MULTI-TEXTURE MODELING OF 3D TRAFFIC SCENES

Karsten Mueller, Aljoscha Smolic, Michael Droese, Patrick Voigt, and Thomas Wiegand

Fraunhofer-Institut für Nachrichtentechnik, Heinrich-Hertz-Institut

ABSTRACT

We present a system for 3D reconstruction of traffic scenes. Traffic surveillance is a challenging scenario for 3D reconstruction in cases, where only a small number of views is available that do not contain much overlap. We address the possibilities and restrictions for modeling such scenarios with only a few cameras and introduce a compositor that allows rendering of the semi automatically generated 3D scenes. Some of the occurring problems concern camera images, which might show a common background area, but can still differ drastically in lighting effects. For foreground objects nearly no common visual information might be available, as angles between cameras may exceed even 90°.

1. INTRODUCTION

A variety of applications for protection of public and private areas, measuring data or obtaining information for statistical analysis utilize visual surveillance systems. One of these applications is traffic control, where visual information is mainly used for control and surveillance purposes. In scenarios, where cameras monitor traffic density at successive crossings, traffic flow can be optimized by controlling the traffic lights along these crossings. Such problems in traffic surveillance have been previously investigated proposing parameter-based [1] or object-based vehicle tracking [2]. One approach shows a complex scenario in video surveillance including a simple object classification process [3].

We present a 3D reconstruction system that receives 2D images from a number of cameras via a multi server - one client video streaming system. One optimization goal in our application is to get a good coverage of the observed area with as few cameras as possible to keep the costs of the system reasonable. Consequently, these few camera views usually show only very little overlap. Hence, some 3D scene reconstruction algorithms, like voxel-based reconstruction [4], light fields [5], or disparity-based reconstruction [6] cannot be used for such a scenario, because to obtain reasonable reconstruction results they rely on a rather dense camera grid with much overlapping image content.

Our approach consists of a model-based reconstruction method requiring a priori knowledge about the scene geometry and camera calibration. The scene is separated into static parts (streets, sidewalks, buildings, etc.) that are modeled semi-automatically and dynamic objects (vehicles, pedestrians, bicycles, etc.), which are modeled automatically using predefined geometric primitives. All components are combined, using a 3D compositor that allows free navigation within the scene. As a result, normal traffic as well as emergency situations can better be observed and evaluated. The system setup is shown in Fig. 1.

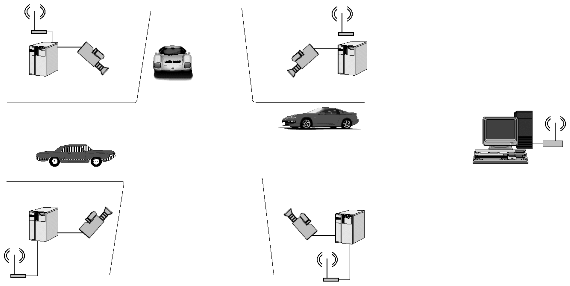

Fig. 1: Surveillance setup with 4 wireless servers and a central client for traffic monitoring and scene rendering

The remainder of this paper is structured as follows: Section 2 describes scene segmentation and calibration. The description of multi-texture background reconstruction is presented in Section 3. Section 4 explains moving foreground object reconstruction. The final 3D scene integration is presented in Section 5.

2. SEGMENTATION AND CALIBRATION

In our approach, static and dynamic scene parts are processed separately. Therefore we first extract moving objects from the static background, using a segmentation-by-motion approach with adaptive background estimation [7]. One benefit of this method is the fast and robust adaptation to environmental changes, e.g., changes in lighting, fog, rain, snow etc., since a background update is carried out for each new frame using a Kalman filter formalism. Fig. 2 shows a segmentation example for one input camera view.

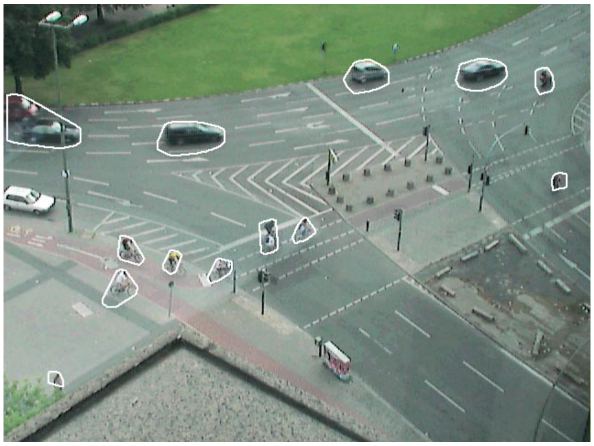

Fig. 2: Detected foreground regions highlighted by convex polygons

To create the background image for each camera view, moving objects are segmented throughout a sequence of 200 pictures and removed from each picture. The holes corresponding to the occluded areas are filled from successive frames, with information that is revealed by the moving objects.

For the projection of object textures in each view into a common plane and to properly position artificial objects in the 3D scene, we make the assumption that correspondences between points in the 3D world and image points are available for each view to obtain calibration information. A common DLT (Direct Linear Transform) algorithm is used to calculate the projection matrices for each view [8].

3. STATIC BACKGROUND MODEL

The 3D model of the scene background contains a ground plane of the traffic scene and additional side planes. These side planes either show surrounding buildings or information that is far away from the place of interest. The original textures from all views are mapped onto a common plane. This is possible, since homography information for projecting data from the 3D scene into each camera view is derived from common corresponding points in all views and the 3D world, as described above. After the segmentation of moving objects from the background, the following processing stages are carried out:

Ground plane/Side plane selection:
In traffic scenes, street and sidewalk areas are the typical dominant parts, which are considered as part of the ground plane. The ground plane is typically surrounded by adjacent areas such as buildings etc, which are considered as part of the side plane. Since the geometrical relationship of ground plane and side plain typically remain unchanged over a long period of time, we perform the geometrical modeling off-line when setting up the system as follows. First the ground plane is selected with an interactive segmentation tool. From this selection, adjacent side areas are highlighted by the tool and the user can choose, which of these areas are to be included into the model. Second, the side areas are mapped onto side planes that are perpendicular to the ground plane, as shown in the 3D background model in Fig. 3. In Fig. 3 the ground plane is composed of textures from all views and side plane textures from adjacent buildings from one view were added.

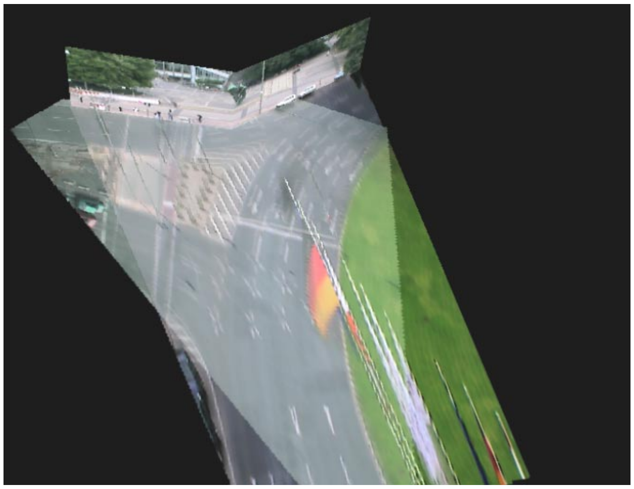

Fig. 3: Background reconstruction with multi-texture ground plane and side plane objects

Other static objects that do not belong to the ground plane:
Beside the required ground plane information, the static scene contains further small and less important elements such as traffic lights, lanterns, or traffic signs. Such elements cannot be eliminated by the initial motion segmentation. Since such elements do not lie in the ground plane they have to be removed. Otherwise the quality of the resulting interpolated image would be poor. Therefore a semiautomatic segmentation step needs to be applied, where color and edge features are utilized to remove such elements from the scene. Currently these elements are not further processed but that is foreseen for the next version of the system.

Lighting and reflection properties:
Typically, large differences between camera positions and viewing angles of the scene cause different illumination of the same areas for most textures. In these cases, a texture might appear bright in one view, whereas in other views it might appear completely dark. This effect has to be compensated if the user navigates through the scene to produce a smooth transition.

DirectX provides tools that enable interpolation and lighting compensation and it is used as rendering framework in this paper. To enable interpolation, we use multi-texture surfaces to model the ground plane as well as the moving objects. These multi-texture surfaces allow mapping of a number of textures onto a single geometric primitive. Different normal vector directions are assigned to each texture, depending on the relationship between the viewing direction and the light source. Then the surface lighting of each texture is determined by the combination of the actual position and viewing direction, the normal vectors of the textures and the position of the light source. The rendered view is interpolated from all textures with their individual weight depending on the actual position and viewing direction in the scene. This enables a smooth transition when navigating the scene as shown in Fig. 4.

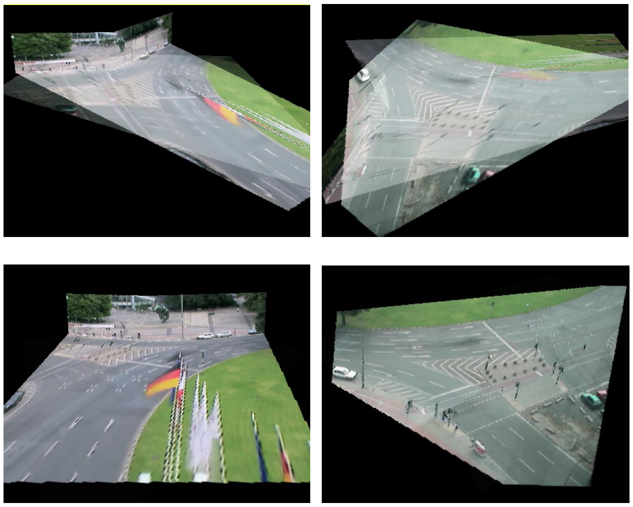

Fig. 4: Background scene from different viewpoints.
***Top*: near the central viewpoint from that is shown in Fig. 3,**
***Bottom*: near the original camera viewpoints**

In Fig. 4, the background scene is shown from different viewpoints. Starting from the viewing position in Fig. 3, where ground plane textures are weighted equally, the viewing position is moved to the left and right, as shown in the left and right top pictures of Fig. 4, respectively. By moving the viewing position away from the middle, one of the textures becomes more visible than the other. When finally the original camera positions are reached, only the ground texture from viewed from the corresponding camera is completely visible as shown in the bottom pictures of Fig. 4, whereas the texture views from other cameras are rendered transparent, i.e. they are not visible. Thus, smooth texture interpolation between different viewpoints is achieved

4. MODELING OF DYNAMIC OBJECTS

The modeling of dynamic objects starts with the motion segmentation described above, and continues with the following two steps:

2D and 3D Tracking:
The algorithm proceeds as follows.

- The motion and shape parameters of the object are estimated by two linear Kalman filters as described in [9].
- Each moving object is assigned an internal label for its identification.
- A moving object from one view can be projected into all other views, using homography information that is calculated in the initial camera calibration step. Here the nearest object is identified by comparing the object from one image to all objects in a particular view. If this object is similar in terms of size and center of gravity, it is marked with the same label. Thus all views of an object are associated with one unique object at a higher semantic level.
- The 3D object position of the moving object is extracted by calculating the 2D center of gravity of an object in each original view from the contour spline representation. We assume that moving objects are relatively small and that the center of gravity therefore lies approximately on the ground plane.
- The center of gravity from each view is projected onto a common ground plane in the 3D scene, as shown in Fig. 5. Usually, the centers of gravity are not projected onto the same point, due to variations in the segmentation process and the described ground plane assumption. In such cases the common center of gravity is approximated using a least squares approach.

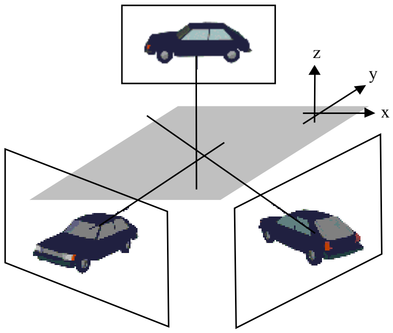

Fig. 5: Center of gravity projection of moving objects onto ground plane in 3D world, using homography information.

- Another linear Kalman filter is used to calculate the corresponding 3D motion trajectory, starting with the 3D center of gravity for each frame. The Kalman filter approach includes a prediction and correction ability, which also allows tracking of objects even if they are temporarily occluded by other objects. The position is determined from the Kalman prediction in such cases.
- The 3D object is positioned along the motion trajectory.

3D Object Reconstruction:
For object reconstruction, the corresponding textures from all views are combined with a suitable synthetic 3D wireframe object from a database. The appropriate position and orientation have already been determined during the tracking procedure leaving only the size of the wireframe model to be adjusted by projecting the artificial model into each view and comparing its 2D bounding box with the bounding box of the segmented textures. After proper sizing, the original textures are mapped onto the object. Fig. 6 illustrates the 3D reconstruction of a moving object, where a 3D wireframe model and 2D texture patches are connected to create the final object. This object is finally integrated into the scene background, considering proper positioning and orientation.

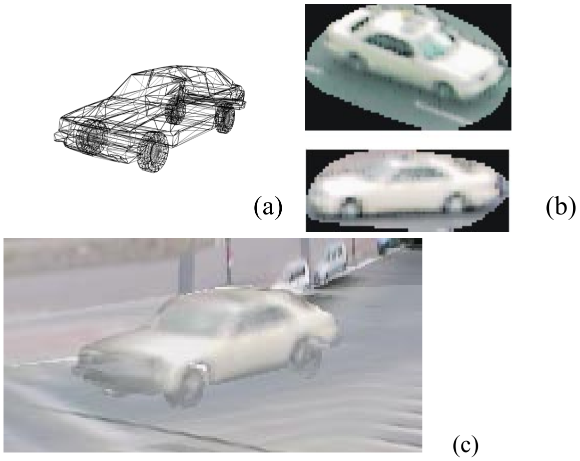

Fig. 6: Synthetic 3D object (a), 2D image patches obtained in tracking procedure (b), reconstructed vehicle placed in the scene background (c)

5. FINAL SCENE INTEGRATION

In the 3D rendering environment, the multi-texture background with additional side views and the dynamic 3D objects are combined to create the final scene. The final scene consists of scene elements with different real time requirements and modeling complexity as follows:

- Object position: The position of each foreground object needs to be updated at each render cycle. This is the most time critical information but requires only very few parameters, i.e. the 3D coordinates of the centers of gravity.
- Object 3D geometry: A 3D geometry needs to be assigned, if an object enters or leaves the scene.
- Object texture: A texture needs to be updated, if an object enters or leaves the scene or if significant new information is revealed in case of exposure or a change of direction.

- Background: This is the scene part with the lowest real-time requirements. Only if the whole scene drastically changes, e.g. due to strong lighting changes, an update of texture is required. Therefore, the modeling algorithm can be completely performed off-line. If the topology of the observed scene changes, a new offline modeling is required.

Since only the object position needs to be updated at every render cycle, more complex algorithms can be applied for processing the other scene elements such as 3D object fitting, texture adaptation and, most of all, for background setup. Furthermore, since intermediate object positions can be interpolated from their motion trajectory, the render cycle time becomes independent of the camera frame rate and the scene can be rendered as fast as possible. The position interpolation for every render cycle is also useful in cases of jitter of the camera frame rate. Fig. 7 shows the final scene with 3 reconstructed cars. Again the central view is shown in the large image, whereas in the small images below a viewpoint near the original camera positions was selected.

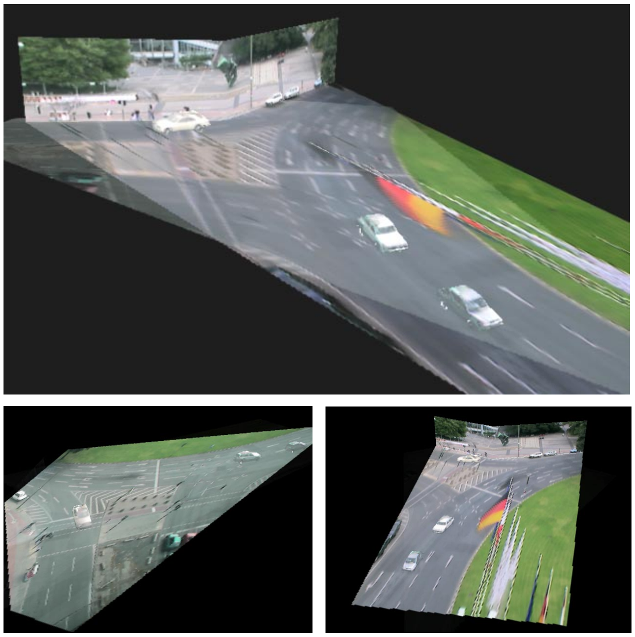

Fig. 7: Final scene views with reconstructed dynamic objects from different directions

Besides the obtained results, there are some drawbacks on the current procedure. First, a traffic scene needs to be fully calibrated and exact homographies have to be available. Otherwise, ground planes do not fit correctly, as mentioned in Section 3. One mayor problem results from incorrect segmentation, especially from shadows that are segmented as foreground objects and therefore disturb the visual impression of the rendered textures. Although even strong changes in lighting conditions have no influence on background processing, however, they influence the segmentation process by changing shadow appearance. Currently, the segmentation routine cannot separate vehicles moving close together at the same velocity until the Kalman filter tracking approach splits these objects once an object is correctly identified.

5. CONCLUSIONS

A system for 3D traffic scene reconstruction from a small number of camera views has been presented. These views typically have large viewing angles between each other and show little overlap in order to cover large areas. Therefore calibration information is exploited to provide an adequate 3D model of the scene. The reconstruction process starts with segmentation to separate dynamic foreground objects from static background information. This approach is well suited for scenarios such as this specific traffic surveillance setup, where static cameras and the scene provide a stationary background. In the reconstruction process, the final background model is composed of a multi-texture ground plane and additional side planes. The ground plane textures are weighted automatically by the renderer according to the point and direction of view. Foreground objects are modeled by mapping the appropriate original textures from all views onto a synthetic best-match model. The position is calculated from the 3D object motion trajectory and updated each render cycle to keep the scene model up to date. This interpolation also supports render cycle times that are faster than the original input camera frame rate.

6. REFERENCES

[1] Remagnino, P, *et al.* “An Integrated Traffic and Pedestrian Model-Based Vision System”. *Proc. Of British Machine Vision Conference 1997*, vol. 2 pp.380-389, 1997

[2] D. Koller, K. Daniilidis, H.-H. Nagel: ”Model-Based Object Tracking in Monocular Image Sequences of Road Traffic Scenes“, *International Journal of Computer Vision*, vol. 10, no. 3, pp. 257-281, 1993

[3] R. Collins *et al.* "A System for Video Surveillance and Monitoring”, *VSAM Final Report," Technical report CMU-RI-TR-00-12*, Robotics Institute, Carnegie Mellon University, May, 2000.

[4] S. M. Seitz, and C. R. Dyer, “Photorealistic Scene Reconstruction by Voxel Coloring”, *International Journal of Computer Vision*, 35(2), 1999, pp. 151-173.

[5] M. Levoy, and P. Hanrahan, “Light Field Rendering”, *Proc. ACM SIGGRAPH*, pp. 31-42, August 1996.

[6] C. Fehn, E. Cooke, O. Schreer, and P. Kauff, “3D Analysis and Image-Based Rendering for Immersive TV Applications”, *Signal Processing: Image Communication Journal, Special Issue on Image Processing Techniques for Virtual Environments and 3D Imaging*, Oct. 2002.

[7] C. Ridder, O. Munkelt, and H. Kirchner, “Adaptive Background Estimation and Foreground Detection using Kalman-Filtering”, *Proceedings of International Conference on recent Advances in Mechatronics ICRAM’95*, pp. 193.195, 1995

[8] P.H.S. Torr, and D.W. Murray. “The Development and Comparison of Robust Methods for Estimating the Fundamental Matrix”, *International Journal of Computer Vision*, vol. 24 no. 3, pp. 271-300, September 1997.

[9] D. Koller, J. Weber, and J. Malik, “Robust Multiple Car Tracking with Occlusion Reasoning”, *Technical Report UCB/CSD 93/780*, University of California at Berkeley, pp. 189-196, 1994