

Probabilistic Adaptive Agent Based System for Dynamic State Estimation Using Multiple Visual Cues

Alvaro Soto and Pradeep Khosla

Robotics Institute, Carnegie Mellon University

Abstract

Most of current machine vision systems suffer from a lack of flexibility to account for the high variability of unstructured environments. Here, as the state of the world evolves the information provided by different visual attributes changes, breaking the initial assumptions of the vision system. This paper describes a new approach for the creation of an adaptive visual system able to selectively combine information from different visual dimensions. Using a probabilistic approach and uncertainty metrics, the system is able to take appropriate decisions about the more relevant visual attributes to consider. The system is based on an intelligent agent paradigm. Each visual algorithm is implemented as an agent, which adapts its behavior according to uncertainty considerations. The proposed system aims to achieve robustness and efficiency. By combining the outputs of multiple vision modules the assumptions and constraints of each module are factored out resulting in a more robust system. Efficiency is achieved through the on-line selection and specialization of the agents. An implementation of the system for the case of human tracking showed encouraging results.

1. Introduction

As the state of the art of computing technology is advancing, providing more powerful and affordable machines, computers are becoming widely used in diverse aspects of modern society. As computers start to perform new types of tasks in less structured and less predictable environments, there is an increasing need to provide them with a higher degree of awareness about the changing conditions of their virtual or natural surrounding.

In particular, the case of visual perception is a very attractive option to capture information from a natural surrounding. In contrast to other sensor modalities, vision can allow to perceive a large number of different features of the

environment such as color, shape, depth, motion, and so on. This multidimensionality of visual information is one of the key strength that explains the great robustness observed in most advanced biological visual systems [1].

Most current machine vision systems show a lack of flexibility to consider the wide variety of information provided by visual data. The typical approach relies on simplifications of the environment or on good engineering work to identify relevant visual attributes that allow solving a specific visual task. For example, consider the case of a robot localization system based on artificial visual landmarks. In this case, previous knowledge about the visual appearance of the landmarks provides strong constraints that allow constructing algorithms especially designed to detect the key visual attributes [2].

The main problem with this approach is a lack of flexibility to account for the great variability of most dynamic unconstrained environments. Problems such as changes in the field of view, partial occlusion, changes in illumination, or different postures constantly modify the information provided by the different visual attributes. As a consequence, there is a high variability about the more adequate set of visual attributes to extract the knowledge needed to complete a task.

As an example, consider a stationary visual system designed to track people using an intensity based background subtraction algorithm [3]. Under a slowly varying background the system will be very robust. Now in situations of heavy moving shadows, heavy wind, or people wearing clothes similar to the background, the system will perform poorly. In these cases an algorithm based on segmentation using depth information [4] can help to reduce the ambiguities, but at the expense of extra processing time. Now, if the system needs also to keep the identity of the targets, an algorithm based on color information can provide key information to identify each target, especially if they exhibit different colors. For the case of similar colors, it is possible that an algorithm based on shape or texture information can help to reduce any further ambiguities.

The key observation in the previous example is that as the state of the world evolves, the potential knowledge provided by different visual attributes can change dramatically, breaking the initial assumptions of the vision system. So in order to keep a balance between robustness and efficiency, it is crucial to incorporate suitable adaptation mechanisms that allow to combine and to select the more appropriate set of visual features.

In this paper we propose a new probabilistic approach for the creation of an adaptive visual system able to selectively combine information from different visual algorithms. The basic scenario is an agent embedded in an unpredictable and dynamic environment. The agent is able to receive different types of visual information from the environment. As new information arrives, the goal of the agent is to select the more adequate information in order to update the knowledge about its relevant part of the world.

The state of the world is characterized using a state space representation. For example, for the case of visual tracking of a single target, the state of the system is represented by 4 state variables (x,y,w,h) which determine a bounding box surrounding the target: (x,y) represent the center of the box in the image plane, (w) its width, and (h) its height. The goal of the system is to keep track of a joint

probability density function (jpdf) over these state variables. The level of uncertainty in this state estimation is the key element used by the system to select the more adequate visual information sources. For example, if we are tracking a target using color information and there is a high level of uncertainty about the position of the target, the system will automatically activate new sources of visual information to reduce the current ambiguities.

Using the power of probability theory for representing and reasoning under uncertainty, and elements from information theory to lead the inference engine to prominent hypotheses and information sources, the proposed system aims to achieve robustness and efficiency. By combining the outputs of multiple visual routines the assumptions and constraints of each module are factored out resulting in a more robust system. Efficiency is achieved through the on-line selection and specialization of the agents according to the quality of the information provided by them.

The research proposed in this work is particularly relevant for the case of dynamic visual tasks with a high variability about the subsets of visual attributes that can characterize relevant visual structures. This includes visual tasks such as dynamic target tracking, obstacle detection, and identification of landmarks in natural scenes. In particular, the advantages of the approach proposed here are demonstrated for the case of human target tracking using a mobile robot.

This paper is organized as follows. Section 2 describes our approach and its main components. Section 3 presents related work. Section 4 describes an implementation of the intended system. Section 5 describes the results of the implementation. Finally, section 6 presents relevant conclusions.

2. Probabilistic Adaptive Agent Based System

The system is based on an intelligent agent paradigm. Each visual information source is implemented as an agent that is able to adapt its behavior according to the relevant task and environment constraints. The adaptation is provided by local self-evaluation functions on the agents. These functions are based on considerations about the level of uncertainty present at each time in the state estimation. Cooperation among the agents is given by a probabilistic scheme that integrates the evidential information provided by them.

2.1. Intelligent Agents

There is a general agreement that the main features that distinguish an intelligent agent are *autonomy*, *sociability*, and *adaptation* [5]. Autonomy provides the independence that allows the agent to exhibit an opportunistic behavior in agreement with its goals. Sociability provides the communication skills that allow the agent to interact with other artificial agents and humans.

Adaptation provides the flexibility that allows the agent to change its behavior according to the conditions of the environment.

This work makes use of multiple agents that can simultaneously analyze different dimensions of the incoming information. Each agent is implemented as an independent visual algorithm. In this sense the agents act as a group of experts where each agent has a specific knowledge area.

The rest of section 2 describes the probabilistic representation used in this work, and how the system provides the agents with sociability and adaptation mechanisms. Autonomy is given by a distributed multi-agent software architecture briefly described in section 4.

2.2. Probabilistic Representation

In this work we use a probabilistic approach, which allows accounting for the inherent ambiguity of most unconstrained scenarios. The basic idea is to keep track of a probability distribution over a set of possible hypotheses. Each hypothesis represents a possible configuration of the objects of interest, for example the (x,y,w,h) state of a target.

In this work we use a Bayesian framework for reasoning under uncertainty. Assuming that at time t the current visual evidence e_t can be totally explained by the current hypothesis h_t , and that the dynamic of the system follows a first order Markov process, it is possible to obtain (1), which is the standard way to perform Bayesian inference for the dynamic case.

$$P(h_t / \vec{e}_t) = \mathbf{b} * P(e_t / h_t) * \sum_{h_{t-1}} P(h_t / h_{t-1}) * P(h_{t-1} / \vec{e}_{t-1}) \quad (1)$$

Where \vec{e}_t contains all the historic evidence until time t , and \mathbf{b} corresponds to a normalizing factor.

Equation (1) is a recursive formulation that requires knowledge about the observation model $P(e_t / h_t)$ and the system dynamics $P(h_t / h_{t-1})$. The observation model or likelihood function evaluates the fitness between each sample hypothesis and the observations. The system dynamics determines the level of exploration for new hypotheses as the system evolves.

In practice, except for the case of some finite state-space Hidden Markov models, full Bayesian inference is only possible when the models have suitable analytical expressions. The more typical case is linear-gaussian models. For this case, the pdf over the possible states of the system (state-pdf) remains Gaussian at all times, and the well-known Kalman Filter gives the optimal solution. For the case of nonlinear models, it is possible to use the Extended Kalman Filter, but still under a Gaussian assumption.

The Gaussian assumption severely limits the use of Bayesian inference for state estimation. High ambiguity is one of the inherent features that emerges in most unstructured environments. In this case the state-pdf can have a complex multimodal shape that cannot be accurately modeled by a Gaussian density. Fortunately

stochastic sampling provides an alternative and efficient estimation approach for these cases.

In stochastic sampling a pdf is represented through a set of samples, each with an associated weight representing its probability. The great advantage is that it is possible to approximate any functional non-linearity and system or measurement noise. In this paper we approximate equation (1) using a particle filter approach, also known in the literature as Bootstrap filter [6], Condensation algorithm [7], or sequential Monte Carlo [8]. Figure 1 shows pseudo code for the operation of the algorithm. Starting from an initial approximation of the state-pdf given by n sample hypothesis h_i , the algorithm uses the system dynamic $P(h_i/h_{i-1})$ and its current belief B_t to propagate the more prominent hypotheses. Then these candidate hypotheses are weighted according to the support received by new incoming evidence e_t , represented as a likelihood function $P(e_t/h_i)$. The nice feature about the particle filter is the dynamic allocation of the sample hypotheses h_i according to the current belief. This helps to reduce the problem of sample depletion, and allows a great efficiency in the representation.

```

At  $t = 0$ 
Sample  $h_i$   $_{i=1}^n$  from initial prior  $P(h)$ 
set  $B_t = \{ \mathbf{p}_i, h_i \}_{i=1}^n$  with  $\mathbf{p}_i = P(h_i)$ 

For  $t = 1, 2, \dots, N$ 
  For  $i = 1$  to  $n$ 
    Sample  $h_{i-}$   $\sim B_i$ 
    Sample  $h_i \sim P(h_i / h_{i-})$ 
    Evaluate  $\mathbf{p}_i = P(e_t / h_i)$ 
  end
  Normalize weights  $\mathbf{p}_i = \frac{\mathbf{p}_i}{\sum_i \mathbf{p}_i}$ 
  Set new  $B_t = \{ \mathbf{p}_i, h_i \}_{i=1}^n$ 
end

```

Fig. 1. Pseudo code for particle filter algorithm

2.3. Sociability

The integration of information is performed using Bayes nets. Figure 2 shows the typical tree structure of the Bayes nets relevant to this work. Agent nodes directly measure different dimensions of the incoming visual information, such as color or shape. Abstraction nodes allow the integration of information and the updating of the state representation. Also, abstraction nodes allow introducing conditional independence relations among the visual agents.

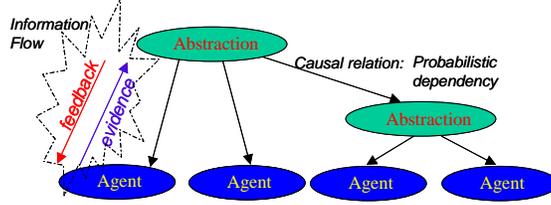


Fig. 2. Bayes Net

Figure 2 can be considered as a hierarchical representation of the simpler case of just one abstraction node. In this case equation (1) can be expressed as:

$$\begin{aligned}
 P(h_t / \vec{e}_{1,t}, \vec{e}_{2,t}, \dots) &= \mathbf{b} * \overbrace{P(e_{1,t} / h_t) * P(e_{2,t} / h_t)}^{\text{Agents}} * \dots \\
 &\dots * \sum_{h_{t-1}} \underbrace{P(h_t / h_{t-1}) * P(h_{t-1} / \vec{e}_{1,t-1}, \vec{e}_{2,t-1}, \dots)}_{\text{Abstraction node}} \quad (2)
 \end{aligned}$$

Where $\vec{e}_{i,t}$ corresponds to the historic evidence provided by agent i until time t . Equation (2) shows the decoupling between the evidence provided by each agent through a likelihood function, and the state updating performed by the abstraction node. The abstraction node acts as a central inference that keeps track of the state estimation represented by a set of sample hypotheses and their probabilities. Using the current estimations and the system dynamics, the abstraction node decides which hypotheses need further considerations and it sends this new set of hypotheses to each of the agents. According to its own local information sources, each agent evaluates the supporting evidence for each hypothesis, and it sends this evidence back to the abstraction node as a likelihood function. Finally the abstraction node uses this information to update its beliefs about the state of the world, and it starts a new iteration of the state estimation by predicting a new set of relevant hypotheses.

2.4. Adaptation

In contrast to most traditional applications of Bayes nets, where the structure of the net is fixed, the system intended in this research includes adaptation mechanisms that allow a dynamic reconfiguration of the net according to the characteristics of the incoming information.

The adaptation mechanisms are based on the evaluation of the level of uncertainty present in the state estimation and the evaluation of the quality of the information provided by each agent in terms of uncertainty reduction. The design goals are to perform robust estimation keeping uncertainty low, and also to perform an efficient estimation avoiding the processing of irrelevant, misleading,

or redundant information. In order to achieve these goals it is needed to introduce two performance metrics.

The first metric called uncertainty deviation (UD) is intended to evaluate the level of uncertainty in the state representation. The intuition behind this metric is to quantify the dispersion of the state representation with respect to the most probable hypothesis known as maximum a posteriori hypothesis (MAP). Equation (3) shows this metric; here d corresponds to a distance metric between hypotheses:

$$UD = \sum_{h_i}^{|h|} d^2(h_i, MAP) * P(h_i) \quad (3)$$

For the case of Euclidean distance and state variables (x,y,w,h) , equation (3) can be expressed as :

$$UD = \sum_{h_i} \sum_{l=x,y,w,h} (h_i(l) - MAP(l))^2 * P(h_i) \quad (4)$$

$$UD = \mathbf{s}_x^2 + \mathbf{s}_y^2 + \mathbf{s}_w^2 + \mathbf{s}_h^2 \quad (5)$$

Where $h_i(l)$ represents the value of the hypothesis h_i in the l dimension, and \mathbf{s}_l^2 represents the variance in the l dimension calculated with respect to the MAP estimator. So the UD metric is equivalent to the trace of the covariance matrix or sum of its eigenvalues [9].

The second metric is intended to evaluate the quality of the information provided by each agent in terms of uncertainty reduction. The intuition is if an agent is providing good information its local likelihood should be close to the state-pdf maintained by the abstraction node. So the problem reduces to quantify similarity between distributions. In this work we compare distributions using the Kullback-Leibler divergence [10], which is given by equation (6).

$$D(f, g) = \sum_i f(i) * \log \frac{f(i)}{g(i)} \quad (6)$$

$f(i)$ = pdf for the state estimation

$g(i)$ = local normalized agent likelihood

Using the previous performance metrics we introduce two adaptation schemes to the state estimation. The first scheme is performed by the abstraction node. Using the UD metric the abstraction node measures the level of ambiguity in its state representation. If this level exceeds a predefined threshold the central inference sends an activation signal to any inactive agent to start sending supporting evidence that can eventually reduce the current ambiguities. Also, in case that this level is lower than a predefined threshold, meaning that the uncertainty is low, the abstraction node stops the less informative agent in order to increase the efficiency of the state estimation. The selection of the less informative agent is performed in base to the relative values of the Kullback-Leibler divergence among the active agents.

The second adaptation scheme is carry out locally by each agent using the UD metric. In this case, given that each agent calculates a likelihood function, the MAP is replaced in equation (3) by the maximum likelihood hypothesis (ML). Using this metric each agent evaluates the local level of uncertainty in its information sources. If this level exceeds a predefined threshold the agent modify its own local actions in order to improve its performance. In case that after a number of cycles the agent still cannot improve its performance, it stops processing information becoming an inactive agent.

Section 4 describes an implementation of these adaptation mechanisms for the case of visual algorithms based on color, depth, and shape visual cues.

3. Related Work

The idea of reducing uncertainty by combining knowledge from difference sources is by not account new. In several fields it is possible to find studies that recognize the relevance of integrating information in order to create more robust and flexible systems. Although all the abundant literature, there have been a gap between the conceptual idea and the production of working systems for real problems. Important issues such as the organization and control of the pieces of knowledge, and in special the development of mechanisms that allow the adaptation and feedback among the knowledge sources have not been tackled in depth, and they are still open questions.

In the AI domain the blackboard model for problem solving is one of the first attempts to adaptively integrate different types of knowledge sources [11]. The blackboard conceptualization is closely related to the ideas presented in this work, but as a problem-solving scheme the blackboard model offers just a conceptual framework for formulating solutions to problems. This research aims to extent the blackboard conceptualization to a computational specification or working system, providing specific mechanisms to perform probabilistic inference and adaptive integration for the case of dynamic visual information.

In the computer vision literature, there have been an interest in the creation of systems that integrate information from different visual cues, but we are not aware of working systems that include mechanisms to adaptively select the most appropriate set of visual cues, or that incorporate mutual feedback between the individual visual algorithms. Among the works that have shown the gain in robustness of combining several visual modules, it is possible to mention [4][12]. Unfortunately, most of these works have not considered in their systems topics such as adaptation and general types of uncertainties, being the works by Isard and Blake [7], and Rasmussen and Hager [13] some of the notable exceptions.

4. Implementation

Bounding boxes were used to describe the state of a tracked target. These bounding boxes were modeled by their center position (x,y) , width (w) , and height (h) . The rectangular boxes in the right side of figure 6 are examples of hypotheses used to perform state estimation. In all the examples shown in this work the state-pdf was approximated using 1000 samples. Also for the dynamic of the system we used a stationary Gaussian model with sigma equal to 20 pixels.

4.1. Visual Agents

At this moment the implementation is based on tree visual agents: Color, Stereovision, and Shape agents. These agents are implemented using a cooperative distributed multi-agent architecture [14]. Each agent is an independent process that runs concurrently and can be started on as-needed basis in a remote or local machine.

4.2.1 Color Agent

The color agent uses as observation the hue histogram of the pixels inside each hypothesis in the sample set. Hue is calculated according to the relative influence of the dominant RGB component with respect to the other 2 components. This is achieved using a circular transformation, where the RGB components are equally spaced on the circle. Red at 0° , Green at 120° , and Blue at 240° . In this way, the hue value for an input point with dominant color Green is given by:

$$Hue(R,G,B) = 120 + \frac{B - R}{\max(R,G,B) - \min(R,G,B)} \quad (7)$$

Shifting equation (7) by 120° it is possible to obtain the hue for the cases of red and blue dominance. We prefer this heuristic transformation to the traditional RGB-HSI transformation based on a color pyramid, because it produces a smoother mapping.

In order to express its observations in terms of a likelihood function, the color agent calculates the similarity between each hypothesis and a reference histogram. A detailed description of this process can be found in [15].

A special agent called a detector selects the reference histogram automatically. The idea here is to divide the agents in two types: detectors and specialists. Detector agents are general tools able to initialize the tracking of a target. Detectors explore the incoming information looking for possible structures without relying in specific illumination, views, posture, etc. Once possible candidate structures have been identified, it is possible to use specialist agents. These agents are more specific tools able to look for specific supporting evidence in order to provide more efficient and robust appearance models. More details can be found in [15].

4.2.2. Stereovision Agent

The stereovision agent uses as observation the depth values of the pixels inside each hypothesis in the sample set. The algorithm is based on depth segmentation and blob analysis, a detailed description of the algorithm can be found in [16].

To express its observations in terms of a likelihood function, the stereovision agent estimates four properties of the depth values of each hypothesis: depth variance, heights of the depth blobs, blob shape as a ratio between the width and height of the blob, and number of points with valid depth values. Using these depth features the stereovision agent estimate a likelihood function using a multivariate Gaussian pdf given by equation (12):

$$\text{Likelihood}(\mathbf{J}) = \frac{\exp\{-0.5 * (\mathbf{J} - \mathbf{m})^T * \mathbf{B}^{-1} * (\mathbf{J} - \mathbf{m})\}}{(2\mathbf{p})^{1/2} |\mathbf{B}|^{1/2}} \quad (12)$$

\mathbf{J} = feature vector (depth, height, blob shape, valid points)

μ = feature vector for the reference target

\mathbf{B} = diagonal covariance matrix estimated by previous examples

In the case of depth information there is also a detector agent. This agent initializes candidate targets using the results of the depth based segmentation algorithm. Also the reference feature vector is updated using expressions similar to equations (10) and (11).

4.2.3. Shape Agent

The shape agent is based on a novel algorithm that uses as observation the silhouette of the targets. A set of training examples is used to generate a shape model. The training examples consist of a set of shape vectors. Figure 3 shows the different steps involved in the calculation of each shape vector. The input to the process is a gray level image consisting of a bounding box containing a target, as shown in figure 3a. A canny edge detector is applied in this area obtaining an edge image, as shown in figure 3b. Starting from a configuration similar to the one presented in figure 3c, a snake curve is fit to the edge information, as shown in figure 3d. Using the resulting snake points a uniform B-spline curve [17] is fit to the target boundary using $N=40$ control points. After scaling these control points to a uniform size, they are used to generate a continuous close contour around the silhouette of the target, as shown in figure 3e. Finally, this close contour is used to generate the shape vector, which consists of the signed distance function [18] calculated from a regular grid of points inside the input image. Figure 3f shows a gray level image of the signed distance function obtained for the example, each point encodes the distance to the nearest point on the boundary with negative values inside the boundary.

The training examples are used to build a point distribution model using principal component analysis. After the PCA decomposition only the eigenvectors corresponding to the largest eigenvalues are considered to build the shape space.

This corresponds to the eigenvalues that accounts for more than 95% of the shape variations. In this way, the resulting model consists of a mean shape and an orthonormal shape space built with the more significant eigenvectors.

To express new observations in terms of a likelihood function, each hypothesis is transformed into a shape vector using the process described in figure 3. After subtracting the mean shape, the resulting vector is projected to the shape space obtaining a vector S . Assuming a Gaussian distribution of shape the likelihood of the shape vector is calculated by:

$$P(S) = \frac{1}{\sqrt{(2\pi)^k |\Sigma|}} \exp\left(-\frac{1}{2} * (S^T \Sigma^{-1} S)\right) \quad (13)$$

Where Σ is a $k \times k$ diagonal matrix containing the more significant eigenvalues.

It is important to note that the use of a signed distance function to build the shape vectors allows a greater tolerance to slight misalignment with respect to the shape model. Although in this work the shape model was learned offline, it is possible to think in a case where the model is automatically learned and updated online using as training examples the tracking results of other visual agents.

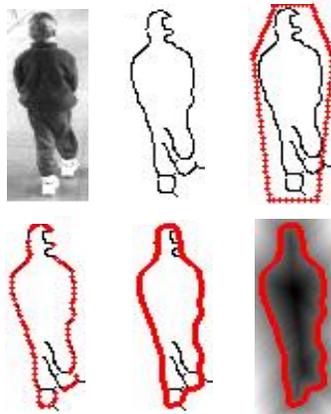


Fig. 3. a)Input image, b)Edge image after applying canny edge detector, c)Initial configuration to run snake algorithm, d)Resulting snake fitting the contour, e) Uniform B-Spline fitting the contour, f)Shape vector given by signed distance

5. Results

A version of the system proposed in this work has been developed for the case of person tracking. Figure 4 shows three frames of a video sequence used to evaluate the system (for more results check [15]). In this sequence the system tracks 2 targets that move without overlap.



Fig. 4. Three frames of a video sequence used for target tracking

After an initial detection given by the stereovision agent detector, the system starts the tracking of each target using the color, stereovision, and shape agents. After 5 frames the system decides to track the left side target using just the simpler color agent. In the case of the right side target, its proximity to a large bright window and limitations in the dynamic range of the color CCD make the color information unreliable, and the system decides to track this target using the stereovision and shape agents. In this case, as it is shown in figure 5, the tracking using just stereovision is not totally reliable because the stereovision segmentation tends to link the target to the glass wall. In the same way, the tracking based just on shape tends to miss parts of the silhouette when the image becomes too bright. As it is shown on figure 6, a combination of the two cues allows performing a proper tracking for this target.



Fig. 5. Result of stereovision segmentation. The close distance between the right side target and the glass wall confuses the stereovision agent

Figure 7 shows the evolution of the UD index for the case of the left target. The peak in the curve around frame 62 is due to the increase in the lighting coming from the left window, which produces an abrupt change in the color appearance mainly due to limitations in the dynamic range of the CCD camera. In this case the system activates the stereovision agent reducing the uncertainty. This shows how adaptation allows the system to operate successfully even when the assumptions of some of the algorithms, in this case color constancy, are not applicable at all times.

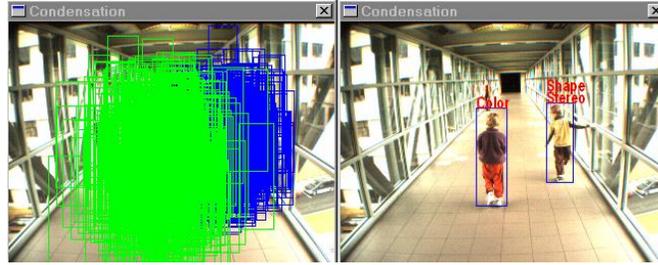


Fig. 6. (Left) hypotheses for state estimation at frame 7. (Right) MAP hypotheses frame 7

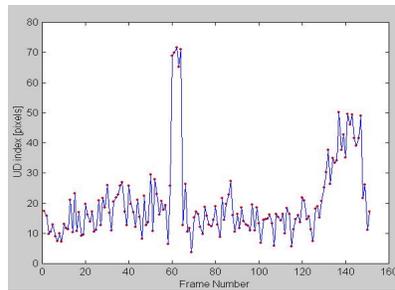


Fig. 7. Evolution of the UD index at the abstraction node for the left side target

6. Conclusions

This paper presented a new approach for the creation of an adaptive visual system. Using an intelligent agent paradigm in combination with a probabilistic approach and uncertainty metrics, we developed a sound methodology to adaptively combine dynamic visual information.

Taking into account the level of uncertainty in the information provided by the agents and the uncertainty in the state estimation, the system was able to take appropriate decisions about the more suitable use of the incoming information.

The implementation of the system for the case of human tracking showed encouraging results. The comparative analysis with respect to the case of operation without adaptation and/or integration shows that the adaptive integration of information increases the robustness and efficiency of the system in terms of accuracy and output rate.

There are still further research avenues to improve the system. At this moment we are currently adding more visual cues. We are also adding alternatives algorithms for the case of color and depth cues that differ in terms of assumptions and complexity, so the system can select the most appropriate. In the case of target tracking we are adding reasoning schemes for the case of target occlusion.

We believe that a synergistic combination of elements from computer vision, intelligent agents technology, probabilistic reasoning, and information theory is a viable way for the creation of *flexible, robust* and *efficient* vision system for tasks like target tracking.

References

1. Zeki S (1993) A vision of the brain. Oxford, Blackwell scientific publications
2. Nourbakhsh I, Bobenage J, Grange S, Lutz R, Meyer R, Soto A (1999) An affective mobile educator with a full-time Job. Artificial Intelligence, 114 (1-2), pp 95-124
3. Kanade T, Collins R, Lipton A, Burt P, Wixson L (1998) Advances in cooperative multisensor video surveillance. Darpa Image Understanding Workshop, Morgan Kaufmann, pp 3-24
4. Darrell T, Gordon G, Harville M, Woodfill J (1998) Integrated person tracking using stereo, color, and pattern detection. Proc of the conf on computer vision and pattern recognition, pp 601-609
5. Jennings N, Wooldridge M (1995) Applying agent technology. J. of Applied Artificial Intelligence, special issue on Intelligent Agents and Multi-Agent Systems
6. Gordon N, Salmon D, Smith A (1993) A novel approach to nonlinear/non Gaussian Bayesian state estimation. IEE Proc on Radar and Signal Processing, vol 140, pp 107-113
7. Isard M, Blake A (1996) Visual tracking by stochastic propagation of conditional density. Proc of 4th European conf on computer vision, pp 343-356
8. Kitagawa G (1996) Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. J of Computational and Graphical Statistics, vol 5, pp 1-25
9. Fukunaga K (1972) Introduction to statistical pattern recognition. Academic Press
10. Cover T, Thomas J (1991) Elements of information theory. Wiley series in telecommunications
11. Reddy R, Erman D, Richard N (1973) A model and a system for machine recognition of speech. IEEE transactions on audio and electroacoustic AU-21, pp 229-238
12. Krotkov E, Bajcsy R (1993) Active vision for reliable ranging: cooperating focus, stereo, and vergence. Intl J of computer vision, vol 11, n 2, pp 187-203
13. Rasmussen C, Hager G (1998) Joint probabilistic techniques for tracking multi-part objects. Proc of the conf on computer vision and pattern recognition, Santa Barbara
14. Dolan J, Trebi-Ollennu A, Soto A, Khosla P (1999) Distributed tactical surveillance with ATVs. Proc of SPIE, vol 3693, AeroSense, Orlando, Fl
15. Soto A, Khosla P (2002) Adaptive agent based system for state estimation using dynamic multidimensional information sources. Lectures notes in computer science, in print.
16. Soto A, Sapharishi M, Dolan J, Trebi-Ollennu A, Khosla P (1999) CyberATVs: dynamic and distributed reconnaissance and surveillance using all terrain UGVs. Proc of the Intl conf on field and service robotics, August 29-31
17. Bartels R, Beatty J, Barsky B (1987) An introduction to splines for use in computer graphics and geometric modeling. Morgan Kaufmann
18. Sethian J (1996) Level set methods. Cambridge University Press