Lens Model Selection for a Markerless AR Tracking System

B. Streckel, J.-F. Evers-Senne, R. Koch
Institute of Computer Science and Applied Mathematics
Christian-Albrechts-University of Kiel, 24098 Kiel, Germany
streckel@mip.informatik.uni-kiel.de

Abstract

This paper describes a visual markerless real-time tracking system for Augmented Reality applications. The system uses a firewire camera with a fisheye lens mounted at 10 fps. Visual tracking of 3D scene points is performed simultaneously with 3D camera pose estimation without any prior scene knowledge. All visual-geometric data is acquired using a structure-from-motion approach. The lens selection was driven by research results that show the superiority of a fisheye lens to a standard perspective lens for this approach. 2D features in the hemispherical image are tracked using a 2D point tracker. Based on the feature tracks, 3D camera ego-motion and 3D features are estimated.

1. Introduction

Augmented Reality (AR) systems aim at the superposition of additional scene data into the video stream of a real camera. In this contribution we consider online augmentation, where a user typically carries a head mounted display (HMD) and camera, and a wearable computer. Additional information is either superimposed onto the video stream or it is projected into the visual path of the users gaze direction [1]. Usually the AR-gear must be carried by the user for a long time, hence it should be lightweight and ergonomic. Despite these restrictions camera pose computation must be fast and reliable, even in uncooperative environments. This requires high computational power of the system and a good quality camera.

There are some recent research activities on online AR, inspired by online tracking algorithms from robotics and computer vision. In robotics the real-time SLAM approach (Simultaneous Localization And Mapping) was recently extended to visual tracking [2]. In computer vision Structure from Motion (SfM) has been in the focus for some years, where simultaneous camera pose estimation and 3D structure reconstruction is possible [8]. Both approaches have much in common and can be merged towards a real-time AR system [4].

2. Online AR System Design

In the following we will describe the components of our online AR system that is based on the SfM approach. It allows robust 3D camera tracking in complex and uncooperative scenes where parts of the scene may move independently. The robustness is achieved in two ways:

1. A 160° hemispherical fisheye lens is used that captures a very large field of view (FoV) of the scene, the camera is oriented in the users viewing direction.
2. The 3D tracking is based on robust camera pose estimation using SfM algorithms that are optimized for real-time performance [8]. These algorithms can handle measurement outliers from the 2D tracking using robust statistics.

The AR system has to be a lightweight wearable solution that allows real-time augmentation via a HMD. The computational load of such a system is too high for current wearable computers, so we use the wearable unit for the HMD and the image acquisition, it is connected to a backend PC via WLAN.

The backend system is currently able to process 10 fps, thus a raw data rate of 1.6 MB/s is transferred through the WLAN channel. In addition the camera rotation is measured by a 3 DoF inertial sensor, this data is used to compensate fast head rotations and to predict image feature positions. The backend system estimates the 3D pose and hands it back to the wearable unit where visual augmentation is superimposed onto the users view. In this paper we only discuss the tracking unit and do not handle augmentation.

Figure 1 gives an overview on the system components. The backend system runs two separate threads (possibly on a 2-processor unit) that separate initialization and 2D feature tracking from the 3D pose estimation.

Initialization and 2D feature tracking In an initial step a set of salient 2D intensity corners is detected in the first image of the sequence. These 2D features are then tracked throughout the image sequence by local feature matching with the KLT operator [9]. If feature tracks are lost, new tracks are constantly reinitialized. The new tracks are merged with previous tracks in the 3D part to avoid drift.

To aid 2D tracking, the 3D camera rotation is measured and compensated by the inertial sensor.

3D feature tracking and pose estimation From the given 2D feature tracks, a SfM approach [6] is applied to estimate the camera pose and 3D feature positions. Given tracks of reliable 2D features, the pose of the cameras can be computed. Simultaneously, 3D feature points can be triangulated from the 2D correspondences and the pose. The camera pose and the 3D feature positions are determined relative to an initial camera position and up to an unknown overall scale, which must be inserted into the system.

SfM assumes a scene with static 3D features between views, therefore moving objects and measurement outliers must be handled robustly. Robustness is introduced by robust statistical methods, like i.e. RANSAC [6], moving objects are treated as measurement outliers that are discarded.

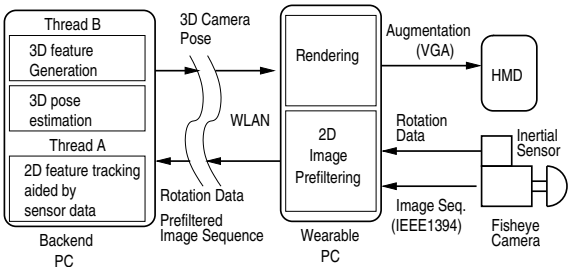

Figure 1. Overview of the AR-System.

3. Selection of the proper lens

There was already some research done on which camera is best suited for the SfM approach [3][7]. The most extensive theoretical approach is made in [7]. It is stated, that a fisheye camera is more appropriate for SfM than a perspective camera, with a drawback because of the low resolution. Therefore a multi-camera or polydioptric camera is proposed. This is not practical for an AR-system, since the camera selection is limited strongly by size and weight.

There are different approaches showing that a wide FoV stabilizes the pose estimation [3]. For perspective cameras with small FoV, the motion towards the optical axis is always ill defined because the camera moves towards the focus of expansion (FOE). Only the motion perpendicular to the FOE can be estimated reliably. In a spherical image with a wide FoV, there will always be an image position perpendicular to the FOE, hence the estimation of the camera motion is more reliable.

Simulation of Structure from Motion In the following we will show that the optimum camera for AR-tracking is indeed a fisheye camera. As a test scenario for the camera comparison, SfM was modeled up to the 3rd image. We computed the effects of noise in the 2D feature tracking on the 3D pose estimation by simulating SfM for perspective and fisheye lenses of varying FoV. All cameras are modeled without distortion, though nearly distortion-free perspective cameras can be built only upto a FoV $< 100°$. Since only the projection model is simulated, the results would also hold for catadioptric cameras that are designed for a constant angular resolution. These cameras are less well suited for augmented reality tasks, since the image center, commonly the users viewing direction, is blocked by the camera itself.

For simulation first a 3D point cloud of 2000 points was generated. This cloud is centered around the camera starting position and used for all experiments throughout all the paper. All the time part of this cloud is visible and projected into the cameras at different positions. The only error source in SfM is the feature position measurement of the 2D point tracker. The tracker is able to generate feature points with a standard deviation of $\sigma = 0.25$ pixel [9]. Mismatches or moving points are neglected for the model. To model the tracking error Gaussian noise with a standard deviation of $\sigma = 0.25$ pixel is added when the 3D points are projected into a camera. With noise proportional to the pixel size the tracking accuracy depends mainly on the camera resolution which was set to $N = 1024$ in x- and y-direction.

With point correspondences for the first two cameras, the pose of the second camera relative to the first can be estimated and 3D points can be triangulated up to scale. Knowing estimated 3D points and the corresponding 2D projection, the pose of the third camera relative to the first can also be evaluated. The camera poses are calculated by a simple least squares approach. This is possible, because there are no real outliers or wrong matches in the model.

Error models for perspective and equidistant projection The difference between a perspective and a fisheye camera is the projection defining how a 3D point is mapped on the camera CCD. The perspective camera forms the image by a perspective projection while the ideal fisheye performs an equidistant projection [5]. These projections define the transformation of the modeled tracker noise into angular noise, which affects the triangulation quality.

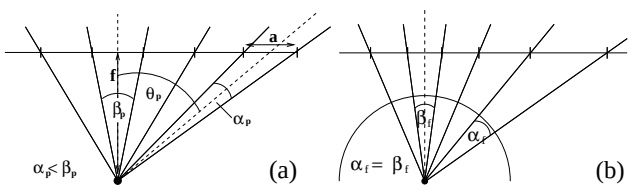

Figure 2. Angular resolution of a perspective camera (a) and a fisheye camera (b).

An optimal fisheye lens gives a circular picture. The angle between the cameras optical axis and a ray through a 3D point is linear to the distance of the focal point to the 3D points projection in the camera image, so for a fisheye lens the angular resolution is constant (see fig. 2(b)). In a per-

spective camera the angular resolution is higher for a greater angle θ_p, points closer to the image border (see fig. 2(a)).

To derive an error model for the perspective camera, it is necessary to calculate its angular resolution. Deriving $f/a\ \sin(\alpha_p) + \sin(\alpha_p/2) = \cos^2\theta_p$ from figure 2(a), with $f >> a$ we can approximate

$$\alpha_p \approx \text{asin}(\frac{a}{f}\cos^2\theta_p) = \text{asin}(\frac{2\tan(\theta_{max})}{N}\cos^2\theta_p), \quad (1)$$

where α_p is the angular resolution of the perspective camera, θ_p is the angle between 3D point ray and cameras optical axis, f is the focal length, a is the size of a single CCD-Pixel, N is the full CCD-resolution and θ_{max} is the half FoV. The approximation error rises with θ_{max}, but for $\theta_{max} = 80°$, which is the inspected range for perspective cameras in this work, the Error is still $< 0.6°$. With (1) we can compute the angular resolution of each pixel from its position on the chip.

The angular resolution of a fisheye camera (see fig. 2(b)) is much simpler written as

$$\alpha_f = \frac{2\theta_{max}}{N}. \quad (2)$$

The angular resolution of the fisheye camera is constant, while the one of the perspective camera rises to the image borders. The functions for α_p and α_f are plotted in figure 3 for fisheye and perspective cameras with different FoVs.

The center angular resolution for a perspective camera with a wide FoV is very bad, so from the same Gaussian error on the pixel position follow two different models for the angular errors, which directly affect the scene reconstruction quality.

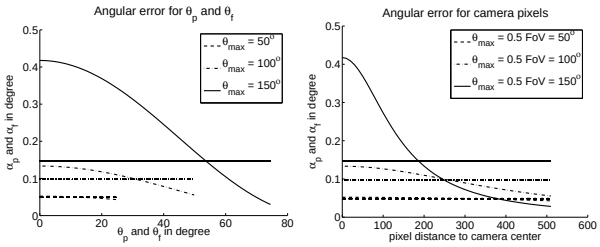

Figure 3. Angular res. for different FoVs.

Reconstruction quality The camera FoV has a great effect on the reconstruction error, the lower the cameras FoV the higher is its angular resolution for a constant number of pixels. On the other hand with a wide FoV the trackable features have a better spacial distribution which is necessary for a stable pose estimation, since only points perpendicular to the FOE can be estimated most reliable. To simulate the effects of a changing FoV a critical and a non-critical camera movement was chosen. For the non-critical y-y-movement the camera moves twice in y-direction. For the critical z-y-movement the camera moves first in z-direction (the FOE is in the image center), and then in y-direction. Also diagonal movements were tested, these results are not shown here, as they are in between of the 2 presented movements.

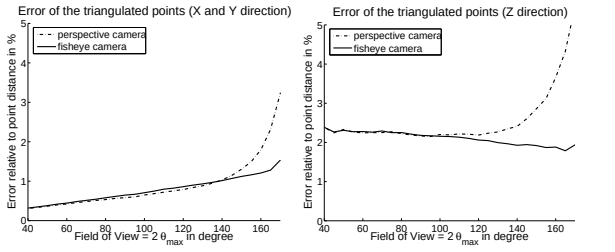

Figure 4. Error of triangulated points from cameras 1 and 2 (z-displacement).

From the first two camera positions the camera pose is estimated up to scale and 3D points are triangulated. The errors of the triangulated points for the critical z-y-displacement are shown in figure 4, the curves show an average of 100 runs on the same test data with Gaussian noise applied. The standard deviation is Gaussian distributed and omitted. The results for x and y are mostly similar and get worse with an increasing FoV, according to the decreasing angular resolution. The perspective camera z-error ascends fast for a greater FoV, because of the cameras bad angular resolution at the center. The fisheye error characteristic is much better suited for triangulation using a wide FoV camera. The resolution degradation is compensated by the good estimation of points lying at the image borders, perpendicular to the FOE.

From the estimated points and their known projection into the third camera it is possible to reconstruct its pose. The reconstruction error is given in figure 5. The third cameras pose prediction depends very much on the triangulated points quality and also on the number and the spatial distribution of the points used for the estimation.

The pose quality depends very much on the camera movement. The optimal fisheye lens has a FoV of $140° - 160°$, with the best pose estimation for all analyzed movements. This is due to the fact that the number of well triangulated points for the z-y-movement gets higher, the more points perpendicular to the FOE are used for pose estimation. In all cases and for all tested movements the fisheye lens performs similar or better then the perspective lens. For FoV $> 100°$ the fisheye lens is much superior.

The optimal perspective camera for SfM has a FoV of approx. $70 - 80°$. In this range the estimation gives good results for all movements. For sideways movements the estimation gets better with a rising FoV with good values from a FoV $> 70°$. For the z-y-movement the pose quality degrades fast (see Fig. 4), analog to the triangulated points quality, and from a FoV $> 80°$ the results get really bad.

Lens selection From the preceding analysis follows, that a fisheye camera with a large FoV is superior for SfM. There

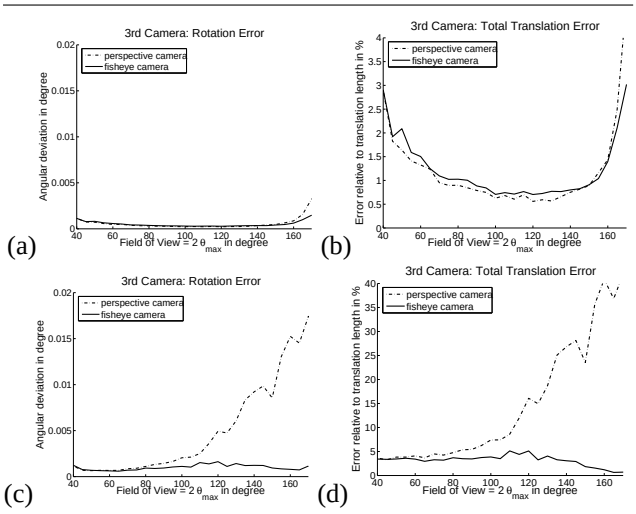

Figure 5. Camera 3 pose error, (a),(b) y-y-displacement , (c),(d) z-y-displacement.

are additional advantages for the use of a fisheye camera that can hardly be modeled.

At first the large FoV achieves robustness even in the presence of moving objects. The fisheye camera covers a wide scene area and moving objects tend to be only in certain regions, i.e. in the working area in front of the user. So there are always many static features in the camera view, i.e. at the sides or in front above the working area.

Secondly a camera mounted on a human head is subject to large and jerky rotations. These rotations are partially compensated by the rotation sensor, but still the camera may rotate quickly out of view. This will not happen easily with a hemispherical view.

As a drawback the system is mainly designed for indoor use, in outdoor scenes the sun light falling directly onto the CCD sensor is causing problems. A cloudy sky can also cause trouble, since a huge scene part is moving slowly enough to be still trackable. These problems can be facilitated by using CMOS sensors with logarithmic response and high dynamic range together with a configuration where the fisheye camera is mostly looking to the ground.

Based on the above research we have chosen a 640x480 camera with with a 12 mm microlens fisheye. The CCD chip size is chosen to reduce the FoV in y-direction to 160° and a quadratic sub-image with 400x400 pixel is processed. The resulting angular resolution is $3\ \mathrm{pix}/^\circ$.

4. Experiments

We have performed extensive experiments with the presented system. To evaluate the timing, 80 features were tracked on a 400x400 pixel image using a 3.0 GHz P4 single and double processor PC. With 1 CPU the tracking uses 133 ms per frame (7.5 Hz). In 2 CPU mode and separated 2D and 3D tracking running in parallel the time per frame drops to 93 ms (10.8 Hz) for the 3D pose reconstruction. But since the 2D tracking runs at 30 Hz in parallel mode, the 2D part processes 3 frames while 3D pose is estimated for only one frame. This stabilizes the whole system since a high 2D tracking rate keeps the differences between the tracked images small.

5. Conclusions

The presented approach shows that robust markerless 3D tracking from a fisheye camera system is possible in real-time. Also a detailed analysis is given that shows the superiority of the fisheye camera for the SfM approach in context of an augmented reality system.

There is still optimization potential in the 3D processing speed as well as in the algorithm design. I.e. there is currently no feedback from the 3D features into the 2D stage, which would further stabilize the 2D tracking. Also one could think of a higher weight for 3D points perpendicular to the FOE to improve the pose estimation.

Acknowledgement: This work was supported by the Federal Ministry of Education and Research project ARTESAS (www.artesas.de) and the EU project MATRIS IST-002013 (www.ist-matris.org).

References

[1] R. Azuma, Baillot, Behringer, Feiner, Julier, and MacIntyre. Recent advances in augmented reality. In *IEEE Computer Graphics and Applications, Vol. 21 , No. 6*, pages 34–47, Nov. 2001.

[2] A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings International Conference Computer Vision, Nice*, 2003.

[3] A. J. Davison, Y. G. Cid, and N. Kita. Real-time 3D SLAM with wide-angle vision. In *Proc. IFAC Symp. on Intelligent Autonomous Vehicles, Lisbon*, July 2004.

[4] A. J. Davison, W. W. Mayol, and D. W. Murray. Real-time localisation and mapping with wearable active vision. In *Proceedings of the IEEE International Symposium on Mixed and Augmented Reality, Tokyo*, 2003.

[5] M. Fleck. Perspective projection: the wrong imaging model. *Technical Report 95-01, Comp. Sci., U. Iowa*, 1995.

[6] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge university press, 2000.

[7] J. Neumann, C. Fermüller, and Y. Aloimonos. Eyes from eyes: New cameras for structure from motion. In *IEEE Workshop on Omnidir. Vision*, pages 19–26, 2002.

[8] M. Pollefeys, R. Koch, and L. J. V. Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. *International Journal of Computer Vision*, 32(1):7–25, 1999.

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