Analysis by Synthesis Techniques for Markerless Tracking

Martin Schumann, Sabine Achilles, Stefan Müller

Universität Koblenz-Landau
Institut für Computervisualistik
Arbeitsgruppe Computergraphik
Universitätsstrasse 1
56070 Koblenz
Tel.: +49 (0)261 / 287 - 2727
Fax: +49 (0)261 / 287 - 2735
E-Mail: {schumi, sachilles, stefanm}@uni-koblenz.de

Abstract: In contrast to knowledge based computer vision approaches, where 3D information is exploited to enhance the tracking process, we render synthetic images based on an estimated camera pose (the pose of the last frame or delivered by additional coarse tracking devices). Comparing the synthetic image with the image provided by the tracking camera finally yields the requested camera pose. While computer vision approaches for markerless AR tracking typically start with an image as a list of features (pixel intensities, corners, lines, descriptors) without additional knowledge about the image generation process behind, the physical process of illumination and light material interaction is very well understood in computer graphics and can be simulated with hundreds of frames per second. Combining both research areas by rendering a synthetic image of the scene provides additional information for each pixel that can improve the generation and selection of the most significant features for stable tracking.

Keywords: Augmented Reality, Markerless Tracking, Analysis by Synthesis, Rendering

1 Introduction

Many approaches in tracking derive the movement of the camera by using markerless tracking techniques, mostly based on the examination of changes in corresponding features between succeeding frames of the video input. Features in the video frames are detected with methods of image processing by using the pixel information of the image in form of differences in neighboring pixel intensities. Thus building correspondences between features in succeeding frames becomes difficult and ambiguous. Disturbing lighting conditions and occlusion of features may stay undetected and thus it is not sure that a feature can be found again in the next image and will be followed steadily over time.

The problem may be reduced by generating a very huge amount of features, weakening the influence of the erratic ones on the result. A better procedure would be to regard an optimized, noise-free image as a reference for every single camera image. An interesting approach to markerless tracking is the strategy of Analysis by Synthesis, in which the environment to be tracked is

represented by a 3D-model that can deliver information to create good features. With computer graphics methods and well known conditions while rendering a synthetic image, we can simulate various rendering parameters and properties of the real environment for dynamically adapting to changing environmental conditions. Thus it is possible to select only those features, bearing the best information for the realization of stable tracking in the present situation.

The synthetic reference image not only permits better feature detection but also avoids common disadvantages on frame-to-frame tracking like drift, occlusion of features, changes in lighting conditions or the initial tracking problem by locating the local camera coordinate system into the world coordinate system. The aim is to improve feature based tracking with all the information the computer graphics render process can deliver. On the other hand, this approach raises a number of questions like: quality of the tracking results, how does uncertainty and incompleteness of the 3D model influence the robustness and quality, what about dynamic scenes, how much photorealism is needed in rendering etc.

In this paper we present our first results in this area by focusing on two different approaches for the comparison of synthetic and camera images: the feature-based and the similarity-based one. In the first case the general suitability of using feature correspondences between synthetic and camera image has been regarded. We analyzed if common feature detectors used in image processing are able to correctly detect the same features in a rendered image as in a camera image. The second work realizes tracking without features, comparing global similarity of synthetic and camera image. We present very optimistic results that proof the potential of the Analysis by Synthesis approach. However, the results are not yet optimal and we will outline research possibilities how the potential of the rendering process can be further exploited to deliver robust and high quality tracking results.

2 Discussion on related work

When using the recursive frame-to-frame approach of markerless tracking, drift occurs due to summing up errors. One possible solution is to use reference images of different views of the environment [Str01]. These keyframes have to be taken from known positions in advance and are stored in a database. With Analysis by Synthesis the preparatory procedure of creating an image database can be replaced by rendering reference images online. In further approaches a learning step is applied by recording the environment beforehand and building a reference feature model with the help of markers [GRS$^+$02]. This method allows reinitialization after loss of tracking but is limited to the area covered in the learning stage. In [GL04] the user takes several reference images of the environment before tracking and the scene structure is discovered by methods of Structure from Motion. With the approach of Analysis by Synthesis none of these preparation steps are necessary before the start of the tracking process.

Current research is focused on SLAM (Simultaneous Localisation and Mapping) algorithms. In AR, this method allows the creation of maps consisting of feature edges or point clouds on the visible surroundings. Scene geometry and camera postition are derived from keyframes in the

image sequences by reconstruction. The enlargement of the map (extensible tracking) is possible in case the camera enters an unknown surrounding. In [KM07] an accumulation of thousands of such points as low quality features is shown. Due to the enormous amount of data created, this method is yet limited to small scenes and struggles with occlusion problems when not recognizing self-occlusion of already recorded features. Particularly corrupt map entries of features, caused by matching false correspondences, are not managed and affect tracking negatively. Depth reconstruction by triangulation introduces uncertainty and it is also necessary for triangulation to get sufficient translation of the camera, which is a problem, if the camera is not moved right from the beginning of the tracking process. The approach of using a known model permits a direct and safe extraction of exact world coordinates out of the model, which even covers occlusion problems and gives absolute reference coordinates from initialization instead of relative ones.

Most model-based methods are constructed on lines and connected structures which are projected into the camera image. The camera pose is derived from minimization of the distance error between these projections and strong gradients in the image. [CMC03] introduced tracking on a CAD model describing complex structures of the tracked object for matching point-to-point correspondences in the image. This work has already made it obvious that knowledge about the scene can contribute to improvements in tracking, concerning stability and speed. Respectable successes in tracking a line model in combination with point features could be demonstrated by [VLF04]. [WS07] used the contour lines of a 3D model for tracking. These were aquired by taking the depth buffer of the rendering pipeline to extract discontinuities with the help of an edge detector, which shows that the synthesis process of computer graphics can contribute valuable information to tracking.

Another example for the application of Analysis by Synthesis in tracking is given by [KBK07] who build a free-form surface model of the scene beforehand by using a fisheye camera and Structure from Motion reconstruction. Thus a synthetic fisheye image is rendered from the model to compare with the real camera image and to minimize the difference for pose estimation. The advantage is that a fisheye camera has a large field of view and can track features over a longer time than a perspective camera does. [SJBP99] realized tracking and modeling of faces with Analysis by Synthesis while using computer graphics information as normals and depth.

3 Analysis by Synthesis

The only prerequisite for the Analysis by Synthesis approach is that a 3D model has to be available for rendering. Such models are usually often present in assembly, installation, and maintenance scenarios. It is also conceivable that in the near future 3D models of whole cities and particular touristical attractive buildings will be available (like Google Earth) or can be made available easily with small effort. As additional information to the surface models the attributes of materials can be annotated, but may also be detected automatically for the diffuse case [RG06] and it is even possible to automatically aquire the textures of reflection degrees on objects by subtracting current lighting conditions. Advances in the area of photorealistic image synthesis make it possible to get

all necessary lighting information. An HDR camera is able to reconstruct the real lighting situation [HSK+05]. Serveral rendering methods have been developed to cast precise shadows and display complex materials in real time with the help of the GPU [RGKM07]. Even ray-tracing methods have been extended under aspects of real time capability and to simulate global light effects in dynamic scenes ([BAM08],[SAM07]). In [GEM07] and [KSvA+08] lighting estimation of the far field has been extended by near field lighting effects and indirect lighting effects. Building upon these works, it is possible to render a virtual scene with complex material attributes under consideration of current lighting in real time and with photometrical and colormetrical consistence.

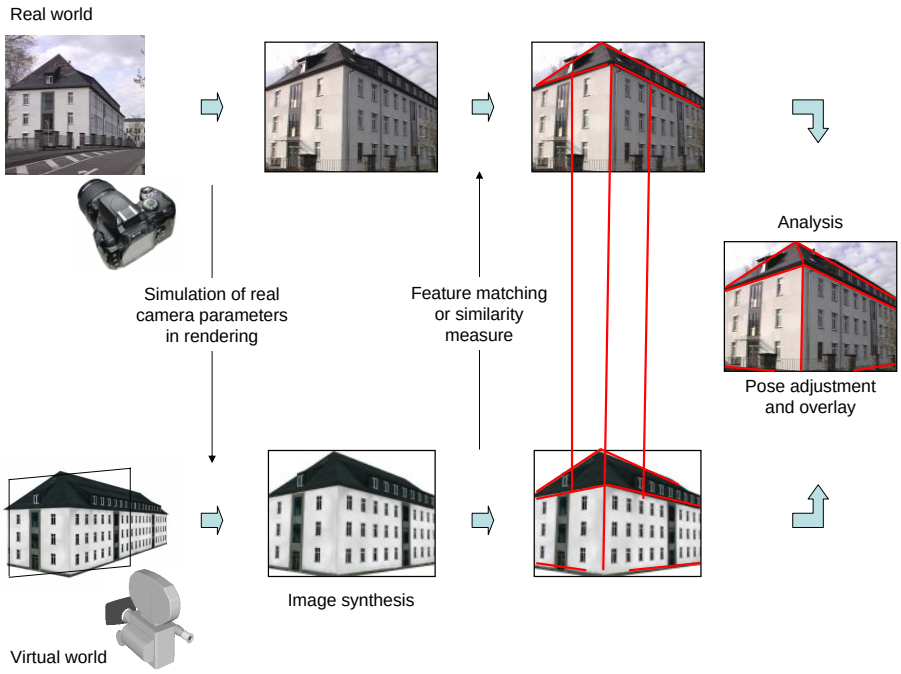

Figure 1: Analysis by Synthesis

Beginning from an initially estimated pose (as by GPS outdoor or rough tracking indoor) or the pose of the last image, an image of the given 3D model is rendered (synthesis) and compared to a real camera image (analysis) to estimate the current pose of the camera (Fig.1). The approach of Analysis by Synthesis may later be enhanced to not only delivering a synthetic reference frame for pose estimation, but also to help improve the process of feature detection itself, as we will see in the future work section.

The aim is to find the unknown parameters of the real camera image by means of known parameters when rendering the synthetic image. The parameters to optimize can be described as a pose-vector $p = (t_x, t_y, t_z, r_x, r_y, r_z)$ that consists of the variables for position and orientation of the camera, the translation and rotation. In the feature-based approach (Fig.2 left) the pose is estimated by error minimization between features detected in the synthetic image and their matches in the camera image. Given a set of features f_r in the synthetic image, which is rendered from the last camera pose p and the corresponding features f_c in the camera image, the error E between camera features and synthetic features after reprojection must be minimized to retrieve the new camera pose $\tilde{p}$:

$$\tilde{p} = \underset{p}{\operatorname{argmin}} E(f_r(p), f_c)\,. \tag{1}$$

Another possible way to realize Analysis by Synthesis for pose tracking is following a more intuitive optimization method apart from using features. In the similarity-based approach (Fig.2 right) the virtual pose is varied in small steps around the last correct pose to render several slightly different synthetic images for comparison with the camera image. The optimization is realized by an iterative correction of the known virtual pose on the synthetic image to approximate the real camera pose. A measure of similarity can determine the rendered image with highest correspondence to the camera image and its virtual pose can then be regarded as valid corrected pose for the real camera situation. Given a rendered synthetic image R with its known virtual camera pose p_r and the camera image C with the pose p_c to be retrieved, p_r must be optimized until the measured similarity S between R and C is maximized:

$$p_r = \underset{p_r}{\operatorname{argmax}} S(R(p_r), C(p_c)) \tag{2}$$

$$until$$

$$R(p_r) \approx C(p_c) \;\; \Rightarrow \;\; p_r \approx p_c\,.$$

Two preliminary projects were realized to lay the foundations for further research of Analysis by Synthesis. We analyzed how methods of computer graphics can improve or support methods of image processing used in the context of an optical markerless tracking system.

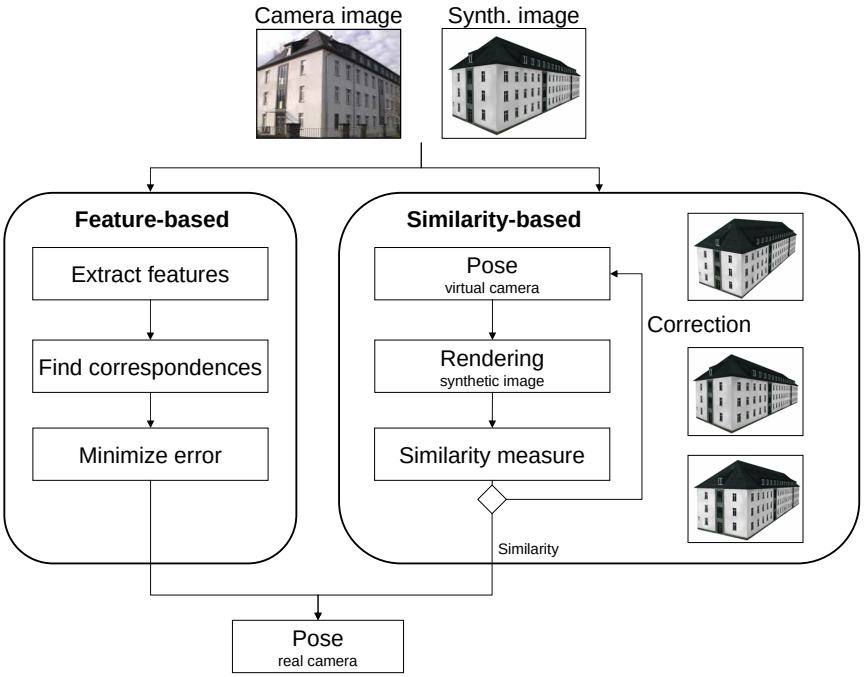

Figure 2: Two methods on tracking

3.1 Feature-Based Pose Estimation

The feature-based approach realizes tracking by finding corresponding features in a camera and a synthetic image of a rendered 3D-scene to estimate the camera pose. Therefore, an analysis of feature detectors, that are suitable to find matching features in both images for best results in this context, has been done. Further tests concern the aspect of the level of detail in the rendering process and the influence of textures as well as the importance of light for the exactness of the tracking results. The following common operators are applied: The Harris Corner Detector, Kanade-Lucas-Tomasi Detector (KLT), the Smallest Univalue Segment Assimilating Nucleus (SUSAN), Features from Accelerated Segment Test (FAST), the Scale Invariant Feature Transform (SIFT) and the Foerstner Operator.

In the synthetic image and the camera image features are detected and matched. Errors in the located correspondences are eliminated by applying the Random Sample Consensus algorithm (RANSAC). The 2D features found in the synthetic image are then reprojected onto the model to determine their 3D world coordinates. These resulting 2D/3D feature correspondences are used to approximate the pose of the camera in the current video frame using a robust M-Estimator (Tukey Estimator) with Downhill-Simplex optimization. In every iteration of the pose estimation process, the reprojection error between the 3D features projected onto the image plane and the 2D features in the video frame is determined repeatedly, summed up and minimized.

The correspondence matching was implemented by two different distance measures: the Normalized Cross Correlation (NCC) and the Normalized Sum of Squared Differences (NSSD). In both cases the descriptor was limited to a vector containing the gaussian weighted pixel neighborhood of the feature to be analyzed. An exception poses the implementation of SIFT, in which case the SIFT descriptor was used. Generally speaking, a broader neighborhood leads to more stable results but slows down the algorithm. The experiments determined a best fit size for the chosen neighborhood of around 9x9 to 11x11 pixels. Some results of located correspondences are shown in Fig.3. An assumption of small camera movements with only little changes between two images can reduce the search window for the corresponcences to a 30x30 pixel neighborhood (at a resolution of 720x540 pixels). This speeds up the computation time and avoids false matchings.

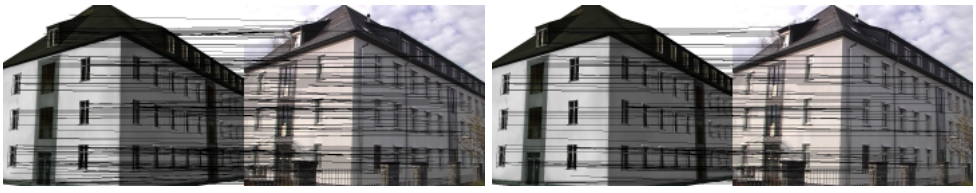

Figure 3: SIFT correspondences between rendered and camera image with limitation to a 50x50 pixel search window (left) and under use of RANSAC (right).

The reprojection error was measured in pixel for every frame of a 50 frame tracking sequence, then summed up and averaged as root mean squared error (RMSE). In general, the pose can be estimated better the more correspondences are found, but (depending on the used feature detector) only about 10-20% of the features detected in both images can finally be matched to correspon-

dences (Table1). Time consumption is listed for every feature detector, running on CPU. They are not optimized and thus lead to tracking frame rates between 0.2 fps (SIFT) and 1 fps (Harris). Better performance results can be expected with GPU-based implementations.

	features synth. image		features camera image		resulting correspondences
Harris	800	17ms	800	18ms	125
SIFT	450	1,5s	2200	3,3s	120
KLT	800	23ms	800	25ms	110
Foerstner	900	55ms	1200	55ms	90
SUSAN	530	39ms	1200	49ms	60
FAST	830	7ms	1000	7ms	50

Table 1: Number of features detected and correspondences matched

The test results show that FAST can not establish a sufficient number of point matches for tracking when using a synthetic image. Although enough features are found, only about 6% of them result in matched correspondences. The SUSAN detector only found about half of the features in the synthetic image, that were detected in the camera image. This reduces the basis for possible matches and therefore leads to a low number of point correspondences. The other feature detectors are able to deliver a number of correspondences acceptable for tracking. It should be remarked, that Harris-Corner-Detector and KLT gain their good results by detecting an equal number of features in synthetic and camera image, whereas SIFT proves to be good in establishing correspondences out of a strongly varying number of features in both images.

In addition, the distribution of the features has to be even enough throughout the two images to obtain good tracking results. This is not the case with SUSAN, whose detected features show a large amount of clustering. While KLT can generate enough correspondences for tracking, their distribution does not suffice for stable tracking. Feature detectors showing an adequate number of correspondences as well as good distribution are Harris-Corner-Detector, SIFT and Foerstner.

A comparison of the error by different feature detectors throughout the whole video sequence (using NCC matching algorithm with RANSAC support) can be seen in Fig.4. At slow camera movements the exactness hardly profits from RANSAC usage, because there are only few wrong correspondences due to the limited search space. When the sequence gets to a point of fast camera movement at the end, the NCC error grows slightly and almost all detectors benefit from RANSAC eliminating the outliers. We also tested NSSD matching, which has little advantage in calculation time but is obviously insufficient for fast changing image content, due to fast growing errors.

All feature detectors were also tested on the influence of lighting and texture. The NCC algorithm is intensity invariant and the change of ambient light has only a slight effect on the matching results with advantages by SIFT and FAST. The tests showed, that the direction of the light has to be set as exact as possible to avoid tracking errors, especially in outdoor scenarios with fast changing light situations. Wrongly simulated shadows make it difficult to find unique features, because shadow edges are likely to be undistinguishable from real physical edges. The problem gets even worse if the model of the scene does not map reality close enough, as was the case with the model used in the testing stage. Rendering the model without texture shows, that only Foerstner does

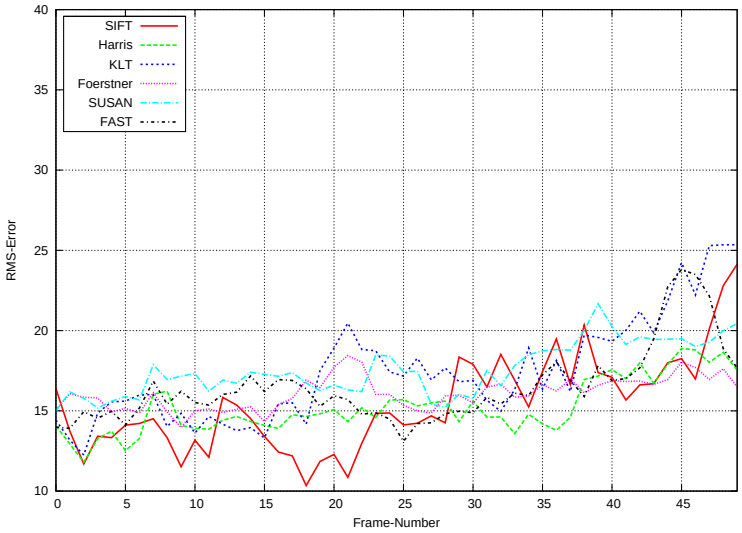

Figure 4: Error over image sequence on NCC matching

not seem to profit from the additional information given by the texture. SIFT, SUSAN and FAST obviously benefit from using textures to detect more correct features. Textures contain substantial information for detecting features, therefore geometry and texture of the model should be as accurate as possible. The result of an inaccurately modeled scene is jittering, e.g. when distances and scales are not consistent.

Acceptable results for an Analysis by Synthesis approach were gained by SIFT, Harris Corner Detector and Foerstner-Operator (using NCC, RANSAC, texture, correct lighting). FAST could be optimized to detect a sufficient number of point features, but at the cost of creating more false correspondences and slowing down the NCC matching too much. However, the test also showed that the used descriptors are not optimal for Analysis by Synthesis tracking. As an example, features on a shadow-edge in the real image are likely to be matched to features on a model-edge in the synthetic image. The tracking could benefit from a descriptor especially designed for the use in Analysis by Synthesis, using information on the model and lighting situation to discriminate edges from each other. The cost of a simultaneous feature detection in both images could be decreased by annotating the model with precalculated Analysis by Synthesis features, which can be rendered directly and would then only have to be matched with the camera image.

3.2 Similarity-Based Pose Optimization

The similarity-based approach realizes tracking without detecting features, instead measuring the similarity between camera image and synthetic image. Open questions concern the qualitative demand for rendering to realize a tracking robust against errors. Especially the degree of detail in the rendering (realistic rendering with and without shadow, abstract rendering like Toon-Shading, Gooch-Shading or Sobel-Edge-Images) and suitable similarity measures for best comparison with the camera image have to be regarded. The parameters to be optimized are translation and rotation values in the pose-vector of the virtual camera. To sample the space around the last camera pose for possible movements of the camera, these parameters are varied adaptively by stepping in intervals

around the last pose. Thus some slightly different images of the scene are rendered. Due to the impossibility to render an infinite number of images from various poses, the search space has to be restricted.

As a simplification for every optimization step $2n$ new poses are generated around the pose of the last maximum similarity, where n is the number of degrees of freedom on the parameters to be determined. Thus we render 6 synthetic images for translation along the three coordinate axes (for each translation in positive and negative direction), 6 for rotation in the same scheme, and one image from the current pose. These intervals can be chosen adaptively to span diverse search-windows, taking different accelerations in movement into account. When the camera moves slowly, which results in high measured similarity between the renderings and the camera image, step size is chosen small. Accordingly at faster camera movement, resulting in lower similarity, a wider step size is used. After comparison with the camera image, the parameters of the pose-vector with the best similarity function result are used as the starting point to generate the next poses, resulting in the direction of the virtual camera movement to be followed. The initial camera pose is expected to be known approximately.

With a similarity measure to compare the rendering and the camera image, neither extraction nor search for correspondences is necessary because the information on the position of the pixel values is independent from content and structure of the image. The whole image can be evaluated in one step and preprocessing is omitted. Pixel values of two images corresponding in their co-ordinates can be compared directly pairwise where the observation of difference or correlation is possible. We used the *Sum of Squared Differences (SSD)* and the *Normalized Cross Correlation (NCC)* for testing similarity of two images f and g:

$$d_{NCC}(f,g) = \frac{\sum_{x,y}(f(x,y)-f_\mu)*(g(x,y)-g_\mu)}{\sqrt{\sum_{x,y}(f(x,y)-f_\mu)^2}\sqrt{\sum_{x,y}(g(x,y)-g_\mu)^2}} \quad . \tag{3}$$

The average-free Normalized Cross Correlation (NCC) is in advantage due to its insensibility on lighting variations, reducing noise that influences the stability of the similarity values. Compared to the distance measure (SSD), its values showed better distribution in a normalized similarity range between 0 and 1, leading to higher precision.

The case of realistic rendering showed best results in reaching highest stable similarity values. Using abstraction methods for the image content showed limits on complex textured models and proved unsuitable. While tracking was precise enough in translation, rotation values became worse when abstracting from reality. Repeating structures (like windows) cause noise and introduce jittering. Most influential on the quality of tracking was the positioning of the virtual light source. Unprecisely adjusted light direction and wrong shadowing lead to errors in pose tracking.

Incorrect shadowing introduced rotation errors up to 2 degrees. Better performance could be achieved by using HDR light source tracking with a higher number of virtual light sources for sampling the real lighting situation to prevent a bad quality of virtual lighting as cause for errors. For a 10 cm untextured object using realistic rendering the average ground error was 0.1 cm and 0.3 degrees. On constant camera movement over distances of 60-80 cm and rotations of 90 degrees it grew to 0.5 cm and 1 degree in average, depending on the optimization sampling step size chosen.

Figure 5: Real and synthetic image of test objects

Reference data for error measuring was delivered by ARToolKit. The process of rendering one synthetic image of an untextured scene, storing it to texture, and comparing it to a 640x480 camera frame takes about 7 ms on CPU. For rendering 13 images and including processing of the camera image, without further optimization the tracking process delivers ~ 3 fps.

4 Summary

In this paper we focused on two different pose estimation methods for Analysis by Synthesis: The feature-based and the similarity-based approach. Both methods are not yet optimal, since we are using elementary feature detection and correspondence techniques. However, the results are convincing enough to prove the concept and the potential of Analysis by Synthesis supported by computer graphics methods.

Both approaches deliver comparable results and interactive frame rates on a CPU implementation without further optimization. The similarity based approach seems to be faster, while the feature based approach seems to be more robust. We achieved best results with SIFT, Harris and Foerstner features under NCC matching on textured models with the feature based approach. The similarity based tracking delivered best results on untextured objects (with NCC), which makes it appropriate for a rough tracking step. Light source tracking and realistic shadow simulation proved important for the exactness of tracking in both approaches and especially for the avoidance of false matchings when using common feature detectors.

As a general result of our tests it became obvious, that classical feature detection known from image processing is not optimal to build correspondences between a synthetic image and a real camera image. With only up to one fifth of the features found leading to matches, less correspondences than expected are established and error tolerances have to be chosen much higher accordingly.

5 Future Work

To conclude, in further research new possibilites should be developed for detection and prediction of features, that can use the knowledge of model and environment for generating a sufficiently small amount of unique correspondences between synthetic and camera features. Therefore, our future focus will be on the generation and selection of features, which can be highly prioritized during the process of rendering the model by exploiting attributes in form of topological information, lighting information or perspective representation.

Analysis by Synthesis provides a more complete range of information for each pixel: Attributes as the depth value, face of the model, normal and difference of normals on neighboring faces, texture and attributes of assigned material, lighting with dynamic reconstruction by an HDR fish-eye camera and shadow, as well as occlusion and real size of a feature in object space and its distance after projection in image space, may be determined easily. In Fig.6 an overview of possible parameters for prediction is given. Long term vision is the development of a new feature-renderer with small priority windows in the image, highlighting where the feature is found, annotated by a proposed feature detector which is most appropriate in order to find optimal correspondences.

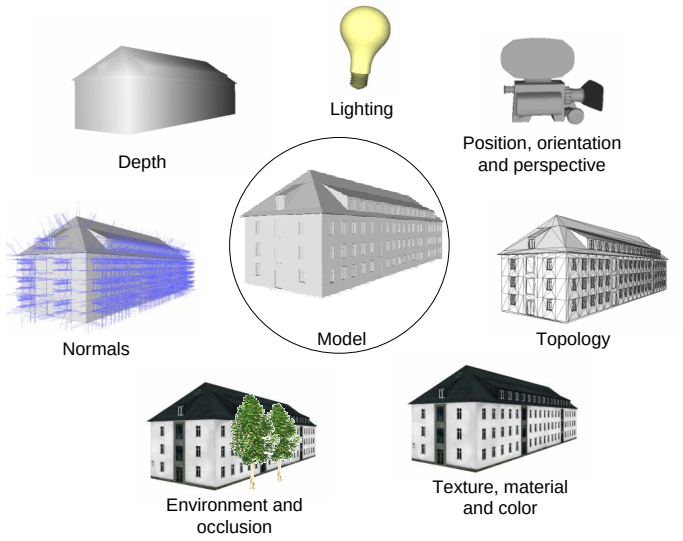

Figure 6: Sample parameters for finding features

Acknowledgements

This work was supported by grant no. MU 2783/3-1 of the German Research Foundation (DFG).

References

[BAM08] J. Baerz, O. Abert, and S. Mueller. Interactive particle tracing in dynamic scenes consisting of NURBS surfaces. In *IEEE/EG Symposium on Interactive Ray Tracing*, 2008.

[CMC03] A.I. Comport, E. March, and F. Chaumette. A real-time tracker for markerless augmented reality. In *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 36–45, 2003.

[GEM07] T. Grosch, T. Eble, and S. Mueller. Consistent interactive augmentation of live camera images with correct near-field illumination. In *ACM Symposium on Virtual Reality Software and Technology (VRST)*, 2007.

[GL04] I. Gordon and D. G. Lowe. Scene modelling, recognition and tracking with invariant image feaures. In *3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 110–119, 2004.

[GRS+02] Y. Genc, S. Riedel, F. Souvannavong, C. Akinlar, and N. Navab. Marker-less tracking for AR: A learing-based approach. In *International Symposium on Augmented Reality (ISMAR02)*, pages 295–304, 2002.

[HSK+05] V. Havran, M. Smyk, G. Krawczyk, K. Myszkowski, and H.-P. Seidel. Importance sampling for video environment maps. In *ACM SIGGRAPH Eurographics Symposium on Rendering*, 2005.

[KBK07] K. Koeser, B. Bartczak, and R. Koch. An analysis-by-synthesis camera tracking approach based on free-form surfaces. In *29th annual pattern recognition symposium of Deutsche Arbeitsgemeinschaft für Mustererkennung (DAGM)*, pages 122–131, 2007.

[KM07] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *International Symposium on Mixed and Augmented Reality (ISMAR07)*, 2007.

[KSvA+08] M. Korn, M. Stange, A. von Arb, L. Blum, M. Kreil, K.J. Kunze, J. Anhenn, T. Wallrath, and T. Grosch. Interactive augmentation of live images using a HDR stereo camera. *Journal of Virtual Reality and Broadcasting (JVRB)*, 2008.

[RG06] T. Ritschel and T. Grosch. On-line estimation of diffuse materials. In *3rd Workshop Virtual and Augmented Reality of the GI-Group VR/AR*, 2006.

[RGKM07] T. Ritschel, T. Grosch, J. Kauz, and S. Mueller. Interactive illumination with coherent shadow maps. In *Eurographics Symposium on Rendering (EGSR07)*, 2007.

[SAM07] F. Scheer, O. Abert, and S. Mueller. Towards using realistic ray tracing in augmented reality applications with natural lighting. In *4th Workshop Virtual and Augmented Reality of the GI-Group VR/AR*, 2007.

[SJBP99] J. Strom, T. Jebara, S. Basu, and A. Pentland. Real time tracking and modeling of faces: An EKF-based analysis by synthesis approach. In *IEEE International Workshop on Modelling People*, page 55, 1999.

[Str01] D. Stricker. Tracking with reference images: A real-time and markerless tracking solution for out-door augmented reality applications. In *Virtual Reality, Archaeology, and Cultural Heritage (VAST2001). Glyfada, Greece*, 2001.

[VLF04] L. Vacchetti, V. Lepetit, and P. Fua. Combining edge and texture information for real-time accurate 3D camera tracking. In *International Symposium on Mixed and Augmented Reality (ISMAR04)*, pages 48–57, 2004.

[WS07] H. Wuest and D. Stricker. Tracking of industrial objects by using CAD models. *Journal of Virtual Reality and Broadcasting*, 4(1), 2007.