How Useful is Structure in Motion? *

Walter G. Kropatsch
Vienna University of Technology
Institute of Computer Aided Automation 183/2
Pattern Recognition and Image Processing Group
Treitlstr. 3, A-1040 WIEN, Austria
Tel. +43-1-58801-18350
krw@prip.tuwien.ac.at

Abstract

Structural properties of objects are robustly mapped into images due to their insensitivity to minor geometrical distortions. This becomes particularly important if the imaging/scene geometry changes from frame to frame in a motion sequence. Based on our experience with hierarchically organized graph representations, several issues of motion analysis are addressed from the structural point of view in contrast to the usual geometry based methods like differential motion analysis, optical flow, tracking interest points. The goal of the examples in this contribution is to stimulate discussion on the use of 'structure in motion' analysis as opposed to 'structure from motion'.

1 Introduction

Classical image analysis aims at finding regions in the image that correspond to objects or object parts as well as their mutual relations. Representations (see Fig. 1) that describe the resulting relationships (e.g. adjacent, close, inside) are, among others, the region-adjacency-graph (RAG) [7] or the area Voronoi diagram [2]. Subsequent grouping strategies fit well into the above representation by graphs. Such graphs embedded in the image plane can be computed from pixel neighborhoods by dual graph contraction (DGC) [5].

Visual motion analysis searches the location and shape of 3D objects and their motion trajectories from a dynamic image sequence. Most of the frequently used methods try to estimate (x, y, z, t) coordinates for all points in the image sequence. The methods can be categorized [7] as:

- optical flow computed by spatio-temporal derivatives;

*This work was supported by the Austrian Science Foundation under grant S 7002-MAT.

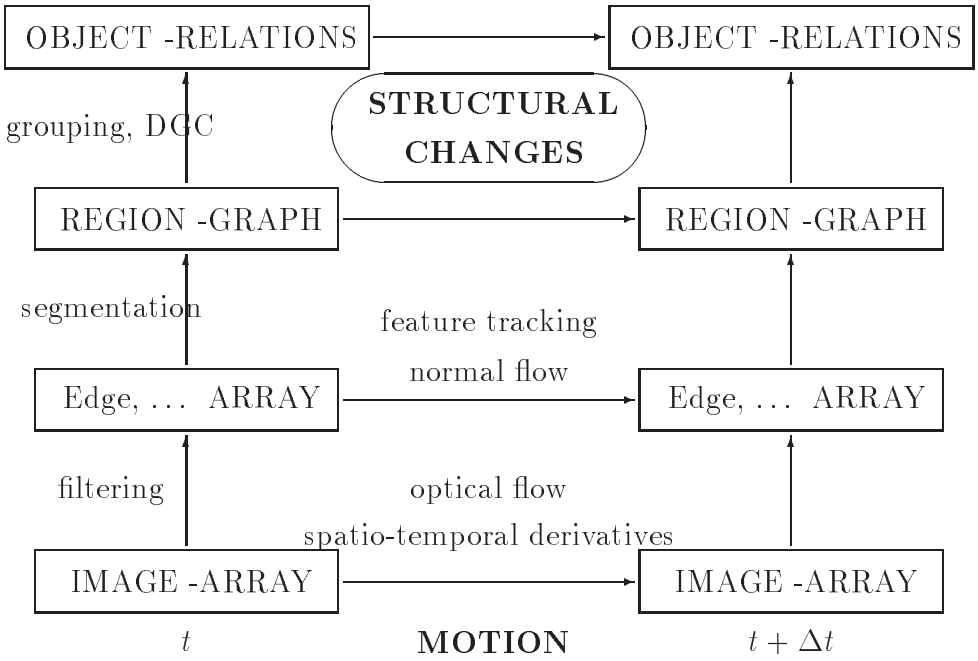


Figure 1: Structure in Image and Motion

- detection of appearance change (e.g. of human faces [1]);
- normal flow [4];
- tracking points of interest.

These approaches face several problems: the resulting motion is based on the given time resolution (e.g. frame rate), high-frequency noise has dramatical consequences: a general lack of robustness; regularization to overcome this situation uses very general smoothness constraints; optical flow is often expressed in image-centered coordinates; sensitivity to unavoidable geometrical inaccuracies and to camera callibration; the high computational complexity; although interpretations of real scenes are highly constrained by physical and functional conditions they cannot be integrated in the above aproaches.

In this paper we would like to stress the question whether the solution to a motion problem really needs highly accurate coordinates as it is the declared goal of most current approaches. Would it not be sufficient to know whether one approaches an obstacle or not? Or to determine whether one moves in the same direction as the people in the direct neighborhood? Such questions can be efficiently answered by structural descriptions of a dynamic scene[1] as the basis for comparison rather than metric spaces.

Two different motion situations can be distinguished:

[1]By *structure* we mean here the spatio-temporal relations between moving and stationary objects (in-front-of, in-between, etc.) and not the objects' (x, y, z, t)-coordinates in space!

- when the observer moves (egomotion problem); or
- when a moving object is seen from a static camera.

In both cases occlusion may occure and cause changes in the arrangement of objects in the observed image. We will start with the idealistic scenario of viewing two lines of trees. Then we let the very simple object move in front of a structured background. In both cases we study the operations that transform the object arrangments from frame to frame in the motion sequence.

2 Moving along an avenue

Let us consider following arrangments of objects in reality: a row of trees (along an avenue); the windows of a building; houses along a street; picture frames in a gallery etc. These arrangements have in common that they have a given structure and that this structure maps always into the same structure in the image as long as the observer "is on the same side of the objects". In the following we study an exemplary case. Imagine you walk along an avenue bounded by two rows of trees. The (*far*) background row (P_f) consists of five trees (A, C, E, G, I). We assume that the individual trees are thin and can be identified, e.g. either by a characteristic feature or by its geometrical configuration (i.e. cross ratio of distances). The order (e.g. from left to right, denoted by $<$) in which the trees appear on line P_f is preserved in the image P_i: $A < C < E < G < I$ (Fig. 2). This order does not change while you walk along the avenue parallel to P_f, only the geometrical distances are affected by your motion. The geometrical similarity between the different views has been modeled by Chakravarty's characteristic views [3], and is effectively used by Peleg [6] in his manifold projection method.

The order is preserved also for the (*close*) forground row P_c of trees $B < D < F < H$. A simultaneous projection of both lines into the image plane P_i merges the two ordered sequences, e.g. $A < B < C < D < E < F < G < H < I$, see Fig. 2.

The merged order changes when you reach position O_2: $A < B < C < D < E = F < G < H < I$, see Fig. 3. In this position tree E is aligned with tree F and changes its place with F when you continue your walk to the right (in the image).

This is apparent when you reach position O_3 (Fig. 4). In addition A and B are aligned and H and G have changed their order: $A = B < C < D < F < E < H < G < I$.

The above considerations can be generalized to any arbitrary trajectory of the observer in a way similar to the *aspect graph* or the *characteristic view* [3]. These approaches describe *all* possible views of a compact (and mostly convex) 3D object, often approximated by polyhedrons. In contrast to this set-up we consider arbitrary arrangements of objects idealized by points in two and, later on, also in three dimensions, and their maps in an image taken from different locations of the observation space. In our method any location (outside of the objects) is considered and not only rays perpendicular to the Gaussian sphere. The observation space can be partitioned into regions where the projected order does not change. To find this partitioning we note that any change in the order has to

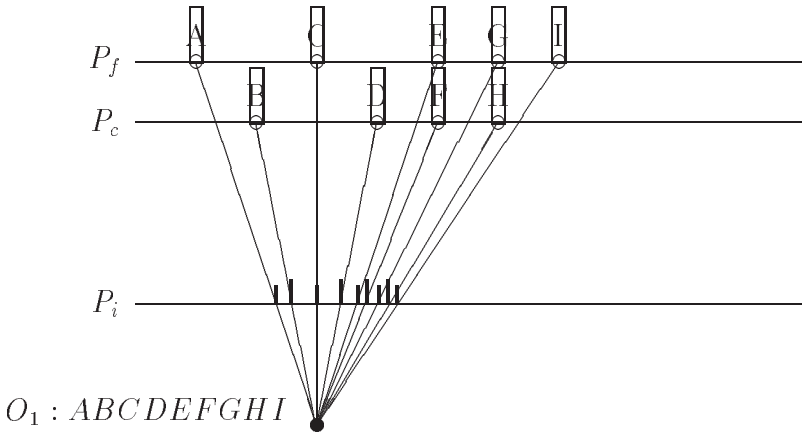


Figure 2: Observer sees 9 trees from first position O_1.

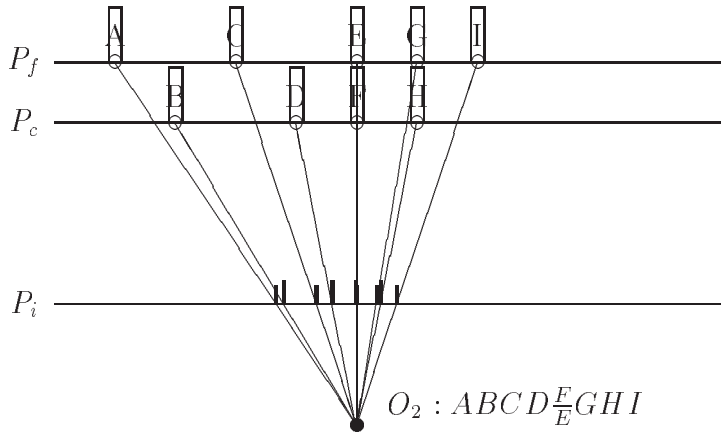


Figure 3: Observer sees 8 trees from second position O_2.

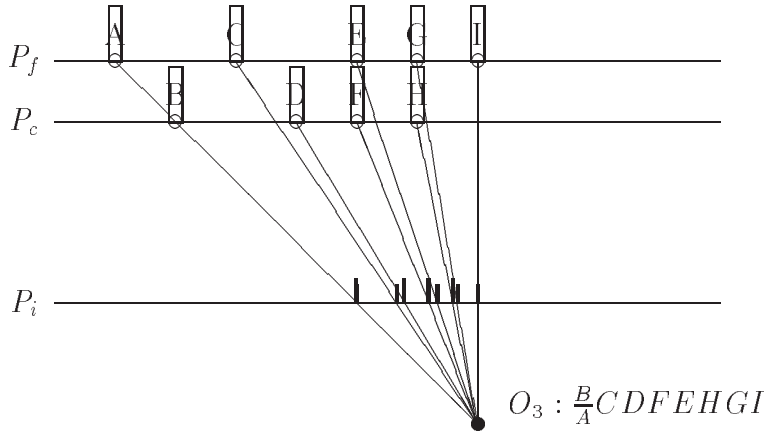


Figure 4: Observer sees 8 trees from third position O_3.

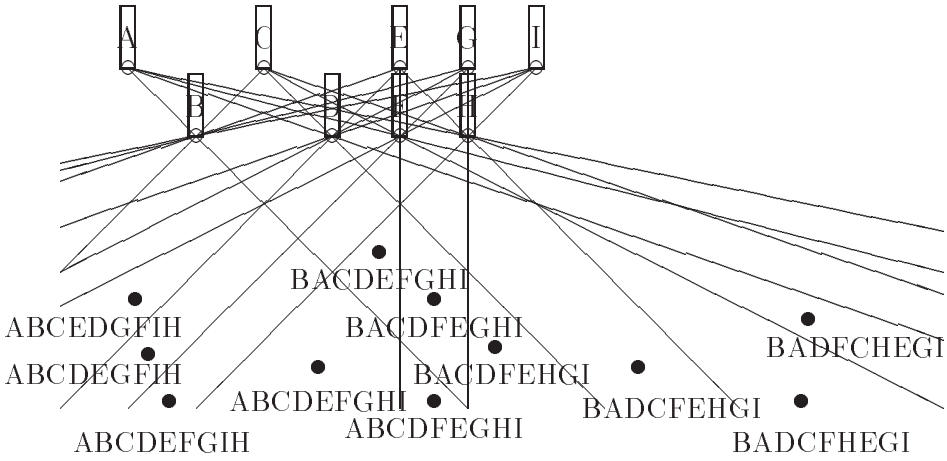
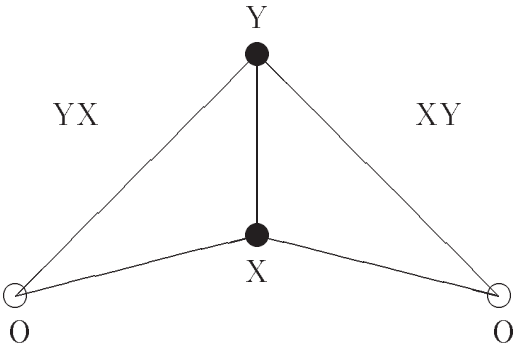


Figure 5: The 9 trees partition the observation space.

occure after two trees are aligned. Traversing such a line corresponds to permuting the order of the two trees in the view. Hence the lines connecting a tree in row P_f with a tree in row P_c identify the partition lines. Fig. 5 shows some partition lines for the above example.

The operation that transforms the order of trees when crossing such a partition line is a **permutation** of the two aligned trees.

2.1 Moving Object vs. Moving Observer



$X < Y \iff$ the triangle OXY is clockwise oriented.

Figure 6: Moving Object vs. Moving Observer

So far we described the situation where the trees are observed by a moving observer. Now consider a stationary observer seeing a row of moving cars in front of a row of parked cars. The situation is very similar. Two objects X, Y seen from observer O appear $X < Y$ as long as the triangle OXY is clockwise oriented (Fig. 6). Hence we conclude:

A moving object in foreground seen from static observer is structurally equivalent to a moving observer.

2.2 Occlusion

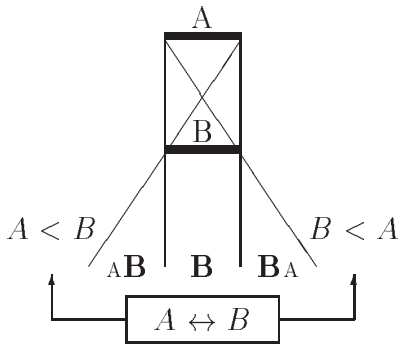


Figure 7: Object width: occlusion.

Our second assumption concerns the width of the trees. By assuming width 0 we avoided problems with occlusion. The situation with nonzero width trees is depicted in Fig. 7. In this case the transition first enters a partial occlusion of the far tree A, then, possibly, it is hidden behind B, before it re-appears on the other side of B. Outside the area of partial or total occlusion the orders are again permutations of A and B.

3 Box crossing a Boundary

In order to study the structural changes that occur if an object moves in front of a stationary background we consider another very simple example (Fig. 8). We study the

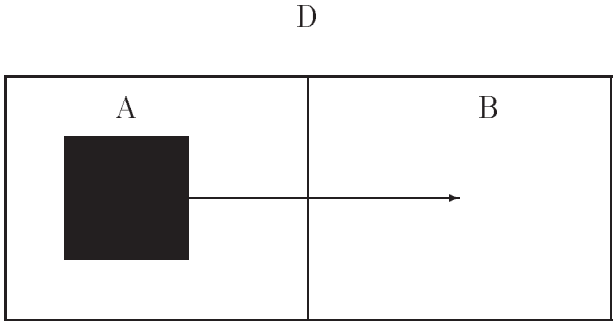


Figure 8: The initial configuration

structural effects of occlusion that the moving object creates on the (simply) structured background. Let our moving object be a small square denoted C and let C move from left

to right. The stationary background consists of two regions, A, B, which are imbedded in another (infinite) region D.

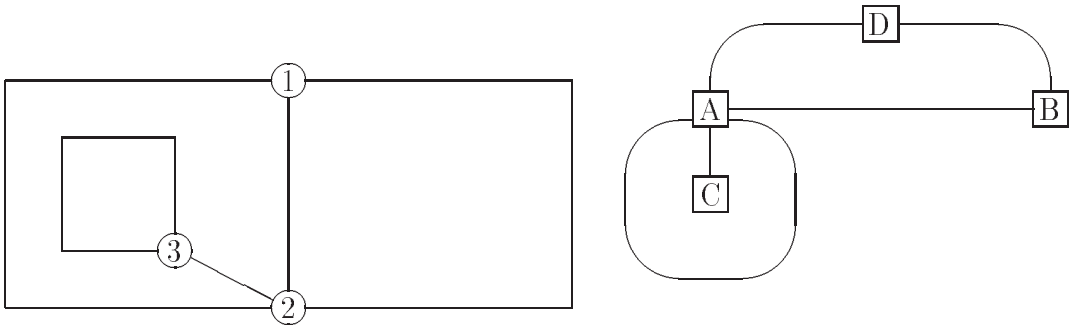


Figure 9: The dual graphs: BCG and RAG

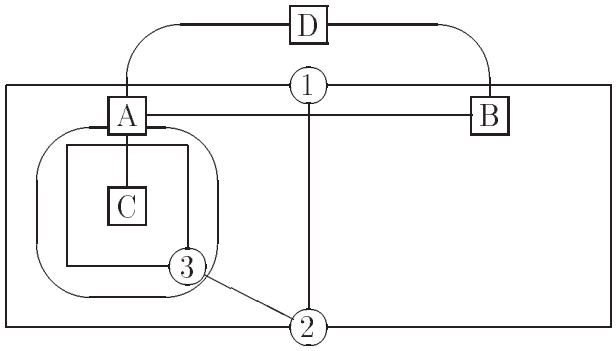
The structure of this configuration is described by a pair of dual graphs, the boundary connection graph (BCG) $G(V, E)$ and the region adjacency graph (RAG) $\overline{G}(F, \overline{E})$ (Fig. 9). Regions (also called *faces*) $F = \{A, B, C, D\}$ having a common boundary are related by an edge in the RAG: $\overline{E} = \{(D, A), (D, B), (A, B), (C, A), (A, A)\}$. The role of the self-loop (A, A) will be explained after studying the BCG.

The vertices of the BCG (e.g. $1, 2, 3$) correspond to points where boundaries meet, the edges represent connected boundary segments. Vertices 1 and 2 are connected by three different boundary segments requiring multi-edges $(1, 2)_1, (1, 2)_2, (1, 2)_3$. Since C is completely contained in A the boundary of C is a closed curve. The placement of the end point (3) of the corresponding edge is therefore arbitrary. Duality between BCG and RAG implies that there is a $1 - 1$ correspondence between the edges, e.g. edges crossing each other in the drawing and appearing on the same line of the table (Fig. 10). The fact that A contains C is expressed in the RAG by the self-loop (A, A) which intersects its counter part edge in BCG, $(2, 3)$. Note also that $(2, 3)$ is not a real boundary since it separates A from A. We maintain such *pseudoedges* to keep the graphs connected.

As long as C moves within A the structure of the RAG does not change until the right boundary of C touches the boundary between A and B. The coincidence of C's right boundary with edge $(1, 2)$ has the following structural consequences:

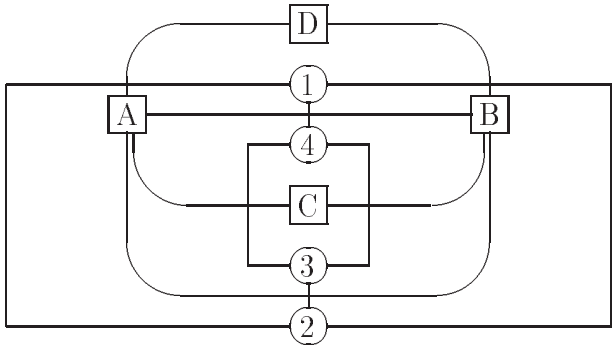
- the right upper corner of C becomes a new vertex 4;
- $(1, 2)$ is split into three parts, $(1, 4), (3, 4)_2$, and $(2, 3)$;
- C becomes a adjacent to B as expressed by edge (C, B);
- the self-loop around C becomes a double edge $(3, 4)$.

In the next stage C occludes a part of the boundary between A and B. Fig. 11 shows the changed configuration. Finally C is completely surrounded by B (Fig. 12). Now C is completely contained in B, the corresponding self-loop (B, B) (re-)appears, pseudo-edge $(2, 3)$ connects the boundary of C with the 'outer' boundary of B, and the three parts of $(1, 2)$ are merged again.



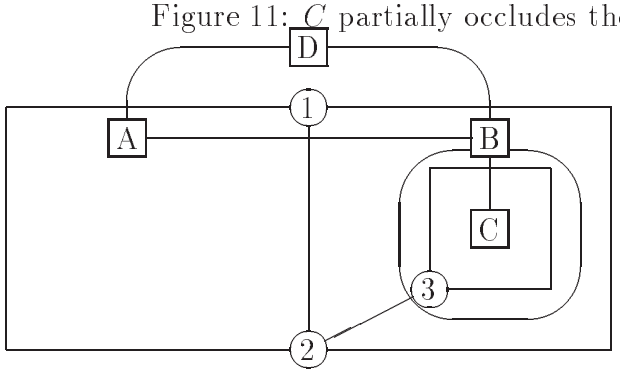
$G_{10} = (V_{10}, E_{10})$		$\overline{G_{10}} = (F_{10}, \overline{E_{10}})$	
V	E	$\overline{E}$	F
1	$(1,2)_1$	(D,A)	A
2	$(1,2)_2$	(D,B)	B
3	$(1,2)_3$	(A,B)	C
	$(2,3)$	(A,A)	D
	$(3,3)$	(A,C)	

Figure 10: Overlaid graphs and formal specification



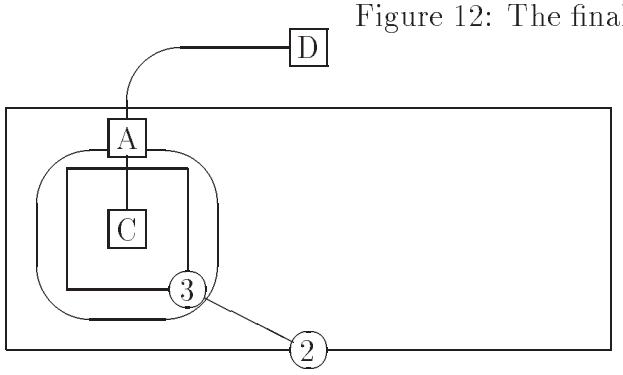
$G_{11} = (V_{11}, E_{11})$		$\overline{G_{11}} = (F_{11}, \overline{E_{11}})$	
V	E	$\overline{E}$	F
1	$(1,2)_1$	(D,A)	A
2	$(1,2)_2$	(D,B)	B
3	$(1,4)$	$(A,B)_1$	C
4	$(2,3)$	$(A,B)_2$	D
	$(3,4)_1$	(A,C)	
	$(3,4)_2$	(C,B)	

Figure 11: C partially occludes the boundary between A and B



$G_{12} = (V_{12}, E_{12})$		$\overline{G_{12}} = (F_{12}, \overline{E_{12}})$	
V	E	$\overline{E}$	F
1	$(1,2)_1$	(D,A)	A
2	$(1,2)_2$	(D,B)	B
3	$(1,2)_3$	(A,B)	C
	$(2,3)$	(B,B)	D
	$(3,3)$	(B,C)	

Figure 12: The final configuration



$G_{13} = (V_{13}, E_{13})$		$\overline{G_{13}} = (F_{13}, \overline{E_{13}})$	
V	E	$\overline{E}$	F
2	$(2,2)$	(D,A)	A
3	$(2,3)$	(A,A)	C
	$(3,3)$	(A,C)	D

Figure 13: Common contracted graph of G_{10} and G_{11}

4 Greatest common contracted graph

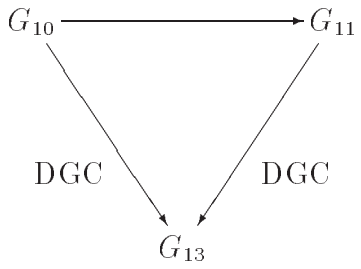


Figure 14: Greatest Common Contracted Graph

Consider the structural changes from Fig. 10 to Fig. 11. It is not difficult to verify that neither of them can be transformed into the other using edge-removal or edge-contraction operations only. However they can be both transformed into isomorphic graphs (Fig. 14): Fig. 13 is the result of merging all background regions (e.g. A and B) partially occluded by the moving object. Formally this corresponds to dually contracting edge (A, B) from $\overline{G_{10}}$, which also removes face B and edge $(1,2)_3$. The resulting double edge between A and D is simplified by dually contracting $(1,2)_2$ into vertex 2. From Fig 11 following operations produce the same graph: dually contract edge (A, B) which deletes B and $(1,4)$; dually contract edges $(1,2)_2, (3,4)_2$ which deletes $1, 4$ and multi-edges $(D, A), (C, A)$.

Obviously common contracted graphs do exist (e.g. a single vertex) but are smaller than the original graphs. To preserve the maximum structural information we may search for the largest such graphs.

5 Conclusion

We presented a few ideas to approach motion analysis differently: instead of recovering complete spatio-temporal measurements we considered the effect of motion on the structure of the observed image. In our examples structure is represented by graphs, and we have shown that

- a view change caused by crossing the alignment of two points can be modeled by a permutation;
- occlusion-caused structural changes can be derived by dual graph contraction and yields common generalized subgraphs;

There are still a lot of open questions. However, our approach can overcome several of the problems mentionned before: Although our results are incremental in terms of a structural change between frames we could use a generalization hierarchy of graphs

to describe motion at a higher degree of generalization. The resulting graphs describe the relations of the moving objects and their changes caused by the motion. The units are the structural entities. Noise has only very little influence on the structure if the image resolution allows a robust identification of the structural entities. In few situations can an accidental error cause a structural error, but this accident can be corrected since the temporal resolution may allow to check consistency and to eliminate the incorrect structural event. The observation that objects change place in the image after being aligned with the observer further constrains the solution space and could be used to identify misinterpretations caused by errors. The computational complexity of dual graph contraction has been shown to be on the order of $\log(diameter)$ of the data.

References

[1] M.J. Black, Yaser Yacoob, and D.J. Fleet. Modeling Appearance Change in Image Sequences. In Carlo Arcelli, Luigi P. Cordella, and Gabriella Sanniti di Baja, editors, *Advances in Visual Form Analysis*, pages 11–20. World Scientific Publishing Company, 1997.

[2] Mark James Burge. *Representation and Analysis of Document Images*. PhD thesis, Johannes Kepler University, 1998. Dissertation.

[3] I. Chakravarty and Herb Freeman. Characteristic Views as a Basis for Three-Dimensional Object Recognition. In *SPIE Proceedings of the Conference on Robot Vision*, pages 336:37–45, 1982.

[4] Cornelia Fermüller. Navigational Preliminaries. In Yiannis Aloimonos, editor, *Active Perception*, pages 103–150. Lawrence Erlbaum, 1993.

[5] Walter G. Kropatsch. From equivalent weighting functions to equivalent contraction kernels. In *CZECH PATTERN RECOGNITION WORKSHOP'97*, pages 1–13. Czech Pattern Recognition Society, February 1997.

[6] Shmuel Peleg and Joshua Herman. Panoramic mosaics by manifold projection. In *Computer Vision and Pattern Recognition (CVPR)*, pages 338–343. IEEE, 1997.

[7] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image Processing, Analysis and Machine Vision*. Computing Series. Chapman & Hall, London, Glasgow, New York, 1993.