A Graph-Based Concept for Spatiotemporal Information in Cognitive Vision*

Adrian Ion and Yll Haxhimusa, and Walter G. Kropatsch

Pattern Recognition and Image Processing Group 183/2,
Institute for Computer Aided Automation,
Vienna University of Technology, Austria
{ion, yll, krw}@prip.tuwien.ac.at

Abstract. A concept relating story-board description of video sequences with spatio-temporal hierarchies build by local contraction processes of spatio-temporal relations is presented. Object trajectories are curves in which their ends and junctions are identified. Junction points happen when two (or more) trajectories touch or cross each other, which we interpret as the "interaction" of two objects. Trajectory connections are interpreted as the high level descriptions.

1 Introduction

Even though there is no generally accepted definition of cognitive vision yet, presumptions about the cognitive capabilities of a system can be made by comparing it's results with that of an entity, already 'known' and accepted to have these capabilities, the human. Also, the *Research Roadmap of Cognitive Vision* [15], presents this emerging discipline as 'a point on a spectrum of theories, models, and techniques with computer vision on one end and cognitive systems at the other'. A conclusion drawn from the previous, is that a good starting point for a representation would bring together the following:

- enable easy extraction of data for human comparison;
- bridge together high and low level abstraction data used for cognitive and computer vision processes.

After 'watching' (analyzing) a video of some complex action, one of the things, that we would expect a cognitive vision system to do, is to be able to correctly answer queries regarding the relative position of occluded objects. Let us take the video[1] given by a simple scenario of two black cups and a yellow ball and describe the scene in simple English words (see the description in Table 1). The description contains: **objects**: hand, cup, ball, table ; **actions**: grasp, release, move, shift etc., and **relations**: to-the-left, to-the-right, in-front-of etc.

* Supported by the Austrian Science Fund under grant FSP-S9103-N04.

[1] http://www.prip.tuwien.ac.at/Research/FSPCogVis/Videos/Sequence_2_DivX.avi

L. Brun and M. Vento (Eds.): GbRPR 2005, LNCS 3434, pp. 223–232, 2005.

Later, we could use this kind of description to compare the results given by the system with ones made by humans. While observing a dynamic scene, an important kind of information is that of the change of an object's location, i.e. the change of topological information. In most of the cases, this kind of change is caused by an active object (e.g. agent: hand, gravity, etc) acting on any number of passive objects (e.g. cup, ball, etc.). Queries like '*where is the ball?*' could be answered if the history of topological changes is created.

From all the work done in the domain of qualitative spatial and temporal information we would like to enumerate the following: Interval calculus [1] is used in systems that require some form of temporal reasoning capabilities. In [1] 13 interval-interval relations are defined: 'before', 'after', 'meets', 'met-by', 'overlaps', 'overlapped-by', 'started-by', 'starts', 'contains', 'during', 'ended-by', 'ends' and 'equals'. In [13], motivated by the work in [1, 7, 8], an interval calculus-like formalism for the spatial domain, the so called region connection calculus (RCC) was presented. The set of 8 region-region base relations defined in [13] ($RCC - 8$) are: 'is disconnected from', 'is externally connected with', 'partially overlaps', 'is a tangential proper part of', 'is non-tangential proper part of', 'has a tangential proper part', 'has non-tangential proper part', and 'equals'. A more expressive calculus can be produced with additional relations to describe regions that are either inside, partially inside, or outside other regions ($RCC - 15$). Different graph based representations have been used to describe the changes/ events in a dynamic space. In [6] graphs are used to describe actions (vertices represent actions). Graphs are also used in [2], but here vertices represent objects. Balder [2] argues that arbitrary changes can be best described by state approach: the state of the world before and after the change characterizes the change completely. The Unified Modeling Language, in its state diagram, also defines a graph based representation for tracking temporal changes. The General Analysis Graph (GANAG) [14] is a hierarchical, shape-based graph that is build and used in order to recognize and verify objects. *The analysis graph can be seen as a 'recipe' for solving industrial applications, stating which kind of decisions have to be made at which stage* [14].

In Section 2 we give the spatiotemporal story-board of the video sequence. In Section 3 we describe two methods of contraction of trajectory of movements: first the spatial contraction followed by a temporal contraction (Section 3.1) and than the temporal contraction followed by a spatial contraction (Section 3.2).

2 Spatiotemporal Story Board of a Film

The scene history is a description of the actions and spatial changes in the scene. It should depict the spatiotemporal changes in the scene, in a way that could be used to create a human-like description (similar to the one presented in Section 4). For this we propose a graph based representation where vertices represent spatial arrangement states and edges represent actions (see Figure 1a).

Each vertex contains a topological description of the spatial arrangement of the objects in the scene, that results through a transition from a previous state,

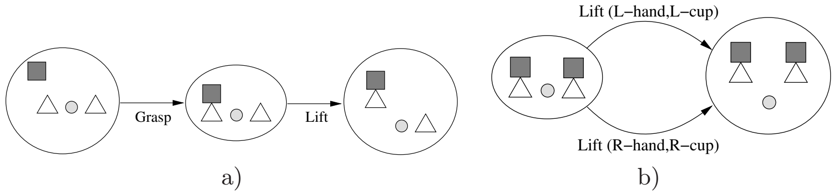

Fig. 1. a) History graph. b) Parallel actions. □ Hand, ○ Ball, △ Cup

by applying the actions that link it to the current. What we refer to as objects are actually detected relevant visual entities, which in the ideal case would be objects or, groups of objects in a "special" physical relation e.g occluding, containing, etc. Vertices are added when the topological description of the spatial arrangement changes. There are no vertices that contain (identify) the same topological description (scene state). If the scene enters a state, which has a topological description identical to one of the descriptions already identified by a vertex in the scene history graph (it has been in the same state in the past), then an edge/edges from the vertex identifying the previous state, to the existing vertex should be added.

Edges are associated with actions and identify the type/class of the action. Also, each edge links to the objects (from the source and destination state vertex) involved in this particular action. If an object taking part in the action cannot be identified as one of the known objects, a new instance should be created and the edge linked to it. Later on, through reasoning, the new created instance, can be identified as a previously known object or a new one (or some presumption can be made, using certain criteria). In case of simultaneous actions, more than one edge is used to connect 2 vertices. Each edge should describe the actions that happened in parallel. (Figure 1b) shows how to describe 2 hands lifting 2 cups at the same time)

The representation of the scene history as a graph allows us to create higher level abstractions. A straight forward example results from the 're-usage' of vertices (disallowing multiple vertices identifying the same state). Imagine the scenario of a hand grasping and releasing the cup 10 times in a row. Besides saving space by not adding a big number of additional vertices, by identifying cycles, we can easily determine repeated actions and find the shortest way from one configuration to another. Higher level abstractions replace more complex subgraphs containing parallel actions and long sequences of actions resulting in small or unimportant changes for the objects in the system's attention.

A type of information that can be directly extracted from the spatiotemporal graph is the one of 'all known actions'. This information can be represented by a directional graph in which vertices represent unique classes of objects part in any previous action and edges represent simple actions that can involve the connected vertices (usually actions that a class of objects can perform on another class). E.g.: a hand can lift, move, grasp, release, etc. a cup.

We can observe that, in time, for a fixed set of classes of objects involved, if the actions vary enough, the graph of 'all known actions' will converge to the graph of ‘all possible actions’ and the presented spatiotemporal history graph, will converge to the graph ‘of all possible states’ (The latter is something that should be avoided, because storing/remembering everything up to the smallest details is guaranteed to sooner or later cause time and memory issues).

Another type of information, that is obtained directly (e.g. tracking) or through reasoning, is that of an object occluding or containing other objects (totally or partially, but still unrecognizable by the detection level). To store this type of information, a relabeling of the class of the occluding object should be done i.e. a cup that has been found out to contain a ball should be labeled ‘cup with ball inside’.

3 Contraction in Spatiotemporal Space

The idea here would be to contract in 3D (2D space + time) along 'the trajectory' of the movements. Every frame could be represented by a region adjacency graph. In order to stretch this into time, these region adjacency graphs (region adjacency combinatorial maps) should be matched to each other, i.e. the region adjacency graph at time t is matched with the one in $t+1$ and so on. In this sense we could define a 'trajectory' of each region This trajectory becomes a curve in 3D and with the techniques analogous with that of contraction of a 2D curve pyramid in [11], we can contract regions adjacent along this curve to produce the more abstract representation of the scene, e.g. where the movement started, where it ended etc (Figure 2).

If the analyzed scene has a structured background, then, depending on it’s granularity, this is enough to detect movement using only topological information. On the other hand, this will increase the number of consecutive frames that differ with respect to topological relations. To reduce the abundance of topological states, to a set containing the most relevant ones, a set of adaptive pyramids is used. There are no constraints regarding the time intervals between 2 consecutive states. Actually, it is expected that in most of the cases where natural movement is present (not robots repeating some predefined action) these time intervals will differ quite a lot.

In subsections 3.1 and 3.2 we present two approaches, to the problem, which basically differ only in the order in which contraction in the spatial and temporal domains, is done. The first, avoids the difficult problem of graph matching by creating pyramids in the first step and then doing the matching using the pyramids. The second, while needing graph matching to be done, should have a lower memory usage. Moreover, in the ideal case, the resulting top level of the 2 approaches should be the same.

3.1 Spacial Contraction Followed by Temporal Contraction

For each frame, whose topological description is different from the one of the previous frame, a space-contraction pyramid is build, that preserves only the spatial information required by the higher functionality levels (i.e reasoning) and by the time-contraction. A space-contraction pyramid is a pyramid where

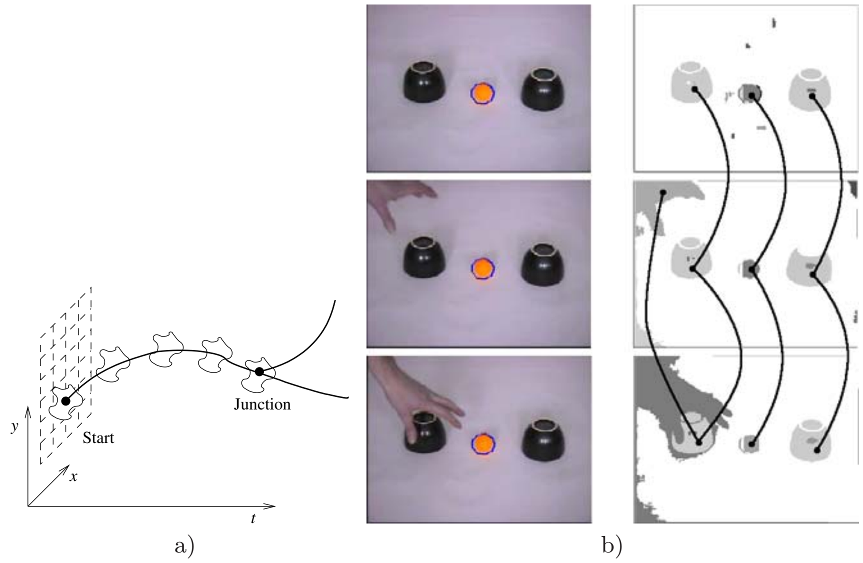

Fig. 2. a), b) Trajectory of movements

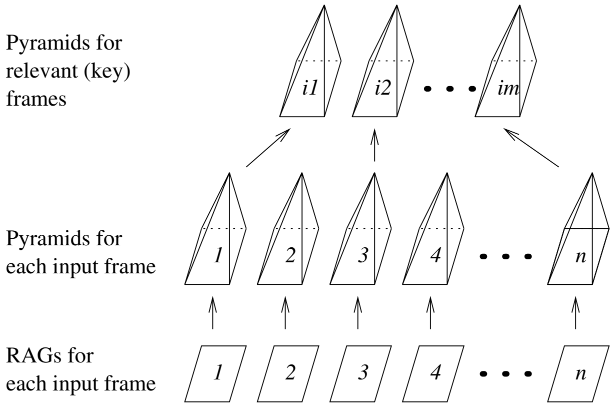
Pyramids for relevant (key) frames
i1
i2
im
Pyramids for each input frame
1
2
3
4
n
RAGs for each input frame
1
2
3
4
n

Fig. 3. Space time contraction

elements, from the same scene state, neighbored from a spatial point of view are contracted, and a time-contraction pyramid is a pyramid where elements, neighbored from a temporal point of view (consecutive scene states) are contracted.

To obtain the base level of the time-contraction pyramid from the set of space-contraction pyramids a matching step has to be performed (Figure 3). Each 2 consecutive pyramids (from a chronological perspective) have to be matched, and the vertices that represent the same object/visual entity should be linked by an edge (if it is possible i.e. if the same object/visual entity exists in both structures - existed in both frames). If a certain object/visual entity, that exists

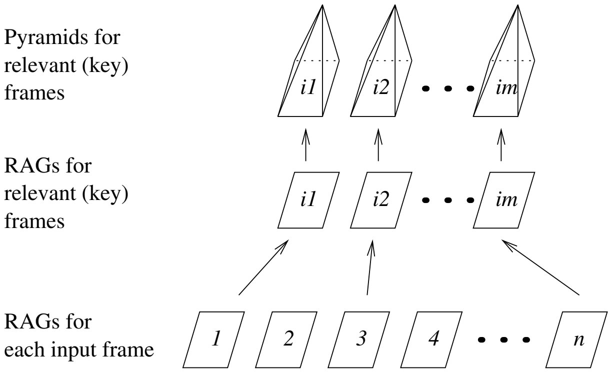
Pyramids for relevant (key) frames
RAGs for relevant (key) frames
RAGs for each input frame
i1
i2
im
i1
i2
im
1
2
3
4
n

Fig. 4. Time space contraction.

in one of the pyramids, does not exists in the other (occlusion, moved out of the field of view, etc.), no connecting edge can be created, thus obtaining a trajectory endpoint. If similar entities disappear and reappear at different time intervals, it will be the job of the reasoning part to decide whether it was the same instance of the same class or not.

The base level of the time-contraction pyramid contains a vertex for each of the frames in the source video, that differ in topological relations from the previous frame. Each vertex will contain the space-contraction pyramid for the region adjacency graph of the respective scene state. These vertices are linked together in a chronological manner i.e. each vertex is linked to the one of the previous and next frames. Also, as a result of the pyramid matching process mentioned before, the vertices from the consecutive space-contraction pyramids are linked together, showing the trajectories of the regions from the first through the last frame. For example: take the topological descriptions for each frame and represent them in a 3D space, where one of the dimensions is time, and the other 2 are used to represent the planar region adjacency graphs. If for every 2 consecutive graphs, the vertices representing the same object/visual entity are linked together by an edge, then following these inter-state connection edges will produce the regions trajectory in 3D space.

Each level of the time-contraction pyramid is a chronologically ordered list of space-contraction pyramids, each element describing the topological relations of a certain scene state. The space-contraction step reduces the spatial information in areas that are not of our interest. The purpose of the time-contraction pyramid is to skip the unnecessary frames caused by the presence of the structured background (which is needed for movement detection using only topological information).

3.2 Temporal Contraction Followed by Spatial Contraction

The base level of the time-contraction pyramid contains a vertex for each of the frames in the source video, that differ in topological relations from the previous frame (Figure 4). Each of these vertices contains the region adjacency graph (RAG) for the respective frames. Through a preliminary process of matching, each vertex in a region adjacency graph should be connected with the vertex(vertices), from the two neighboring graphs, that represent the same object/visual entity (if it is possible i.e. if the same object/visual entity exists in the neighboring region adjacency graphs frame). In other words, the base level of the pyramid is the discretized evolution of the region adjacency graph of the presented scene with the exception that identical consecutive states are merged into a single state.

If we would represent the base level structure in a N dimensional space (3D for 2D state descriptions + time) we would see that we have obtained curves representing the trajectories of the different regions analyzed. A line segment parallel to the time axis, will denote a static region through the respective time interval. Each level of the pyramid is made out of a sequence of region adjacency graphs. Each vertex in a region adjacency graph should be connected with the vertex(vertices), from the two neighboring graphs, that represent the same object/visual entity.

With each new level added to the time contraction pyramid, the number of topological states decreases. After reducing the number of topological states, a contraction of topological information for each state can be considered (at this level the detail regarding the background should not be important any more).

There are 2 ways that can be considered for doing this:

- contract each state independently (create a pyramid for each of the topological states at the top level of the time-contraction pyramid)
- contract all the graphs together (allow contraction kernels to span along more than one state graph)

3.3 Spatiotemporal Entities

The trajectories of (moving) objects (visual entities resulted from segmentation and tracked through the whole time span) represent curves connecting start, end and the junction points. Junction points happen when two (or more) trajectories touch or cross each other, which we interpret as the ‘interaction’ of two objects.

Following the work of Kropatsch [11] the trajectory, which is a curve in 3D, and the cells, which are vertices of the graph, can be related as follows:

0-cell - an empty cell (no trajectory motion within the receptive field)
1-cell - the trajectory starts or ends in this cell (it leaves or enters the cell and intersects only *once* the boundary of the receptive field)
2-cell - the trajectory crosses the receptive field (it intersect *twice* the boundary of the receptive field).
*-cell - a cell where more than one trajectory meet, a junction cell (the boundaries of the receptive field are intersected *more than twice*).
1-edge - trajectory intersects the connected segment boundary of the receptive field.
0-edge - no trajectory intersect the boundary of the receptive field.

It is assumed that: 1) the cells are consistent, i.e. if a trajectory crosses a boundary both cells adjacent to this boundary are in correct classes, and 2) all trajectories are well distinguishable in the base, e.g. there are no more than one single curve in one single cell of the base (except at *-cells).

3.4 Selection of Contraction Kernels

Contraction should be done along the trajectory, like in curve pyramids in 2D [11, 5]. In order to undertake the contraction process, the contraction kernels must be selected. The selection rules are 1-cells and *-cells must always survive. *-cells are not allowed to have children. This prevents the area of unclear information[2] from growing. Branches of contraction kernels follow the trajectory if possible and are selected in following order: 1-cells, 2-cells, 0-cells. Receptive fields are merged as follows:

1. A 1-cell can merge with its adjacent 2-cells, then with any adjacent 0-cell and will become an 1-cell again;
2. a 2-cell can merge with both adjacent 2-cells or with any adjacent 0-cell and remains a 2-cell;
3. a 0-cell can merge with any adjacent cell and remains a 0 cell if it is merged with another 0-cell.

If the rules do not determine the contraction kernels the random selection methods [12, 10, 9] are applied. Applying these rules, the trajectory remains a simply connected curve in spatiotemporal space. At the top level (where no more contraction is possible) we find only 1-cells and *-cells giving on overview of all movements, when and where is started, when and where the cup was grasped, and this is compact for all types.

4 Example

A simple, human language like description of a scene with two cups and a yellow ball is shown in Table 1. Even though the frame numbers are given, they are only for orientation purposes and can be easily eliminated from the description by putting the adverbial for example 'next', 'after that', 'then' etc. The previous description would be represented in the following way (see Figure 5) in the resulting top level of both approaches. The initial configuration contains 3 objects: 2 cups and 1 ball. So we initialize the objects structure with the following: *cup(1)*, *ball* and *cup(2)*. (The numerical ids in parenthesis are present to distinguish the two cups, identification could be done in many other ways. Also in the same interest, vertices are numbered to identify different positions in time.) Vertex(0) in Figure 5 depicts the initial configuration. The next vertices and edges are as follows:

1. action *move*: creates object *hand* and adds *vertex(1)*;
2. action *grasp*: links to objects *hand* and *cup(1)* and adds *vertex(2)*

[2] trajectory may intersect or may be just close to each other.

Table 1. Scene description

cell type	Frame	Description	cell type	Frame	Description
0	16–21:	hand from left	*	91:	grasps the same cup again
*	22:	grasps left cup	*	87–90:	releases it and moves up and down
*	27–30:	moves it over ball	*	85–86:	moves it to the right (but left of the right cup)
*	31:	releases cup	*	84:	grasps it
*	32:	grasp same cup (again)	*	76–77:	moves to the right cup
*	33–36:	shifts it to the left	*	75:	releases it
*	37:	releases cup	*	71–74:	shifts it to the right (but still to the left of the right cup
*	38–40:	moves to right cup	*	70:	grasps it
*	41:	grasps right cup	*	67–69:	moves to the left (most) cup
*	42–58:	shifts right cup in front of left cup (hiding left cup Fr 46–54) to the left of the original cup	*	66:	releases it
*	58:	releases cup	*	63–65:	shifts it to the right
*	59–61:	moves to the other cup	*	62:	grasps it
			. . .	. . .	. . .

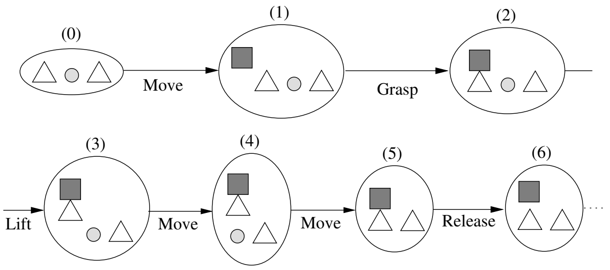

Fig. 5. Example history graph. □ Hand, ○ Ball, △ Cup

3. action *lift*: links to objects *hand* and *cup(1)* and adds *vertex(3)*
4. action *move*: links to objects *hand* and *cup(1)* and adds *vertex(4)*
5. action *move*: links to objects *hand* and *cup(1)* and adds *vertex(5)*
6. action *release*: links to objects *hand* and *cup(1)* and adds *vertex(6)*

Although the presented approaches would work in a different way (one would first try to identify the important visual entities and then key events, while the other would start with the key events and then continue with key entities), the expected result is the same.

5 Conclusion

This paper presents a concept relating story-board description of video sequences with spatio-temporal hierarchies build by local contraction processes of spatio-temporal relations. Since object trajectories are connected curves we identify their ends and junctions and their connections as the high level descriptions. Junction points happen when two (or more) trajectories touch or cross each other, which we interpret as the 'interaction' of two objects. We propose to derive them similar to curve pyramid in 2D [11, 5], For the implementation we plan to use the concept of combinatorial pyramids in 3D [3, 4].

References

[1] J. Allen. An Interval-based Representation of Temporal Knowledge. In *Proc. 7th Inter. Joint Conf. on AI*, p:221–226, 1981.

[2] N. I. Balder. *Temporal Scene Analysis: Conceptual Descriptions of Object Movements.* PhD thesis, University of Toronto, Canada, 1975.

[3] L. Brun and W. G. Kropatsch. The Construction of Pyramids with Combinatorial Maps. Technical Report PRIP-TR-63, Pattern Recognition and Image Processing Group, TU Wien, Austria, 2000.

[4] L. Brun and W. G. Kropatsch. Introduction to Combinatorial Pyramids. In G. Bertrand, A. Imiya, and R. Klette, editors, *Digital and Image Geometry*, p:108–128. Springer, Berlin, Heidelberg, 2001.

[5] M. Burge and W. G. Kropatsch. A Minimal Line Property Preserving Representation of Line Images. *Comp., Devoted Issue on Im. Proc.*, 62:355–368, 1999.

[6] A. Chella, M. Frixione, and S. Gaglio. Understanding Dynamic Scenes. *Artificial Intelligence*, 123:89–132, 2000.

[7] B. Clarke. A Calculus of Individuals Based on Connection. *Notre Dame J. of Formal Logic*, 23(3):204–218, 1981.

[8] B. Clarke. Individuals and Points. *Notre Dame J. of For. Log.*, 26(1):61–75,1985.

[9] Y. Haxhimusa, R. Glantz, and W. G. Kropatsch. Constructing Stochastic Pyramids by MIDES - Maximal Independent Directed Edge Set. In E. Hancock and M. Vento, eds., *4th IAPR-TC15 Workshop on GbR in PR*, LNCS 2726:35–46, York, UK, 2003. Springer Verlag.

[10] Y. Haxhimusa, R. Glantz, M. Saib, G. Langs, and W. G. Kropatsch. Logarithmic Tapering Graph Pyramid. In L. van Gool, ed., *Proc. of 24th DAGM Symposium*, LNCS 2449:117–124, Swiss, 2002. Springer Verlag.

[11] W. G. Kropatsch. Property Preserving Hiearchical Graph Transformation. In C. Arricelli, L. Cordella, and G. Sanniti di Baja, editors, *Advances in Visual Form Analysis*, p:340–349, Singapore, 1998. World Scientific.

[12] P. Meer. Stochastic image pyramids. *Computer Vision, Graphics, and Image Processing*, 45(3):269–294, 1989.

[13] D. Randell, Z. Cui, and A. Cohn. A Spatial Logic Based on Regions and Connection. In *Proc. 3rd Int. Conf. on Knowledge Representation and Reasoning*, p:165–176. Morgan Kaufmann, 1992.

[14] R. Sablatnig. Increasing flexibility for automatic visual inspection: the general analysis graph. *Mach. Vision Appl.*, 12(4):158–169, 2000.

[15] D. Vernon, editor. *A Reasearch Roadmap of Cognitive Vision (DRAFT Version 3.2).* ECVision: The European Research Network for Cognitive Computer Vision Systems, 2004.