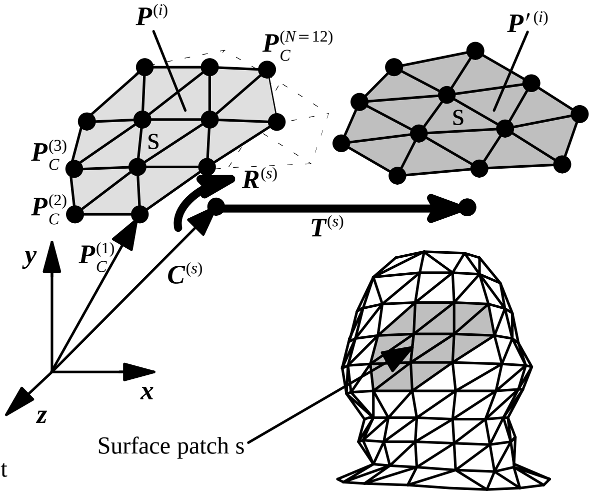

Fig.1 Small surface patch s of a rigid 3D–model object and ist 3D–motion

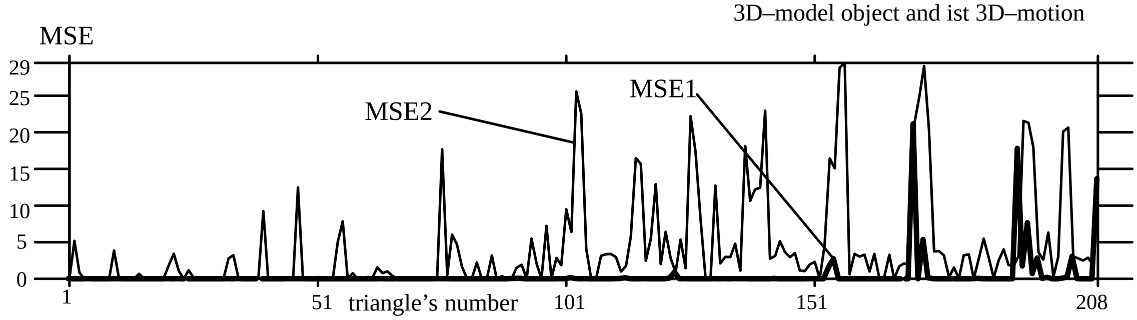

Fig. 2 Mean square luminance error (MSE) after motion compensation for each surface patch (triangle) of the 3D–model object using the first two image of the synthetic image sequence, applying for estimation of 3D–motion the proposed algorithm (MSE1) and the algorithm described in [4] (MSE2).

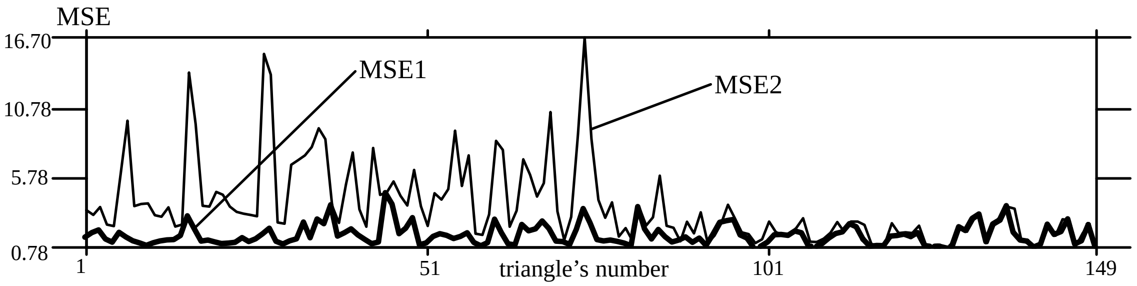

Fig. 3 Mean square luminace error (MSE) after motion compensation for each surface patch (triangle) of the 3D–model object using the first two images of the real image sequence ”Claire” (CIF, 10Hz), applying for estimation of 3D–motion the proposed algorithm (MSE1) and the algorithm described in [4] (MSE2).