A SPACE- AND TIME-EFFICIENT MOSAIC-BASED ICONIC MEMORY FOR INTERACTIVE SYSTEMS

Birgit Möller and Stefan Posch
Institute of Computer Science
Martin-Luther-University Halle-Wittenberg
Von-Seckendorff-Platz 1, 06099 Halle (Saale) / Germany
Email: {moeller, posch}@informatik.uni-halle.de

Keywords: visual memory, mosaic images, online processing, memory- and time-efficiency, polytopes, multi-resolution.

Abstract: One basic capability of interactive and mobile systems to cope with unknown situations and environments is active, sequence-based visual scene analysis. Image sequences provide static as well as dynamic and also 2D as well as 3D information about a certain scene. However, at the same time they require efficient mechanisms to handle their large data volumes. In this paper we introduce a new concept of a visual scene memory for interactive mobile systems that supports these systems with a space- and time-efficient data structure for representing iconic information. The memory is based on a new kind of mosaic images called multi-mosaics and allows to efficiently store and process sequences of stationary rotating and zooming cameras. Its main key features are polytopial reference coordinate frames and an online data processing strategy. The polytopes provide euclidean coordinates and thus allow the application of standard image analysis algorithms directly to the data yielding easy access and analysis, while online data processing preserves system interactivity. Additionally, mechanisms are included to properly handle multi-resolution data and to deal with dynamic scenes. The concept has been implemented in terms of an integrated system that can easily be included as an additional module in the architecture of interactive and mobile systems. As one prototypical example for possible fields of application the integration of the memory into the architecture of an interactive multi-modal robot is discussed emphasizing the practical relevancy of the new concept.

1 INTRODUCTION

To build efficient interactive computer systems able to cope with unknown situations and environments is a present-day research topic. Using active cameras enables task-directed data acquisition and provides increased flexibility to such interactive systems. However, since active visual scene analysis also implies the processing and storage of complete image sequences rather than of single frames, also efficient algorithms for data extraction and analysis as well as appropriate data structures for compact representation of the sequences are necessary ingredients.

Mosaicing algorithms are one possible approach for representing image sequences in a compact and space-efficient way. The basic idea is to fuse all images of a sequence into one single frame. The resulting mosaic image covers the field of vision of the complete sequence and thus may be viewed as extending the camera's field of vision both in space and time. In this sense, it supplies interactive systems with a *visual scene memory*. Processing of image data may be done online during acquisition, but also at anytime later. Thus, e.g., a detailed scene analysis might be performed some time after image acquisition using only the visual scene memory without need for a time consuming physical rescan of the scene nor the necessity to store all images ever acquired by the system.

During the last years a large amount of mosaicing approaches has been published. Most of these works are directed towards the generation of high quality mosaic images as needed, e.g., in computer graphics applications. However, adopting the algorithms published for use with interactive and mobile systems bears several problems since the general conditions are pretty much different putting special demands on mosaicing algorithms. Especially in mobile systems most of the time only limited storage and computational resources are available. Thus it is, e.g., not possible to process whole sequences simultaneously, as it is often done in existing approaches (e.g., (Davis, 1998; Sawhney et al., 1998)). Besides, such a strategy would severely impede a system's interactivity. On the one hand a mosaic image is only available after all images have been processed, and on the other hand the system's ability to immediately address user interactions is limited. Hence, within this field of application mosaicing algorithms working in online mode and providing data access all the time are mandatory, preventing many existing approaches from easily being transferred to this field of application.

Regarding interactive systems not only online data processing and anytime access but also easy access with respect to the structures and representations of the data are of significant importance. Especially the direct applicability of existing image analysis techniques to the mosaic data enables interactive systems to efficiently exploit the visual data without need for any algorithmic adaptations. Consequently, since the majority of existing image processing algorithms assumes an euclidean reference frame such a frame is prerequisite for mosaic images to be applicable in this context and to gain broad acceptance.

In this paper we present a new concept for a mosaic-based visual scene memory that meets the abovementioned requirements for use with interactive mobile systems. It is currently capable of representing image sequences of rotating and zooming, but stationary cameras. Although mobile systems are non-stationary this is not a severe restriction at all since visual data of a scene can usually adequately be represented based on a set of mosaic 'screenshots' taken from different positions within the scene. The most important features of the memory are its support for an incremental online-update of the data and the direct applicability of standard image analysis algorithms due to its euclidean coordinate frame (see next paragraph). It allows a large field of vision and implements efficiently varying resolutions of image data as required for zooming active cameras. This paper focusses on the representation of static scene data within the memory. Nevertheless also mechanimsms to additionally include dynamic as well as more abstract representations of image data are integrated, for details refer to (Möller and Posch, 2002).

One important decision in generating mosaic images is to choose a suitable reference coordinate system for projecting the data. As one of our primary goals is to support the application of conventional image analysis techniques, our memory requires a euclidean reference frame. Spheres or cylinders are usually chosen to represent large field of vision of stationary cameras but do not deliver such an euclidean reference frame. Further on, they are difficult to represent (Bishop and McMillan, 1995) and data access as well as registration and integration of new data in an online fashion using such coordinate frames often requires explicit view rendering (e.g., (Shum and Szeliski, 2000)). Thus, an uniform representation and handling of the mosaic images in registration and integration as well as for data access is favorable. Due to these requirements our approach is based on *polytopes* approximating a sphere. Projecting the data onto tiles of a polytope enclosing the camera center allows to represent the entire field of vision of the cameras while at the same time image distortions are reduced and euclidean coordinates are granted.

Image sequences are typically captured with varying zoom. Using a single mosaic image with fixed resolution is as a consequence not adequate for these sequences. Hence, our memory is hierarchically organized and nests differently scaled incarnations of a polytope. The resulting data structure consists of different image planes arranged as polytopes and is capable of representing different levels of resolution. We call this enhanced mosaic a *multi-mosaic*.

The remainder of this paper is organized as follows. Section 2 outlines the basic mosaicing algorithms while in section 3 our new *multi-mosaic* concept is introduced. Additionally implementatory details can be found there. Section 4 presents some results before in section 5 an exemplary application in the field of human-machine interaction is considered. The paper finishes with a conclusion in section 6.

2 MOSAIC IMAGE BASICS

Mosaic images are usually generated following a two-step strategy. First, for each image of a sequence parameters of a suitable motion model are estimated to compensate for the camera motion *(registration)*. Afterwards all images are warped towards a common coordinate frame and *integrated* by fusing their color information. Both steps can either be performed in an offline or an online fashion.

In the first case, all images of a sequence are processed simultaneously in registration and integration. This yields a globally optimal mosaic representation which is only accessible after *all* images have been integrated. In interactive and mobile systems image data becomes available incrementally and hence it is straightforward to process the data by following a continuous online strategy. This is accomplished by registering and integrating each image separately into the evolving mosaic image. With each new image the mosaic is updated and a complete data representation can be provided after each registration step. However, it should be noted that the resulting representation is only locally optimal since image registration and integration can only rely on the current frame and the mosaic itself. Hence, inconsistent parameters and integration errors due to error accumulation cannot be completely omitted. Nevertheless, even in long-term mosaicing as aimed by our approach image quality is still sufficient to enable further data analysis.

Registration. The parameter estimation is based on a suitable model for the camera motion. This model

Figure 1: Example mosaic demonstrating distortions and uncontrollable image growth during mosaicing an image sequence of a rotating camera. The data is projected onto a single image plane yielding an inadequate coordinate frame.

mainly depends on the degrees of freedem of the camera and is often chosen to be a pure translation or an affine coordinate mapping. In our application we use stationary rotating and zooming cameras. Their motion can best be modeled by homographies (Hartley and Zisserman, 2000) whose parameters are estimated using *projective flow* (Mann and Picard, 1996). The main idea is given to calculate the optical flow between two images constraint by the projective motion model. However, due to the non-linearities within the homographies the algorithm operates in an iterative framework based on piecewise linear homography approximations. Further on a resolution hierarchy (Bergen et al., 1992) is used to cope with large offsets. To reduce the influence of accumulating errors in online image registration as outlined in the previous section our mosaics are generated in *frame-to-mosaic mode* (Burt and Anandan, 1994). Parameters for each image are estimated with regard to a suitable clip reprojected from the mosaic image generated so far. Thus all images formerly integrated at least implicitly influence the current estimation and parameter quality is improved without processing the whole sequence simultaneously.

Integration. During image integration the color information of all sequence images is merged to give single mosaic pixel values. This can be accomplished by fusing the values of all image pixels that are projected onto a mosaic pixel, e.g., calculating an average or median. However, in long-term mosaicing continuously averaging pixel values causes image blurring which is primarily due to small registration errors unavoidable in online mosaic generation. Thus, an integration method needs to be applied that provides high-quality images even for a long period of time.

In the literature several quite sophisticated methods for (offline) mosaic image quality enhancement are to be found (e.g., (Capel, 2004)). However, for efficiency reasons and due to the fact that online integration is required we rely on a more simple but equally appropriate strategy. One single image is selected as source for each mosaic pixel so that averaging pixel values is omitted. The source images are selected based on the time stamps of the input images. Each mosaic pixel is assigned the pixel value from that input image providing the most recent data. Thus, new information is directly integrated whenever it becomes available yielding a continuous data update. This strategy results in a segmentation of the mosaic image into regions with each region originating from a different sequence image. Due to changes in the lighting conditions or camera exposure settings visible seams in the mosaic image might appear at the boundaries between different regions. They are eliminated by applying linear or sigmoid blending functions along region boundaries.

3 MULTI-MOSAICS

Mosaic images share a wide variety of applications ranging from image-based rendering in computer graphics and virtual reality to computer vision applications and visual scene analysis. Depending on camera motion and intended area of application, which in our case is given by mobile interactive systems, a suitable reference coordinate system has to be chosen for the mosaic images as already mentioned. In most work it is defined as a single image plane, e.g., (Mann and Picard, 1996). However, in case of a stationary camera performing large rotations projecting all images onto a single plane usually results in undesirable large distortions (fig. 1). They cause an excessive growth of the mosaic image area and consequently enforce extensive use of pixel interpolations which results in low image quality. This in turn hampers registration and integration and renders image analysis nearly impossible. Using a cylinder (Bishop and

McMillan, 1995) or sphere (Coorg and Teller, 2000) as coordinate frame avoids distortions, but dealing with spherical coordinates in image registration, integration and especially in further analysis steps is bulky and often yields undesirable incompatibilities to existing software modules.

3.1 Polytopial Coordinate Frames

Both requirements of representing the complete field of vision of rotating cameras as well as providing euclidean coordinates for image processing can ideally be met employing *polytopes* to define the reference coordinate frame in registration and integration as well as in representation of the mosaic images itself. Such frames are up to now primarily used with *offline* rendering applications (e.g.,(Shum and Szeliski, 2000)), however, they also offer a great flexibility for *online* scene modelling and representation tasks as discussed in this paper.

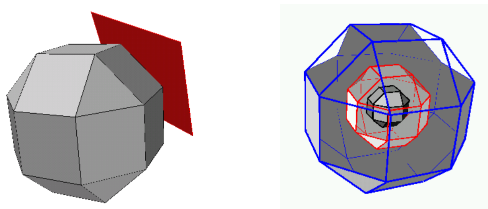


Figure 2: Left, polytopial coordinate frame with FIP attached, and right, hierarchical structure of visual scene memory based on nested polytope incarnations.

The polytopes are centered at the optical center of the camera with their tiles regularly arranged around tangentially to the sphere (fig. 2, left). The origin of the 3D coordinate frame for the polytope is located at the center of the camera and for convenience its z axis is arranged parallel to the optical axis of the camera when acquiring the first image of the sequence. Each tile of the polytope owns a local orthogonal 2D coordinate system. These coordinate frames as well as projective transformations between neighboring tiles are computed offline during an initialization step. All transformations and the neighborhood relations are represented in a graph data structure (fig. 3) providing easy data access and saving time in online generation and update.

The scaling of the polytope is initially chosen according to the focal length of the camera given the assumption that image pixels and pixels on the polytope share the same scaling and aspect ratio. The focal length is currently extracted facilitating an offline calibration strategy based on a 3D calibration pattern (Hartley and Zisserman, 2000) and using a functional mapping between hardware parameters and corresponding focal length in online mode. In principal self-calibration techniques (de Agapito et al., 1999)

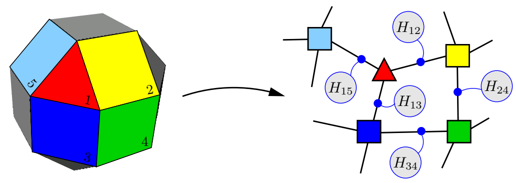


Figure 3: Implicit representation of polytope geometry: the graph data structure stores neighborhood relations in terms of its edges as well as homographies valid between various tiles in terms of corresponding edge lables.

might also be applied, but they have proven to be too unstable in long-term mosaicing so far.

3.2 Online Registration and Integration

One problem during online generation of the multi-mosaic representation results from discontinuities between neighboring tiles of the polytope. Obviously the number of these discontinuities grows with the number of tiles the polytope owns. On the other hand this number should be large for a good approximation to the sphere and a reduction of distortions. We typically use polytopes with about 20 to 30 tiles (e.g. rhombicuboctahedrons, see fig. 2). Nevertheless, efficient handling of the memory data structure requires an elaborate approach for dealing with these discontinuities, as presented below.

Integrating a new image into the multi-mosaic representation requires registration of the image and projecting its data onto related tiles. For registration a suitable reference image is needed which in frame-to-mosaic mode is defined as a clip of the current mosaic representation. This clip might be chosen as part of the one single tile that best approximates the orientation of the new image. However, in worst case an overlap of slightly more than 50% between the clip and the image to be registered will result. Most of the time this is not sufficient to guarantee robust parameter estimation (besides disregarding available information in any case). As an alternative, we can construct a larger reference frame by clipping not only from one tile, but also from neighboring tiles. However, constructing such a composed reference image and back projecting the integrated new data to the polytope is a time-consuming procedure. In a naive implementation this would have to be repeated for each frame of the image sequence.

Focus Image Plane. We solve this problem by facilitating an additional image plane, the so called *focus image plane (FIP)*. It serves as some kind of "cache" storing recently acquired image data and granting easy access to it. The FIP is attached to the polytope (fig. 2, left) and used as reference in registration and integration, thus, masking the underlying topological structure of the multi-mosaic representation. New image data is registered according to the data on the FIP and also integrated into it following the strategy mentioned in section 2. Integration into the polytope itself is accomplished only when the position of the FIP needs to be updated. This is the case if parts of the integration area of a new image do not intersect with the domain of the focus image plane any longer due to significant discrepancies between the orientation of the current image plane and the one of the FIP. As the size of the FIP is usually chosen two to three times larger than that of the input images this occurs only after several images have been integrated depending on speed and/or size of rotation angles of the camera.

Focus Image Plane Update. The shortest point distance of the area of integration of the current image on the FIP to its boundaries is considered to monitor for necessary updates of the position of the FIP. If the distance falls below a certain threshold the image data of the FIP is projected onto related polytope tiles. These are detected by projecting the rectangular bounding box of available image data onto the polytope and calculating intersections with domains of single tiles. A pointer to the tile meeting the orientation of the FIP best is always kept in memory for efficiency reasons and used as starting point for copying the data. If parts of the FIP data project outside of the domain of a tile, the data update is recursively continued on neighboring tiles until the complete image data has been integrated into some tile of the polytope. In this procedure only tiles are checked if their orientations differ by not more than 80 degrees from the one of the FIP.

Orientation and position of the FIP are updated after copying the data. The new parameters are chosen according to the position and focal length of the current input frame. Additionally the history of the motion path of the camera during the last frames is taken into account. It is quadratically extrapolated based on the assumption of smooth camera motion and thus helps to minimize the overall number of FIP updates necessary. Finally, new reference data is projected to the new FIP. To efficiently identify the tiles providing image data to the new FIP, the same strategy as for copying data onto the polytope is applied.

3.3 Multi-Resolution Data

Representing image sequences that contain different levels of resolution puts specials demands on a mosaic data structure. Usually only a single resolution can adequately be represented within one mosaic image. Integrating image data with higher resolution forces to downsample these data causing a loss of information. Contrary, inserting low-resolution data into a mosaic with higher resolution requires interpolating the data and thus enlarges the data volume without gaining more information.

In our memory several differently scaled instances of the polytope are nested into each other (fig. 2, right) covering a discrete set of resolutions. Depending on the current focal length of the camera the polytope entity is chosen for data integration that best meets the focal length of the current input image. The granularity of the resolution hierarchy can freely be configured by the user and is, thus, highly flexible. In particular, each distance between adjacent levels can individually be defined according to the required level of representation detail in certain resolutions.

In contrast to standard and commonly used multiscale representations like, e.g., gaussian resolution pyramids or wavelets, the structure proposed provides direct data access in all resolutions without need for intermediate image reconstruction. Further on representation of image data might be restricted to only some few levels of resolution and need not to be carried out for all available resolutions. On each level only data is represented that was actually provided by the camera. Consequently, the memory allows simultaneous representation of image data of a single scene part that might have been acquired at completely different points in time and with varying zoom.

3.4 Sparse Memory Representation

Integrating an image sequence into a mosaic image significantly reduces the amount of iconic data to be represented in the multi-mosaic. However, covering the whole potential field of vision at all resolutions still requires a lot of memory space. This can cause performance problems especially in mobile systems. Due to the fact that most of the time not all parts of a scene are actually scanned and explored in all resolutions anyway, the space needed can be reduced facilitating a *sparse memory representation*: only tiles of a multi-mosaic are instantiated that actually contain image data. Hence, memory for a single tile is allocated only after the camera has scanned the corresponding regions of the scene and data needs to be stored.

Although restricting the representation to tiles that actually contain data allows a reduction of the memory space needed, in levels of high-resolution an additional segmentation of tiles into subcells is implied. In these levels the size of a single tile usually exceeds the size of acquired images for several times and only some few regions of interest need to be represented. Thus, the single tiles are further segmented into subcells. Their actual number is chosen individually for each tile and is derived according to the relation of the input image size to the tile size so that the number of subcells that have to be checked on integrating

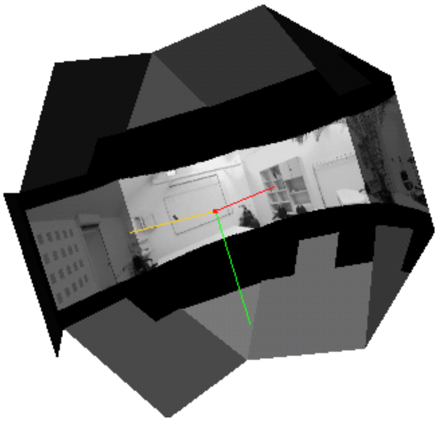


Figure 4: Multi-mosaic resulting from a scene scan covering approximately 180° in horizontal direction, rendered using Open Inventor^{TM}. Black regions indicate areas where waste memory was allocated, and gray regions were added to better illustrate the underlying structure of the complete polytope representation.

new image data is kept small. Subcells where memory has to be allocated are determined by projecting rectangular bounding boxes of available image data onto the tiles. This leads to few regions on the polytopes where memory is allocated without image data present (black regions in the example images). Nevertheless, we prefer this scheme to a polygonal approximation of valid image regions since this complicates memory handling and is less time efficient.

4 IMPLEMENTATION & RESULTS

The concept of the multi-mosaics presented in this paper has been implemented in terms of an integrated system that might be included as an additional module in the architecture of interactive systems. Within this paragraph two exemplary memory representations are discussed in detail while in section 5 a prototypical integration of the new concept into the architecture of a mobile robot is presented.

Figure 4 shows a multi-mosaic representation based on image data including only one single level of resolution. The scene represented is the same as in figure 1, but this time the image data is projected onto an adequately scaled rhombicuboctahedron. Large distortions are no longer present in the mosaic and the image quality is sufficient to allow for image analysis algorithms to be applied directly to the mosaic. It should be mentioned that registration and integration errors cannot be avoided completely over time. Especially in long-term mosaicing small errors accumulate and could only be eliminated by registering all images simultaneously. This is not feasible for an online processing strategy, but errors are reduced by applying the frame-to-mosaic mode as described. Further on the integration heuristic copying new data regionwise to the mosaic minimizes the effects of small errors, but causes some blurring at region boundaries within the mosaic. Since most image analysis algorithms apply smoothing to the data anyway, the mosaic provides sufficient quality for image processing.

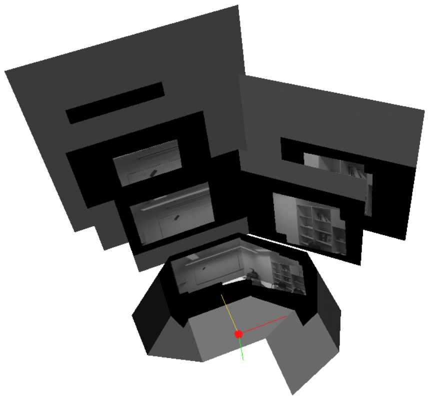


Figure 5: Hierarchical representation of a scene explored. Two areas of the scene (whiteboard and shelves) have been scanned in detail as apparent in the levels of higher resolution of the multi-mosaic image.

An example for the multi-resolution visual memory is shown in fig. 5. The camera first scanned the whole scene in coarse resolution to get an overview. Subsequently the whiteboard and shelves were manually selected for detailed exploration. The corresponding representations for different resolution levels of the shelf are magnified in fig. 6. Although the books in the shelf are visible even in coarse resolution, their title and other details are only accessible by zooming in. The same holds, e.g., for text written on the whiteboard. Contrarily, neither the whiteboard itself nor other parts of the shelf need to be represented in larger detail. The memory, thus, yields a compact representation of the scene requiring only minimal memory resources to store important scene data.

5 MULTI-MOSAICS IN PRACTICE

The multi-mosaic data structure yields an efficient iconic representation of image sequences as acquired by interactive and mobile systems. Due to the online data processing and the euclidean coordinate frame the memorized data provides a suitable base for more efficient image sequence analysis in interactive and particularly in mobile systems. Such systems often explore new environments by first collecting data and

Figure 6: Two example images of low- (left) and high-resolution data (right) extracted from the multi-mosaic in fig. 5. Details are only visible in high resolution, however, there are usually only few sections of a scene where such details need to be represented for scene analysis anyway.

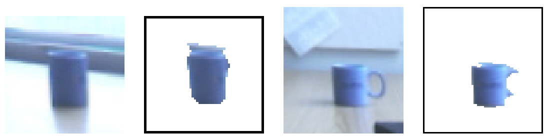
then building some kind of map to be subsequently used in navigation. However, maps are most of the time not based on visual input and vision-based approaches have only been used in a few publications (e.g. (Ishiguro and Tsuji, 1996)). This is mainly due to the fact that pure navigation and localization tasks can often be better solved relying on more robust range data. However, if a mobile robot is supposed to perform some tasks of scene analysis as well, visual information is indispensable. Moreover it is essential that the visual data is adequately represented and easily accessible, ruling out rather indirect representations like, e.g., wavelets or mipmaps. One large field of application for such kind of representations is intuitive human-machine-interaction where visual data is one of the most important sources of information.

As a prototypical example for such an application the visual memory has been integrated into the architecture of a mobile robot (Möller et al., 2005). The robot is supposed to support a human user in his everyday life at home by performing tasks like, e.g., searching and fetching objects (*home tour scenario*). Due to the fact that tasks the robot has to perform are highly user-dependent and may exhibit a large variance one key capability of the robot is autonomous learning of new tasks by instruction. Therefore the robot provides the human user with intuitive multi-modal communication facilities like speech and gesture recognition as well as processing of visual information for efficiently analyzing and understanding human instructions. Besides, another important ingredient in multi-modal learning is object recognition and learning since nearly all tasks more or less deal with objects. The robot solves this task following an appearance-based object recognition strategy.

In appearance-based object recognition objects are recognized by matching current images of the object against formerly acquired views of different objects as stored in the scene model, e.g., using appropriate measures of image correlation. The more of such views are available, the better in principal the final recognition results will be. However, acquiring different views of an object is sometimes quite time-consuming and sophisticated for a mobile robot. In worst case it has to manoeuver around the object completely to get these views which on the one hand might be difficult, e.g., due to obstacles on the ground, and on the other hand limits the robot's interactivity during this period of time boring the human user.

To come up with these problems of view acquisition we adopt the multi-mosaic images as some kind of *visual memory* in the robot's architecture. This idea originates from the observation that the robot sometimes just idles around while waiting for a communication partner. During this time the robot already gathers visual information about the scene and especially about objects included. However, at the time of acquisition this visual information is irrelevant to the robot and, hence, is usually discarded raising the need for later on rescanning the scene when the data is actually needed. This can be avoided by having the robot building up an iconic mosaic-based visual map of the environment in idle periods of time. In this way all visual data ever acquired is kept in a compact and space-efficient way allowing the robot to refer to it afterwards in concrete communication and learning situations. At the same time expensive hardware-based re-explorations of a scene are avoided. Figures 7 and 8 show an example for such a representation, extracted object views of an object to learn and, finally, results from a post-processing step.

Figure 7: An exemplary multi-mosaic image from the scene memory as generated from a 60^o camera pan.



(a) (b) (c) (d)

Figure 8: (a), (c) Subimages as extracted from various multi-mosaics during object learning; (b), (d) exemplary results of a subsequent color segmentation step.

The memory representation itself consists of a set of various multi-mosaic images each acquired at a specific position in the world. Since the concept of the multi-mosaics enforces the robot to remain at a single position during image acquisition it is not possible to gather visual information while moving. Nevertheless, this is not a serious drawback. Most of the time a scene can even better be represented by scanning it from some key positions within the scene than by monitoring all the different pathways the robot pursuits. Besides, building mosaics while the robot is moving bears different problems. On the one hand no closed-form solution for modelling the camera motion is available since homographies do not hold in such situations, and on the other hand appropriate mosaic images, like, e.g., manifold mosaics (Peleg et al., 2000), often exhibit (perspective) distortions of the scene data hampering easy analysis and scene understanding. The multi-mosaics are currently primarily used to extract additional views of an object in object learning and recognition. However, they could also be used for extracting 3D data of the scene provided that the mosaic 3D world positions are given as, e.g., proposed in (Teller, 1998).

6 SUMMARY AND CONCLUSIONS

Active scene analysis and exploration gains increasing importance in computer vision. Since analyzing image sequences of active cameras has proven a suitable base for extracting useful information from a scene, interactive and mobile systems are nowadays often equipped with active sensing devices. The visual scene memory based on multi-mosaics presented in this paper perfectly fits into this framework as an additional module between active data acquisition on the one hand and its analysis on the other. The memory is based on a polytopial reference coordinate system. In contrast to spheres and cylinders the polytopes provide an euclidean reference frame and, hence, allow the direct application of standard image analysis techniques. This is important for interactive systems since they can work on the memorized data as on the originally acquired input images. Further on, based on this euclidean mosaic representation and the chosen data processing strategy, the data within the memory is easily updated in an online fashion. Incremental parameter estimation and integration heuristics are used in combination with the focus image plane. The latter masks the underlying polytope structure of the memory and thus allows efficient data access despite present discontinuities on the memory itself. Given these techniques the memory works quite stable in practice, nevertheless, future work has to be carried out on investigating more robust online parameter estimation techniques and mechanisms for automatically detecting registration errors.

The memory is ideally suited to be used with interactive and mobile systems that have to store and afterwards access image sequences. Especially systems in human-machine interaction significantly benefit from the memory as it provides an improved and more efficient exploitation of available visual data and yields a higher flexibility as it is necessary to act in dynamically changing environments as well as to perform intuitive communication with human beings.

REFERENCES

Bergen, J., Anandan, P., Hanna, K., and R.Hingorani (1992). Hierarchical model-based motion estimation. In *ECCV*, pages 237–252.

Bishop, G. and McMillan, L. (1995). Plenoptic modeling: An image-based rendering system. In *SIGGRAPH Computer Graphics Proceedings*, pages 39–46. Annual Conference Series.

Burt, P. and Anandan, P. (1994). Image stabilization by registration to a reference mosaic. In *Image Understanding Workshop*, pages 1:425–434, Monterey, CA.

Capel, D. (2004). *Image Mosaicing and Super-resolution*. Springer.

Coorg, S. and Teller, S. (2000). Spherical mosaics with quaternions and dense correlation. *International Journal of Computer Vision*, 37(3):259–273.

Davis, J. (1998). Mosaics of scenes with moving objects. In *CVPR*, pages (1):97–100, Santa Barbara, USA.

de Agapito, L., Hartley, R., and Hayman, E. (1999). Linear self-calibration of a rotating and zooming camera. In *IEEE Int. Conference on Computer Vision and Pattern Recognition*, pages 15–21.

Hartley, R. and Zisserman, A. (2000). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

Ishiguro, H. and Tsuji, S. (1996). Image-based memory of environment. In *Proc. of Int. Conf. on Intelligent Robots and Systems (IROS '96)*, pages 634–639.

Mann, S. and Picard, R. (1996). Video orbits of the projective group: A new perspective on image mosaicing. Technical Report 338, MIT Media Laboratory Perceptual Computing Section.

Möller, B. and Posch, S. (2002). Analysis of object interactions in dynamic scenes. In *Pattern Recognition, Proc. of DAGM Symp.*, LNCS 2449, pages 361–369, Zurich, Swiss. Springer.

Möller, B., Posch, S., Haasch, A., Fritsch, J., and Sagerer, G. (2005). Interactive object learning for robot companions using mosaic images. In *Proc. of Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 371–376, Edmonton, Canada.

Peleg, S., Rousso, B., Rav-Acha, A., and Zomet, A. (2000). Mosaicing on adaptive manifolds. *PAMI*, 22(10):1144–1154.

Sawhney, H. S., Hsu, S., and Kumar, R. (1998). Robust video mosaicing through topology inference and local to global alignment. In *ECCV*, pages 103–119, Freiburg.

Shum, H.-Y. and Szeliski, R. (2000). Systems and experiment paper: Construction of panoramic image mosaics with global and local alignment. *IJCV*, 36(2):101–130.

Teller, S. (1998). Toward urban model acquisition from geolocated images. In *Proc. of Pacific Graphics*, pages 45–51, Singapore.