Machine Perception using a Blackboard Architecture

Tim P. Guhl, Murray P. Shanahan

Department of Computing, Imperial College London,
180 Queensgate, London, SW7 2BZ, United Kingdom.
{tim.guhl; m.shanahan}@imperial.ac.uk

Abstract.

Here we present ongoing research in the application of symbolic argumentation to perception in general and vision in particular. Perception is treated as the combination of the possibly contradictory outputs of many specialized processes which communicate via a blackboard data structure. It is demonstrated that our design allows for bottom-up, horizontal and top-down information flow. Progress towards the analysis of unstructured scenes has been made. The principles involved have been explored experimentally and preliminary results are presented.

1. Introduction and Related Work

Research in mainstream computer vision has led to many successful and commercially exploitable systems for object recognition in carefully controlled environments, or where the set of objects to be recognised is small and known in advance. By contrast, the challenges of unconstrained object detection and recognition in uncontrolled, everyday environments are still very much open. These, however, have to be overcome before we can build the robots of the future that, for example, can assist an elderly person with the running of the household.

Let us suppose that such a robot is pre-programmed with a large repertoire of visual Specialised Processes (SPs) that obtain the information required to carry out familiar tasks such as making a cup of tea. However, in everyday life it will also come across unfamiliar objects. It has to be able to detect those objects (a) to avoid bumping into them, (b) to assess their significance to determine whether to try to obtain further information about them and (c) to see whether they can be used. In our system a combination of Low-Level Processes (LLPs) and Knowledge Sources (KSs) analyse the scene to find these more generic objects.

We claim that a system using a combination of SPs, which only return their often task-specific, but relatively reliable and accurate results occasionally, and LLPs and KSs, which return lower quality results more frequently, is capable of tackling the vision problem in an unstructured environment. By this we mean it can detect familiar and unfamiliar objects, returning an accurate description of the object and its pose in the former case and more general, spatial information in the latter.

The main contribution of this work is the provision of an architecture in which the combination of many heterogeneous processes solves this complex vision problem.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

The competition for access to a blackboard data structure [6] facilitates cooperation among these processes without the need for direct communication between them. The computational challenge can therefore be met using a parallel architecture, potentially by implementing parts of the system in programmable hardware.
The expected payoff for combining a blackboard architecture with three-way information flow (bottom-up, horizontal and top-down) is a system for object detection that will be fast and robust in the presence of uncertain and incomplete data. The rationale for this expectation is that high-level inference allows many small pieces of unreliable evidence to be accumulated and combined into an accurate and complete overall picture.
As a benchmark problem, the developed architecture has been tested using a stereo-camera head with a fixed baseline observing unstructured scenes using single or continuous capture with the aim of detecting objects "touchable" for example by a humanoid robot. The system generates a high-level symbolic description of the spatial arrangement of the scene. It is further demonstrated that, as environmental conditions (such as the lighting) change and the contributions made by one KS are rendered useless, other KSs take over smoothly.
Shanahan [8, 9] developed a theoretical, logic-based framework, which employed a combination of deduction and abduction to implement two-directional information flow in a cognitive vision system. Expanding this work, we use a blackboard architecture (BBA) [6] to allow us to integrate possibly contradictory information from a large number of image processing algorithms with the aid of symbolic and preferential reasoning [10]. Each of these processes can be considered an individual sensor, making this a sensor fusion task for "multi-modal" sensors tackled at decision level [4, 2]. The quality of each hypothesis formed is evaluated using preferential grouping [1] and argumentation theory [3]. We are not aware of any similar approaches to solving this vision problem.

2. Information Processing

In our system the tasks performed by Specialised Processes (SPs), Low-Level Processes (LLPs) and Knowledge Sources (KSs) can be grouped into 5 classes and many of them can be executed simultaneously. The direction of the information flow for each class is given in brackets:

- **SPs (bottom-up):** Specialised Processes generate hypotheses directly from the raw sensor data. They only operate bottom-up (e.g. haar-classifier for face detection).
- **LLPs (bottom-up):** LLPs derive symbolic low-level features, in most cases from the raw image data (e.g lines), but sometimes from other features (e.g corners).
- **KSs (bottom-up):** KSs use their background knowledge Σ to form hypotheses Δ to explain the features Γ found such that $\Sigma \wedge \Delta \models \Gamma$ [7]. In other words, the KSs use knowledge about the world to publish hypotheses to explain the sensor data.
- **KSs (horizontal):** Other KSs can increase the informational value of hypotheses posted by others by adding supporting or contradicting evidence. This is possible even where this evidence by itself did not warrant the posting of a hypothesis.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

- **KSs (top-down):** KSs can add noise terms to represent expected-but-not-detected features. As these share a representation with the equivalent detected features, the relevant LLP(s) can confirm or rule out their existence by revisiting the image data.

Object hypotheses formed can at times be mutually exclusive as they describe the same object in different ways, or they use a feature for several objects. The scene analysis forms scene hypotheses from non-conflicting objects. By combining the confidence values of the objects, a qualitative measure for the competing scene interpretations can be derived and the most likely selected. Confidence values are quality measures assigned to features and hypotheses (Section 3.3).

3. Knowledge Representation

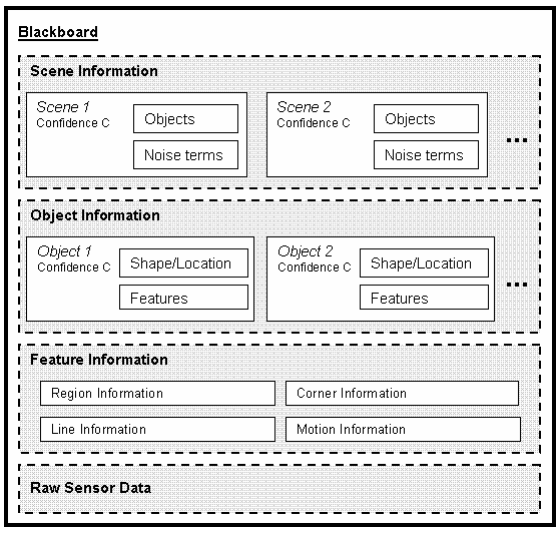

Fig. 1. Data structures on the blackboard

Data stored on our blackboard (Fig. 1) is grouped into the following 4 hierarchical layers: raw sensor, feature-, object- and scene information. The raw sensor data level contains the image data and edge map. From this LLPs predict feature information, which in turn is used by the KSs to derive hypotheses and to publish them at the object information level. Finally, these can be analysed by the scene interpretation KS. Information in these higher layers is stored in a symbolic form, e.g. line(from,to).

3.1. Data structure

The data structure describing an item of data at any level above raw sensor data consists of a public and a private section. While the information in the public part is available to any process, the private data can only be read by those that share the particular representation used. The item's ID and confidence value are public. In the case of an object hypothesis generated by a KS, the public part also includes a general

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

description of the outline of the object to allow other processes to contribute. The content of the private section depends on the type of data. In the case of a line, for example, this is its start and end points. For an object, this section contains lists of the features associated with the hypothesis. Each such feature has a corresponding support value representing its individual relevance to the hypothesis. The combination of the confidence and support values of the features of the hypotheses allows us to select the most successful hypotheses as described in section 5.
Due to this data structure the system is easily expandable. A new process will need to understand the representation of its input data and translate its outputs into the form described here. Other processes can use any results it then publishes on the blackboard. Information can flow horizontally as other KSs can contribute to hypotheses it posts. This also allows for several processes producing the same type of output (e.g. two different line finding algorithms).

3.2. Support Value

The support value determines the impact a feature or object can have on the overall confidence value of the structure it is associated with. There are six distinct levels:

1. **HasToExist:** If the feature exists, its positive effect depends on the confidence we have in the feature. If it doesn't exist, the hypothesis can not be true. This support value is assigned to any feature that a hypothesis is based on when it is initially created. It is often used if a feature changed the overall description of the object.
2. **ShouldExist:** If the feature exists, it again has the full positive effect, otherwise the effect is equal in absolute value but negative. A KS assigns this level if a feature is of importance to the hypothesis but it isn't imperative.
3. **GoodIfExists:** If the feature exists, it has a positive effect proportional to the confidence we have in the feature. If it can not be found, this doesn't influence our confidence in the hypothesis. If a KS determines that a feature's absence doesn't necessarily contradict its truth value, it assigns this level of support.
4. **GoodIfDoesnotExist:** This is the inverse of GoodIfExists.
5. **ShouldnotExist:** This is the inverse of ShouldExist.
6. **MustnotExist:** This is the inverse of HasToExist.

3.3. Confidence Value

The confidence value is a measure of the quality of the data item as determined by the process that posted it on the blackboard. Preferential grouping is employed to specify the quality of the features. The bottom two groups are for features that have been confirmed to not exist ("0") and for expected-but-not-yet-detected ("1") features respectively. A sensitivity study to determine the optimum number of groups above these levels still has to be carried out. The experiments in section 4 were conducted with 3 "positive" groups (so a total of 5).

- **Feature Information:** Here confidence is a measure of the quality of the sensor data the prediction of the features is based on. Here we include two examples to demonstrate the general idea of how these values can be obtained:

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

The confidence value CL of a line was calculated from its relative length LL compared to the longest line ML, the number of pixels found per unit length LP of the line relative to the highest number of pixels found per unit length MP of all the lines and the straightness LS of the line relative to the straightness MS of the straightest line as measured by regression. These three factors are weighted by the experimentally derived, constant values WL, WP and WS respectively.

$$CL = (WL * LL / ML) + (WP * LP / MP) + (WS * LS / MS) \quad (\mathbf{1})$$

The confidence value CC of a corner can be calculated from CL1 and CL2, the confidence values of the two lines of the corner respectively, and the angle Θ between these lines:

$$CC = ((CL1 + CL2) / 2) * abs(\sin(2 * abs(\Theta))) \quad (\mathbf{2})$$

With this formula we express our preference for corners made of lines with high confidence values that are either at right angles to each other or close to being one straight line. The system therefore favours rectangular or circular objects.

- **Object & Scene Information:** The confidence in a hypothesis is calculated from the features associated with it and the KSs that contributed. This part of the work is still under investigation and several possible methods are being explored as described in section 5. All proposed methods have two important properties: firstly, the process assessing the overall confidence of a hypothesis neither needs to understand anything about the features assigned to it nor how they are related to each other. Secondly, should the confidence value of some feature change for example because a noise term was found, this new information will automatically propagate up to the object- and scene information levels. These properties ensure the system is robust and easily expandable.

3.4. Noisy and Incomplete Data

Vision data is generally incomplete and noisy. Two types of noise are relevant here:

- **False Detection**: A process finds a feature/object that does not actually exist and can consequently contradict an otherwise correct hypothesis. When this situation occurs, or any other feature contradicts a hypothesis, the process will add the feature to the hypothesis using a support value of 4 or above. In some cases a correct hypothesis might therefore be assigned to low a confidence value due to interfering noise.
- **Omission:** When observing a scene the system fails to detect a feature/object that should be present (e.g. the third corner of a triangle). Using their background knowledge, KSs can sometimes notice missing features. Such expected-but-undetected features are added to the blackboard as noise terms. Their structure and the information contained are equivalent to the corresponding features, but their confidence value is set to "1". The LLPs can now go back to the original data and reassess the situation (e.g. by changing thresholds locally). After confirming or disconfirming the existence of the sought feature, they set the confidence value to a confidence group value ">1" or "0" respectively.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

4. Experimental Apparatus

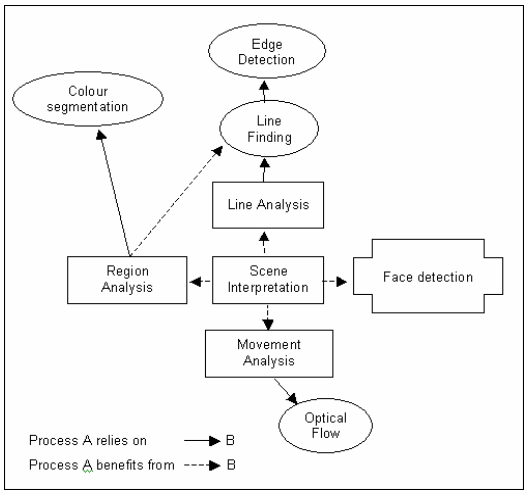

Fig. 2. KS Interdependence (LLPs, KSs and SPs are represented by ovals, squares and "crosses" respectively)

This experimental set-up was designed for a limited domain to test and prove the principles of this architecture and to compare different methods of evaluating the hypotheses. At this time one SP, several LLPs and some of the corresponding KSs were chosen to allow for bottom-up, top-down and horizontal information flow and limited scene interpretation (Fig. 2). Due to the limited quantity, quality and chosen combination of LLPs and KSs, the system was best at picking up on certain types of objects (e.g. JointCorners for objects with round borders; BestCorners for cuboids). For a real world robot, a multitude of SPs would be included to allow it to perform the tasks it was designed for and many KSs would be required to deal with a wide range of unknown objects. Further, it can be expected that the performance of the system could be increased by replacing our off-the-shelf algorithms by the latest research.
For our experiments we used a 3GHz P4 with 1GB of RAM with the firewire MEGA-D colour camera from Videre Design. Most experiments were carried out at 320x240 pixels. The camera was either handheld or mounted on a mini-tripod. Most of the sources were implemented in C++ and OpenCV.

- **EdgeDetection:** Two edge detection processes were used: (1) A Sobel edge detector [5] using a 3x3 grid and (2) the Canny operator of the OpenCV Library. In both cases we thresholded the output.
- **LineDetection:** A regression line finding algorithm and a Hough transform were used to find lines from the output of the Sobel and Canny processes respectively.
- **CornerDetection:** Two lines with adjacent ends were considered to be a corner.
- **MotionDetection:** To detect motion we compared the last 4 frames and then separated motion patches from each other and from global motion. Each patch was approximated by a polygonal curve. The confidence value depended on how uniform the motion was across the patch.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

- **RegionFinding:** Using OpenCV-functions, the area found by a seeded region growing algorithm was approximated by polygonal curves. The confidence was calculated from the proportion of the polygon area covered by the grown region.

One SP was implemented to test whether the combination of the outputs of SPs and LLPs/KSs delivered the desired results. A Haar Classifier for face detection from the OpenCV Library was used for the implementation.

While all data from the feature level upwards was represented symbolically, it was mostly manipulated algorithmically. The following KSs have been implemented:

- **BestCorners (bottom-up and horizontal):** A really good corner was considered a clue for the existence of an object. The five best corners found were used.
- **JoinCorners (bottom-up and horizontal):** If two corners shared a line, they were considered joined. The greater the number of joined corners, the better the clue.
- **Motion (bottom-up):** Motion patches of a similar nature were grouped, the convex hull around them being the motion area. Confidence was calculated from the correlation of the motion direction and speed of the patches and the proportion of the total associated area covered by the detected motions.
- **SeedCorners (top-down and horizontal):** Given a corner, this KS tried to find colour regions between the two lines.
- **SeedHypotheses (top-down and horizontal):** This KS searched for one or more regions such that the area of the hypotheses was filled.
- **History (top-down):** The best hypotheses from the previous frame were used to check whether the object might be present in the current frame.
- **RegionCorners (top-down):** This KS tried to find corners that coincide with the corners of the current outline of the object hypothesis.

Just as important as finding the objects is to know which of your hypotheses are the most successful ones. To allow for any combination of processes to work together this evaluation has to be independent of the different sources and data types. Several methods were tried and compared individually and in different combinations:

- **CountSources:** The number of contributing sources was important.
- **CountWeighSources:** The number and the type of source were considered.
- **CountFeatures:** How many features are associated with the hypothesis?
- **CountWeighFeatures:** Evaluated the hypothesis by the confidence of its features.
- **CountFeaturesWeighSources:** Here the features were counted and while their individual confidences were not important, the system considered their origins.
- **WeighFeaturesWeighSources:** Additionally to considering the confidence and support values of the features, we gave their sources different weightings.

5. Results

Two factors are important when evaluating this system. Firstly, how well does the system detect objects? Performance was measured by comparing the hypothesis area with the object area. Objects covered 90-110% by the hypotheses were considered correctly identified. Secondly, does the system recognise the best solution? If the program identified one of the correct hypotheses as the best solution, the evaluation process was considered successful even if better approximations were rated lower.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

Experiments were carried out on several different types of environments. In each category several different scenes were tried. The following properties were observed:

- **Elementary objects in a structured environment** (Fig. 3): Background noise was eliminated as far as possible with single colour background and by usage of lighting. In at least 95% of runs the system was able to generate hypotheses which covered at least 98% and no more than 102% of the object's surface. While it did not always select the optimum solution, according to the definition above it succeeded in over 95% of the runs.
- **Elementary objects in an unstructured environment** (Fig. 4): With everyday backgrounds and no additional lighting set up but using objects well suited to the LLPs and KSs, the system would still correctly detect objects in 90% of cases. A correct hypothesis was identified in approximately 2/3 of the experiments.
- **Everyday objects in an unstructured environment** (Fig. 4): The system detected other objects as long as they had properties that suited the limited number of LLPs and KSs (e.g. they were moving or they had some straight lines and corners). Here the success rate varied a lot depending on the type of object used. Only weak correlations between what human observers would consider the order of hypotheses and the ranking the system calculated could be identified.
- **Changing environmental conditions** (e.g. lighting; Fig. 6-7): Processes react very differently to such changes. For example lack of light can reduce contrast and therefore the number of lines found. Motion is affected to a much lesser extent. During the limited number of experiments carried out it was observed that, as the results produced by one combination of processes deteriorated, others more suited to the new conditions would smoothly take over in approximately 3 out of 4 runs
- **Specialised tasks:** The output of a face detection algorithm from the OpenCV library was successfully integrated with information from other sources.

Fig. 3. A ball detected in the structured environment by JoinCorners. RegionCorners and SeedHypotheses further contributed.

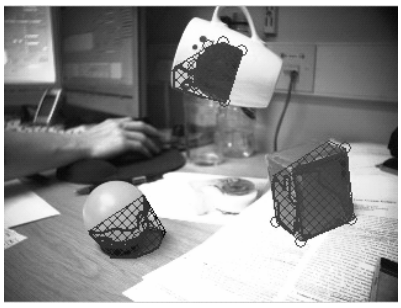

Fig. 4. Unstructured environment with elementary (ball and cube) and everyday (mug) objects. The system drew what it considers the best hypotheses.

The following, more general observations were made:

- Using more channels results in better or equal hypotheses. In Fig. 5 it can be seen that the combination of all channels was always closest to the solution (100%).
- The ultimate confidence value of a hypothesis was independent of the order in which processes contributed to it by adding features and noise terms.

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

- The order in which processes posted new hypotheses did not make a difference if the system was not interrupted prematurely.
- When confidence values changed, the new values propagated through the system.

Further examples can be found online at http://www.doc.ic.ac.uk/~tpg99.

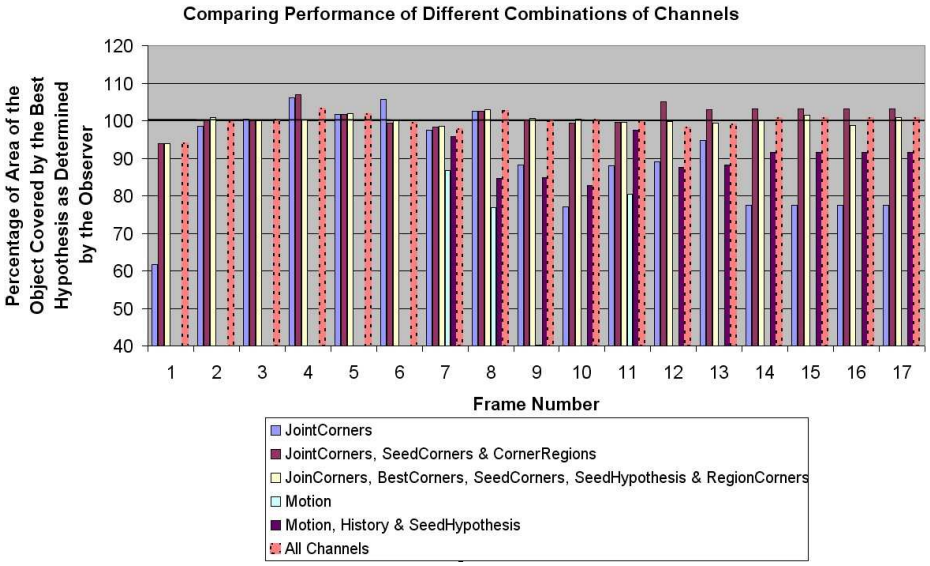

Fig. 5. The best hypotheses found by different combinations of KSs (selected by observer).

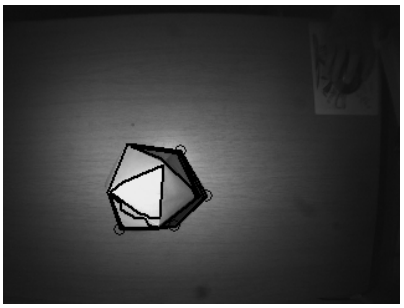

Fig. 6. Block world object in low light. The hypothesis was added by JoinCorners.

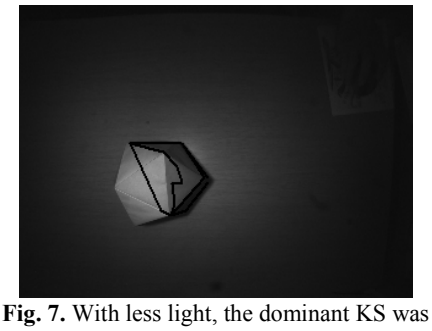

Fig. 7. With less light, the dominant KS was Motion. While the quality of the hypothesis decreased, the object was still detected.

6. Conclusions

At this time a SP and several LLPs and KSs have been implemented allowing for bottom-up, top-down and horizontal information flow and limited scene interpretation. When observing simplified objects correct hypotheses were formed and identified as such in most of the experiments run. Everyday objects were often

Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007)
Published in 2007 by Applied Computer Science Group, Bielefeld University, Germany, ISBN 978-3-00-020933-8
This document and other contributions archived and available at: http://biecoll.ub.uni-bielefeld.de

detected but the hypotheses were of a lower quality. When evaluating hypotheses a general order was established in most cases, but the system had problems detecting outliers such as colour regions spilling out of an object. We conclude further work is required to achieve acceptable results. A sensitivity study has to be carried out with regards to the number of preference groups for each datatype. The experiments indicate that by including more LLPs and KSs more varied scenes will be observable. This could include more complex analysis such as observation of interacting objects.
Due to the prioritisation of hypotheses based on the confidence values, computational explosion was not a problem. With the introduction of parallelism, we are confident the architecture will scale up to include hundreds of processes. To further increase performance the architecture will be deployed in programmable hardware.
The principles developed here emphasise vision. Since there is no restriction on the sources of low-level sensor data, not only new vision processes, but also other types of sensors could be gracefully integrated into an existing system.

7. Acknowledgments

This paper is dedicated to the late Patrick Purcell, a remarkable man who touched our lives and research in many ways. Many thanks to Andreas Fidjeland, Mark Witkowski, David Purcell and many others that helped with advice and support. The presented work was carried out under research grand EP/C51050X/1 from EPSRC.

8. References

[1] S. Bistarelli, M.S. Pini, F. Rossi, and K.B. Venable. Positive and Negative Preferences. 2005. Proceedings CP2005 Workshop on preferences and soft constraints. 1-10-2005.
[2] J.J. Clark. & A.L. Yuille. Data Fusion for Sensory Information Processing Systems. Kluwer, 1990.
[3] P.M. Dung,, R.A. Kowalski, and F. Toni. Dialectic proof procedures for assumption-based, admissible argumentation. Artificial Intelligence. Vol. 170, No. 2, pp. 114-159, Feb.2006.
[4] D. L. Hall and J. Llinas. Handbook Of Multisensor Data Fusion. CRC Press LLC, 2001.
[5] A. Koschan. A Comparative Study On Color Edge Detection. Asian Conference on Computer Vision ACCV, Vol. 3, pp. 574-578, Dec.1995.
[6] H. P. Nii. The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures. In The AI Magazine, 7(2):38-53, 1986.
[7] Murray Shanahan. High-Level Robot Control Through Logic. Proceedings ATAL 2000, published as Intelligent Agents VII, pages 104-121, Springer-Verlag, 2001.
[8] M.P. Shanahan. A Logical Account of Perception Incorporating Feedback and Expectation. Principles of Knowledge Representation and Reasoning, Proceedings of the Eighth International Conference, pp. 3-13, 2002.
[9] M.P. Shanahan. Perception as Abduction; Turning Sensor Data into Meaningful Representation. Cognitive Science, Vol. 29, pp. 103-134, 2005.
[10] T. Son and E. Pontelli. Planning with preferences using logic programming. In V. Lifschnitz and I. Niemela, eds.. Proceedings of the 7th Interantional Conference on Logic Programming and Non-monotonic Reasoning (LPNMR 2004), No 2923 in Lecture Notes in Computer Science, Springer, Pages 247-260, 2004.