MVA2000 IAPR Workshop on Machine Vision Applications, Nov. 28–30, 2000, The University of Tokyo, Japan

13—7

Fitting 3D Models To 2D Imagery: A Physics Based Approach

Martin Berger, Stefan Scherer*
Institute of Computer Graphics and Vision
Graz University of Technology

Vassili Kravtchenko-Berejnoi†
AutomationX
Software for Industrial Automation GmbH

Abstract

This paper presents a physics based approach to fitting object models to images. The process is modelled as a movement of a rigid body in a viscous environment on a flat surface. The driving forces and moments are obtained by analyzing a single image. Gradients and their distances to projected model features support the development of acceleration and velocity of the model until fitting the image. Accurate pose estimates are achieved in real-time, since the computational cost is lower than in Gauss-Newton based approaches. Moreover, the algorithm is less sensitive to illumination changes and shadows. Parameters of the dynamical system can be interpreted in physical terms.

1 Introduction

In many industrial and robotics applications such as quality control, object manipulation or automatic mounting it is essential to determine the exact pose of a single object whereas the object class itself is known. Various computer vision approaches deliver both, the object's class and its pose (cf. [9][4]). This somehow combined approches come with the drawback of mostly inaccurate pose parameters. Recent methods of pose parameter optimization based on the Gauss-Newton approach [7][6][10] are suitable for objects of low complexity and simplest topology, since the computational cost of setting up the Jacobian matrices is very high for complex polyhedral models. Moreover, these methods often demonstrate poor convergence. In this contribution we present a pose refinement algorithm based on the modelling of the convergence process as a physical system. This allows clearer interpretation, and hence control, of the dynamics of the fitting process. In order to ensure proper convergence, the required initial coarse pose estimate is obtained from an Eigenspace approach originally applied in the already mentioned object recognition context. Our working environment is a flat table including a calibrated setup. We are therefore restricted to three degrees of freedom: a translation (x, y) and a rotation φ about the vertical z-axis. An industrial application in which our algorithm is integrated, requires real-time behaviour. In addition to a general efficiency of the procedure, a compromise between accuracy and speed ought to be well controllable. Robustness and capability to handle complex objects stand for reason.

Some related work based on dynamic systems together with exact pose determination was carried out mainly in the fields of vision based tracking (cf. [8]) and augmented reality applications [3][12], where velocity was used to predict model positions. This paper is organized as follows. Sect. 2 will give a short review of Gauss-Newton based techniques, Sect. 3 introduces the new concept of dynamic model fitting based on the development of acceleration and velocity. Results are presented in Sect. 4, Sect. 5 concludes the paper.

2 Previous Work

The fitting of parametric 3D models to images was first addressed by Lowe [7] and extended by Araújo et al. [1]. These methods are based on the Levenberg-Marquardt technique [7], which is able to adapt between Gauss-Newton and pure gradient descent behaviour. Both, gradient weighting [10] and adaptive gradient search [2] help to avoid matching errors and to stabilize the solution. Inherent to all these methods is the fact that a suitable starting parameter vector must be known, in order to ensure convergence towards a global minimum. The solution is calculated from

$$\mathbf{p}^{(i+1)} = \mathbf{p}^{(i)} - \Delta\mathbf{p} \qquad (1)$$

*Address: Inffeldgasse 16/2, 8010 Graz, Austria. E-mail: {berger, scherer}@icg.tu-graz.ac.at

†Address: Teslastr. 8, 8074 Grambach, Austria. E-mail: vassili.kravtchenko@automationx.com

where $\mathbf{p}^{(i+1)}$ and $\mathbf{p}^{(i)}$ denote the pose parameter vectors in iteration $i+1$ and i, respectively. The parameter updates $\Delta\mathbf{p}$ are given by

$$\mathbf{J}\Delta\mathbf{p} = \mathbf{e} \tag{2}$$

where $\mathbf{J}$ is the local Jacobian of the 3D-2D mapping function at $p^{(i)}$ and $\mathbf{e}$ are the observed errors in 2D. Each correspondence between a 3D model line, its projection in the image and the respective perpendicular fitting error results in two rows of the matrix $\mathbf{J}$ and the vector $\mathbf{e}$ (x- and y-coordinate, respectively). The number of equations to solve for an optimal update is $2nk$, where n is the number of visible model lines and k denotes the number of fitting basepoints per line. For complex models, as used in our case, $\mathbf{J}$ becomes very large. The simplest case for computing a solution is solving for $\mathbf{\Delta p}$ in a least squares sense

$$\Delta\mathbf{p} = \left(\mathbf{J}^T\mathbf{J}\right)^{-1}\mathbf{J}^T\mathbf{e} \tag{3}$$

This requires $O(dnk)$ error calculations perpendicular to projected model lines, $O(dnk)$ evaluations of partial derivatives and $O(dnk)$ operations for the least squares parameter estimation. In the case of a Levenberg-Marquardt optimization or gradient weighting, even more operations are needed.

3 A New Physics Based Approach

The proposed method is based on a representation of an iterative fitting process as a development of a dynamical system.

Basic Equations

We describe such a system by means of dynamical equations modelling movements of a rigid body in a viscous environment.

$$\begin{array}{rcl} m\ddot{x} & = & F_x(x,y,\varphi) \quad - \quad \sigma_T\dot{x} \\ m\ddot{y} & = & F_y(x,y,\varphi) \quad - \quad \sigma_T\dot{y} \\ \Theta\ddot{\varphi} & = & M(x,y,\varphi) \quad - \quad \sigma_R\dot{\varphi} \end{array} \tag{4}$$

m represents the mass of the object, Θ is the moment of inertia, σ_T and σ_R denote viscosities. The latter are attenuation constants and are used to delimit acceleration while approaching the solution. $F_x\left(x,y,\varphi\right)$, $F_y\left(x,y,\varphi\right)$ are the x- and y-components of the resulting force calculated from the image, $M\left(x,y,\varphi\right)$ is the resulting moment. Θ can be expressed as

$$\Theta = mr_G^2 \tag{5}$$

where r_G is the radius of gyration of the object. After inserting Eq. 5 in Eq. 4, m can be set to unity mass, without loss of generality.

The proposed method calculates a set of forces (see lines in Fig. 5 and Fig. 7) for each visible model line. Forces are established between segments of projected model lines and edges detected in the image. Depending on the physical model, the magnitude of forces is weighted according to the distance to the gradient and its magnitude.

Modelling Force Magnitudes

A straight-forward implementation for modelling the force magnitudes is the application of the well-known Hook model

$$\mathbf{F} = D\Delta\mathbf{x} \tag{6}$$

where a force $\mathbf{F}$ is proportional (factor D) to elongation $\Delta\mathbf{x}$. Problems arise if $\Delta\mathbf{x}$ becomes too large or if some line segments are matched with wrong image features. This may destabilize the solution like in the Gauss-Newton approach.

Forces can also be interpreted as results of local field gradients ΔI

$$\mathbf{F} = \begin{cases} \frac{\Delta I}{\|\mathbf{x}\|^3}\mathbf{x} & \text{if } \mathbf{x} \neq \mathbf{0} \\ \mathbf{0} & \text{elsewhere} \end{cases} \tag{7}$$

Note the singularity at $\mathbf{x} = \vec{\mathbf{0}}$. It can be easily overcome by inserting a linear slope near zero in Eq. 7.

Finally, a resulting force and a resulting moment in 2D are calculated (see arrows in Fig. 5 and 7). Both are projected onto the workplace and form the input to a differential equation system which models the movement on the workplace. The computational cost of each fitting step is $O(dnk)$ for force calculation and $O(1)$ for resulting force and moment calculation.

Numerical integration of the differential equation is done by a 4th order Runge-Kutta method. It is well known from literture ([5][11]) that the accuracy of the solution at a certain point depends on the choice of the stepsize during integration.

Reducing Oscillations: Adaptive Stepsize

It can be seen that the stability of the solution strongly depends on the progress over time of $\mathbf{F}$ and M and their smoothness. Non-smoothness arises from false edge matching. False edges are likely far from their corresponding model edges, which results in higher force contribution when using the Hook model (Eq. 6). For the field model in Eq. 7, the contribution of false edges in force and torque fields is negligible with a high probability. Thus, avoiding false matches (especially with thin model structures) means avoid oscillations while approaching the solution (see [2]). The solution is computed with a lower number of iterations and lower computational cost.

Since the solution should be computed with a low number of iterations, without loosing accuracy, an adaptive stepsize scheme is applied. The stepsize of the numerical integrator is varied proportionally to the progress of resulting forces and moments. In most cases the adaptive scheme reduces oscillations around the optimal solution.

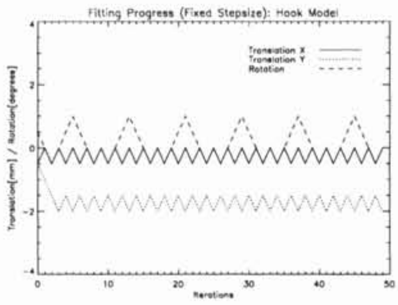

Figure 1: The progress of the computed translation (x, y) and the rotation φ over the number of iterations. Hook's model was applied in this case.

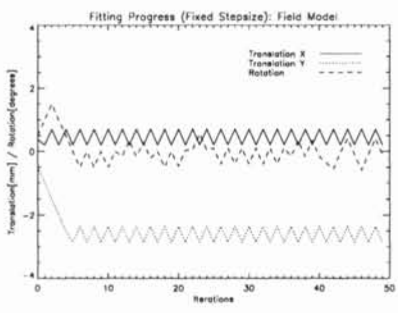

Figure 2: The progress of the translation (x, y) and the rotation φ for the field model.

4 Experimental Results

We tested our method on several objects of different toplogical type and complexity. Their sizes ranged from 30 to 80 mm. The distance to the camera was approximately 700 mm. Two of these objects are shown in Fig. 5 through Fig. 8. The camera (748×576 pixels, 11 mm lens) was calibrated using the Tsai procedure. The starting positions (i.e. the coarse pose parameters obtained from an Eigenspace method), differed from the true solution usually less than 5 mm in translation and less than 5 degrees in rotation. For larger initial displacements, Hook's model (see Eq. 6) turned out to be more efficient, since it is able to accelerate the model very quickly. Switching to the local field gradient model (Eq. 7) when approaching the solution, showed the most efficient way to obtain highly accurate pose parameters. The mean residual displacements from the true

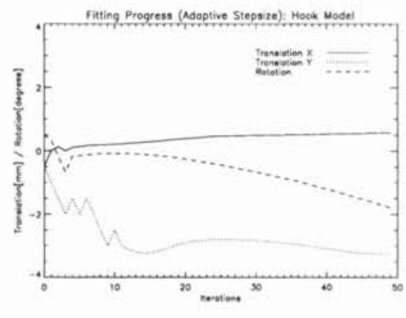

Figure 3: Hook's model together with an adaptive stepsize: Note the progress of the rotation angle, which converges only after approx. 100 iterations (not shown here).

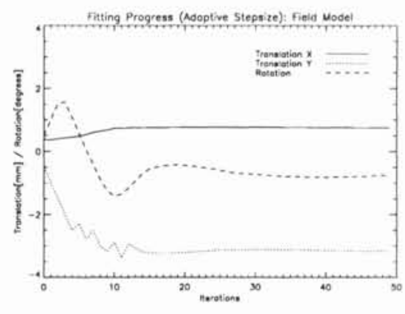

Figure 4: The best case: field model with adaptive stepsize. The convergence is reached after a few iterations.

pose (measured by placing a part with a robot) were 0.3 mm for translation and 0.2 degrees for rotation. Rerunning the procedure from different starting positions ($\pm$5mm and $\pm$5 degrees) resulted in less than 0.05 mm and 0.05 degrees of discrepancy. The described approach turned out to be flexible, fast and able to cope with varying illumination conditions.

5 Conclusion

A physics based approach to fitting 3D models to images was presented. It avoids the calculation of the Jacobian of the mapping function during parameter optimization. Thus, pose can be calculated with less computational effort, even for complex objects. The use of a physical model allows a better control of the fitting process. The main benefits are speed, high accuracy and fast convergence. Further work will focus on the determination of forces based on logical and physical constraints, in order to improve the model-image matching procedure.

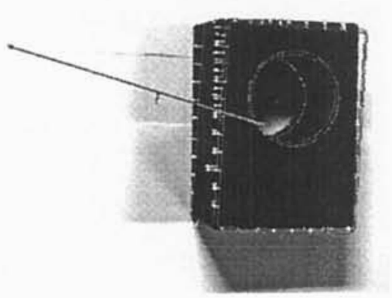

Figure 5: Our method applied to a simple object. Note the small arrows which represent local forces, while the large arrow represents the resulting force, the small arc depicts the resulting moment.

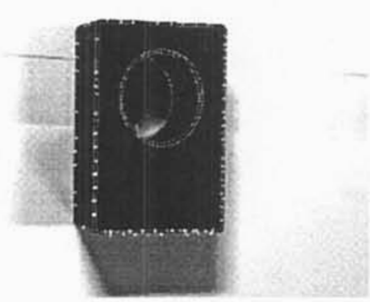

Figure 6: Final position reached after 20 iterations

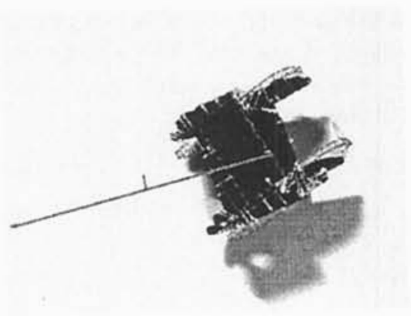

Figure 7: Our method applied to a more complex object.

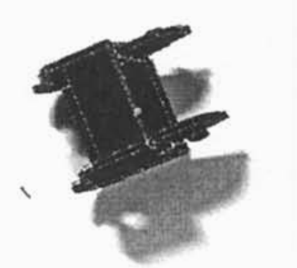

Figure 8: Final position reached after 25 iterations.

References

[1] Helder Araújo, Rodrigo L. Carceroni, and Christopher M. Brown. A Fully Perspective Formulation to Improve the Accuracy of Lowe's Pose-Estimation Algorithm. *CVIU*, 70(2):227–238, 1998.

[2] Martin Berger Thomas Auer, Gernot Bachler, Stefan Scherer, and Axel Pinz. 3D Model Based Pose Determination in Real-Time: Strategies, Convergence, Accuracy. In *Proc. of the 15th ICPR 2000*, Barcelona, Spain, September 2000.

[3] Ronald T. Azuma and Gary Bishop. Improving Static and Dynamic Registration in an Optical See-Through HMD. In *Proc. of SIGGRAPH94*, pages 197–204, Orlando, FL, USA, 1994.

[4] Jeffrey S. Beis and David G. Lowe. Indexing without Invariants in 3D Object Recognition. *IEEE Transactions on PAMI*, 21(10):1000–1015, October 1999.

[5] Lothar Collatz. *Differential Equations: an introduction with applications*. Wiley, 1986.

[6] Dieter Koller, Kostas Daniilidis, and Hans-Hellmut Nagel. Model-Based Object Tracking in Monocular Image Sequences of Road Traffic Scenes. *IJCV*, 10(3):257–281, 1993.

[7] David G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on PAMI*, 13(5):441–450, May 1991.

[8] Éric Marchand, François Chaumette, Patrick Bouthemy, and Valérie Moreau. Robust real-time visual tracking using a 2D-3D model-based approach. In *Proceedings of the ICCV 99*, pages 262–268, Corfu, Greece, September 1999.

[9] Hiroshi Murase and Shree K. Nayar. Visual Learning and Recognition of 3-D Objects from Appearance. *IJCV*, 14:5–24, 1995.

[10] Arthur Pece and Anthony Worrall. A statistically-based Newton method for pose refinement. *Image Vision and Computing*, 16:541–544, 1998.

[11] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge Academic Press, 1993.

[12] Jiann-Rong Wu and Ming Ouhyoung. A 3D tracking experiment on latency and its compensation methods in virtual environments. In *Proceedings of the 8th ACM Symposium on User Interface and Software Technology*, pages 41–49, 1995.