

A Probabilistic Approach for Dynamic State Estimation Using Visual Information

Alvaro Soto¹ and Pradeep Khosla²

¹ Pontificia Universidad Catolica de Chile
Santiago 22, Chile
asoto@ing.puc.cl

² Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, Pa, 15213, USA
pkk@ece.cmu.edu

Abstract. This work presents a computational framework for the adaptive integration of information from different visual algorithms. The approach takes advantage of the richness of visual information by adaptively considering a variety of visual properties such as color, depth, motion, and shape. Using a probabilistic approach and uncertainty metrics, the resulting framework makes appropriate decisions about the most relevant visual attributes to consider. The framework is based on an agent paradigm. Each visual algorithm is implemented as an agent that adapts its behavior according to uncertainty considerations. These agents act as a group of experts, where each agent has a specific knowledge area. Cooperation among the agents is given by a probabilistic scheme that uses Bayesian inference to integrate the evidential information provided by them. To deal with the inherent non-linearity of visual information, the relevant probability distributions are represented using a stochastic sampling approach. The estimation of the state of relevant visual structures is performed using an enhanced version of the particle filter algorithm. This enhanced version includes novel methods to adaptively select the number of samples used by the filter, and to adaptively find a suitable function to propagate the samples. We show the advantages of our approach by applying it to the task of tracking targets in a real video sequence.

1 Introduction

In the robotics domain the problem of understanding sensing information from the environment is highly relevant. In particular, visual perception is a very attractive option to equip a robot with suitable perceptual capabilities. The robustness and flexibility exhibited by most seeing beings is a clear proof of the advantages of a sophisticated visual system. Evolution has managed to provide biological creatures with the timely visual perception needed to interact with a dynamic world.

Insects provide one of the most basic examples of a robust visual system able to successfully operate in a natural environment. Insects are able to navigate

around a 3D cluttered world, avoiding obstacles, recognizing paths, landing in many places, taking off to avoid capture, and so on. Several studies [1] show that insects, such as bees, move around their environment using a set of visual skills based on simple visual cues such as color, position, orientation, and relative change of size from different points of view. They can find food using the salient color of flowers or they can find their way home using the visual changing size of relevant natural landmarks.

In the case of the human visual system, people usually characterize objects using distinctive visual features [11]. For example, most people are comfortable with instructions such as “Go straight until you see the tall round building, then turn right and go until you see the red sign...” .

The previous examples show the richness of visual information. In contrast to other sensor modalities, vision provides information about a large number of different features of the environment such as color, shape, depth, or motion. This multidimensionality provides strong discriminative capabilities to eliminate ambiguities being one of the key strength that explains the great robustness observed in most advanced biological visual systems.

Unfortunately, the state of the art of artificial visual perception is still far behind that of its biological analogue. Currently, most successful applications of machine vision demonstrate a robust operation only under constrained conditions. The typical approach relies on simplifications of the environment or on good engineering work to identify relevant visual attributes to solve a specific visual task. As an example, consider the case of a robot localization system based on artificial landmarks. In this case, the use of distinctive visual features in the landmarks provides strong constraints to construct algorithms especially designed to detect the key visual attributes [8]. In the same way, recent successful vision systems to detect people or cars are examples of specific visual applications where good engineering work provides an off-line identification of key visual attributes [2, 6, 7].

The previous strategy lacks the flexibility needed to account for the variability of most natural scenarios. Problems such as partial occlusion, changes in illumination, or different postures constantly modify the amount of information of the different visual attributes. As a consequence, the most adequate set of attributes to complete a given task is highly variable.

This paper presents a computational framework for the adaptive integration of information from different visual algorithms. The basic scenario is a mobile robot embedded in a dynamic environment and processing visual information. As new information arrives, the goal is to use the most adequate set of information sources in order to update the knowledge about relevant visual structures. The central idea is to deal with the ambiguity and dynamics of natural environments by adapting the operation of the system. The adaptive integration of visual information aims to achieve two desirable attributes of an engineering system: *robustness* and *efficiency*. By combining the outputs of multiple vision modules the assumptions and constraints of each module can be factored out to result in a more robust system overall. Efficiency can be kept through the on-line selection

and specialization of the algorithms according to the relevant conditions present at each time in the visual scene.

The computational framework is based on an intelligent agent paradigm. Each visual algorithm is implemented as an agent that adapts its behavior according to uncertainty considerations. These agents act as a group of experts, where each agent has a specific knowledge area. Cooperation among the agents is given by a probabilistic scheme that uses Bayesian inference to integrate the evidential information provided by them. To deal with the inherent non-linearity of visual information, the relevant probability distributions are represented using a stochastic sampling approach. The estimation of the state of relevant visual structures is performed using an enhanced version of the particle filter algorithm. This enhanced version includes novel methods to adaptively select the number of samples used by the filter, and to adaptively find a suitable function to propagate the samples.

This paper is organized as follows section 2 presents the main components of the framework. Section 3 describes the adaptation mechanisms used to select the visual agents. Section 4 discusses the details of our probabilistic approach. Section 5 shows the results of applying our methodology to tracking targets in a real video sequence. Finally, Section 6 presents the main conclusions of this work.

2 A Multimodal Probabilistic Agent Based Approach

The goal of the approach presented here is to keep track of a joint probability density function (joint-pdf) over a set of state variables that characterizes the state of the world. For example, for the case of visual tracking of a single target, the state of the system is represented by four 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. Each instance of the state corresponds to a hypothesis of the possible position of a target.

The level of uncertainty in the state estimation is the key element used by the system to implement the adaptation mechanisms that select the most adequate visual agent. For example, if the system is tracking a target using color information but similarities with respect to the color of the background or other target produce a high level of uncertainty, the system automatically activates new visual modalities such as texture or stereovision to reduce the current ambiguities.

Figure 1 shows an example of the basic probabilistic architecture used in this work. This can be considered as a Bayes net. Agent nodes correspond to the visual algorithms that directly process the incoming images. Inference nodes keep track of the state estimation represented by a set of sample hypotheses and their probabilities. Inference nodes provide the integration of information and the representation of relevant visual structures. Also, inference nodes introduce conditional independence relations among the visual algorithms. This decoupling of information facilitates the construction of probabilistic models for applying

Bayesian inference and provides a high degree of modularity to reconfigure the system.

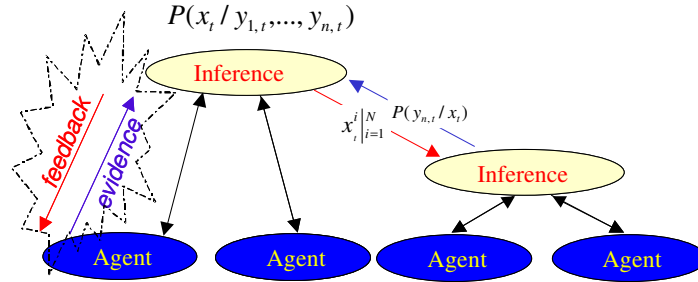


Fig. 1. Typical Bayes net and information flow for the probabilistic architecture used in this work.

A practical difficulty of using Bayesian inference is to find adequate pdfs. The problem is even more difficult in the dynamic case, where one also needs to find a time propagation model. This work uses a sequential Monte Carlo technique commonly known as particle filter as the basic stochastic sampling algorithm to perform Bayesian inference. As opposed to alternative techniques such as the Kalman Filter, the particle filter is able to approximate any functional non-linearity of the pdfs. The key idea is to represent a posterior distribution by samples (particles) that are continually re-allocated after each new estimation of the state.

Figure 1 can be considered as a hierarchical representation of the simpler case of just one inference node. In this case the posterior distribution of the estimation of the state can be expressed as:

$$P(x_t / \mathbf{y}_{1,t} \dots \mathbf{y}_{n,t}) = \beta P(y_{1,t} / x_t) \dots P(y_{n,t} / x_t) \sum_{x_{t-1}} P(x_t / x_{t-1}) P(x_{t-1} / (\mathbf{y}_{1,t-1} \dots \mathbf{y}_{n,t-1})) \quad (1)$$

where x_t corresponds to a hypothesis of the state of the system at time t ; $\mathbf{y}_{i,t}$ contains the historic evidence provided by agent i until time t ; n corresponds to the number of independent sources of information; and β is a normalization factor.

Equation (1) shows the decoupling between the evidence provided by each agent through a likelihood function, and the state updating performed by an inference node. The process is as follows. Agent nodes provide evidential information to an inference node. Using this information the inference node keeps track of a probability density function (pdf) that represents its beliefs about the state of the system. According to these beliefs, the inference node sends high-level feedback to its active descending agent nodes. This feedback consists of a list of hypotheses that need further consideration and a signal to the agents

that need to be active. The idea is that, according to the level of ambiguity, the inference node decides which of its potential descending nodes to activate, and which hypotheses are worth being considered. The agent nodes use this feedback to efficiently allocate their resources and as a reference about the quality of their own performance. The active agents evaluate the set of hypotheses and send this evidence back to the inference node starting a new cycle of the adaptive state estimation. Eventually, an inference node can send feedback to a descending node that is also an inference node. This can happen in problems that require higher levels of inference, such as the case of considering occlusion in a multiple target tracking application.

3 Adaptation

The adaptation mechanisms are based on the evaluation of the level of uncertainty present in the state estimate and the evaluation of the quality of the information provided by each agent in terms of uncertainty reduction. The goals are to perform robust estimation keeping uncertainty low, and also to perform efficient estimation avoiding the processing of irrelevant, misleading, or redundant information. In order to achieve these goals, we introduce two performance indexes.

The first index, called uncertainty deviation (UD), is intended to evaluate the level of ambiguity in the state representation.

$$UD = \sum_{x_i} d^2(x_i, MAP) P(x_i) \quad (2)$$

where d denotes a distance metric and x_i denotes a generic hypothesis of the state of the system. In this work we use Euclidean distance and state variables (x, y, w, h) .

The second index is intended to evaluate the quality of the information provided by each agent in terms of uncertainty reduction. The intuition behind it is that, if an agent is providing good information, its local likelihood should be close to the state-pdf maintained by the inference node. Thus, the problem reduces to quantifying similarity between probability distributions. This work compares probability distributions using the Kullback-Leibler divergence (KL-divergence). The KL-divergence between two probability distributions is given by Equation (3),

$$KL(f, g) = \sum_i f(i) \ln\left(\frac{f(i)}{g(i)}\right). \quad (3)$$

In this case, $f(i)$ corresponds to the pdf of the state and $g(i)$ corresponds to the local normalized agent likelihood.

Using the described performance indexes, this work introduces two adaptation schemes to the state estimation. The first scheme is performed by inference nodes. An inference node measures the level of ambiguity in its state representation using the UD index. If the value of the index exceeds some desired level, the

inference node sends an activation signal to any inactive agent asking for supporting evidence that can eventually reduce the current ambiguities. If the value of the UD index is lower than some desired level, the inference 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 based on the relative values of the KL-divergence among the active agents.

The second adaptation scheme is carried out locally by each agent using the UD index. In this case, given that each agent calculates a likelihood function, the MAP in Equation (2) is replaced by the maximum likelihood hypothesis (ML). Using this index each agent evaluates the local level of uncertainty in its estimation. If this value exceeds some desired level, the agent modifies its own local actions in order to improve its performance. In the case that, after a number of cycles, the agent is still not able to improve its performance, the agent stops processing information, becoming inactive.

4 Probabilistic Inference

Besides the adaptive allocation of prominent hypotheses, the standard formulation of a particle filter lacks of further tools that take into account the varying complexity of the shape and dynamics of the hypothesis space. As the state of a system evolves the standard implementation of the particle filter keeps a fixed number of samples and a fixed propagation function to track the evolution of the state. The next two sections present mechanisms to overcome these limitations. Section 4.1 presents a method that can be used to adaptively estimate a suitable size for the sample set used at each iteration of the particle filter. Section 4.2 presents a method to improve the allocation of samples by updating the importance function using the most recent observation.

4.1 Adaptive Selection of the Number of Particles

The regular implementation of the particle filter uses a fixed number of samples. In most situations this is highly inefficient. The dynamics of most processes usually produces great variability in the complexity of the posterior distribution. As a consequence, the initial estimation of the number of particles can be much larger than the real number needed to perform a good estimation or, even worse, at some point the number can be too low causing the filter to diverge. This section shows a technique that can be used to adaptively estimate a suitable number of particles that, with a certain level of confidence, limits the maximum error in the approximation.

At NIPS'01, Fox introduces KLD-Sampling [4], a novel technique to adaptively estimate the number of particles that bound the error in the estimation of the true posterior. The error is measured by the Kullback-Leibler divergence (KL-divergence) between the true posterior distribution and the empirical distribution, which is a well known nonparametric maximum likelihood estimate [9]. The method is based on the assumption that the true posterior distribution

can be represented by a discrete piecewise constant distribution consisting of a set of multidimensional bins. Using this representation, Fox finds a relation connecting the likelihood ratio to the KL-divergence.

The problem with KLD-Sampling is the derivation of the bound using the empirical distribution, which has the implicit assumption that the samples come from the true distribution. This is not the case for particle filters. In particle filters the samples come from an importance function, moreover, the quality of the match between this function and the true distribution is one of the main elements that determines the accuracy of the filter, hence a suitable number of particles.

To fix the problem of KLD-Sampling one needs a way to quantify the degradation in the estimation using samples from the importance function. The goal is to find the equivalent number of samples from the importance and the true densities that captures the same amount of information about $p(x)$. In the context of MC integration, Geweke [5] introduces the relative numerical efficiency (RNE), which provides an index to quantify the influence of sampling from an importance function. The idea behind RNE is to compare the relative accuracy of solving an integral using samples coming from either the true or the importance function. Accuracy is measured according to the variance of the estimator of the integral.

In the case that the integral corresponds to the estimation of the mean value of the state, the variance of the estimator corresponds to the variance of MC integration, which is given by [3]:

$$\text{Var}[E_{MC}^N(x)] = \text{Var}_p(x)/N \quad (4)$$

where N is the number of samples coming from the true distribution ³.

Now, in the case where the samples come from an importance function $q(x)$, the variance of the estimator corresponds to the variance of importance sampling, which is given by [5]:

$$\text{Var}[E_{IS}^N(x)] = E_q((x - E_p(x))^2 w(x)^2)/N_{IS} = \sigma_{IS}^2/N_{IS}, \quad (5)$$

where $w(x) = p(x)/q(x)$ corresponds to the weights of importance sampling and N_{IS} is the number of samples coming from the importance function.

To achieve similar levels of accuracy, the variance of both estimators should be equal. This relation provides a way to quantify the equivalence between samples from the true and the proposal density:

$$N = \frac{N_{IS} \text{Var}_p(x)}{\sigma_{IS}^2} \quad (6)$$

Replacing (6) in the bound given by KLD-Sampling, it is possible to obtain (7) which takes into account the case that the samples do not come from the true distribution but from an importance function (See [10] for details):

³ The rest of this section concentrates on one iteration of the particle filter and the t subscripts are dropped.

$$N_{IS} > \frac{\sigma_{IS}^2}{\text{Var}_p(x)} \frac{1}{2\epsilon} \chi_{k-1, 1-\delta}^2. \quad (7)$$

4.2 ADAPTIVE PROPAGATION OF THE SAMPLES

The regular implementation of the particle filter uses the dynamic prior $p(x_t/\mathbf{y}_{t-1})$ [3] as the importance function to obtain the set of hypotheses that are sent to the agent nodes. This has the advantage of allowing a computational complexity of $O(N)$ in the operation of the filter, where N is the number of hypotheses. However, the use of this importance function has the limitation of allocating the samples without considering the most recent observation y_t . This section shows a new algorithm that improves this situation by incorporating the current observation in the generation of the samples, and also keeping the computational complexity of $O(N)$.

Consider the following expression for the dynamic prior:

$$p(x_t/\mathbf{y}_{t-1}) = \int p(x_t/x_{t-1}) p(x_{t-1}/\mathbf{y}_{t-1}) dx_{t-1}. \quad (8)$$

Using the method of composition the samples from the dynamic prior are obtained by sampling from $p(x_{t-1}/\mathbf{y}_{t-1})$ and then propagating each of these samples x_{t-1}^k by $p(x_t/x_{t-1}^k)$. From an MC perspective, it is possible to achieve a more efficient allocation of the samples by including y_t in the generation of the samples that are propagated by $p(x_t/x_{t-1}^k)$. The intuition is that the incorporation of y_t increases the number of samples drawn from mixture components $p(x_t/x_{t-1}^k)$ associated with areas of high probability under the likelihood function.

Under importance sampling it is possible to sample from $p(x_{t-1}/\mathbf{y}_t)$ instead of $p(x_{t-1}/\mathbf{y}_{t-1})$ and then adding to each particle x_{t-1}^i a correcting weight given by,

$$w_t^i = \frac{p(x_{t-1}^k/\mathbf{y}_{t-1})}{p(x_{t-1}^k/\mathbf{y}_t)}, \quad \text{with } x_{t-1}^i \sim p(x_t/x_{t-1}^k) \quad (9)$$

The resulting set of weighted samples $\{x_t^i, w_t^i\}_{i=1}^n$ still comes from the dynamic prior, so the computational complexity of the resulting filter is still $O(N)$. The extra complexity of this operation comes from the need to evaluate and to draw samples from the importance function $p(x_{t-1}^i/\mathbf{y}_t)$. Fortunately, the calculation of this function can be obtained directly from the operation of the regular particle filter. To see this clearly, consider the following:

$$\begin{aligned} p(x_t, x_{t-1}/\mathbf{y}_t) &\propto p(y_t/x_t, x_{t-1}, \mathbf{y}_{t-1}) p(x_t, x_{t-1}/\mathbf{y}_{t-1}) \\ &\propto p(y_t/x_t) p(x_t/x_{t-1}, \mathbf{y}_{t-1}) \\ &\quad \times p(x_{t-1}/\mathbf{y}_{t-1}) \\ &\propto p(y_t/x_t) p(x_t/x_{t-1}) \\ &\quad \times p(x_{t-1}/\mathbf{y}_{t-1}) \end{aligned} \quad (10)$$

Equation (10) shows that, indeed, the regular steps of the particle filter generate an approximation of the joint density $p(x_t, x_{t-1}/\mathbf{y}_t)$. After re-sampling from $p(x_{t-1}/\mathbf{y}_{t-1})$, propagating these samples with $p(x_t/x_{t-1})$, and calculating the weights $p(y_t/x_t)$, the set of resulting sample pairs (x_t^i, x_{t-1}^i) with correcting weights $p(y_t/x_t^i)$ forms a valid set of samples from the joint density $p(x_t, x_{t-1}/\mathbf{y}_t)$. Considering that $p(x_{t-1}/\mathbf{y}_t)$ is just a marginal of this joint distribution, the set of weighted-samples x_{t-1}^i are valid samples from it.

5 APPLICATION

This section illustrates the main features of the work presented here, when used to track targets in real video sequences. The main focus is to show the operation of the different adaptation mechanisms introduced for the activation of visual algorithms, the selection of the number of particles, and the propagation function used by the particle filter. See [10] for further examples and comparison with alternative existing techniques. The system operates using visual agents based on color and stereo vision.

Figure 2 shows a set of frames of the video sequence used to test our approach. This sequence consists of two children playing with a ball. In this case, the goal is to keep track of the positions of the ball and the left side child.

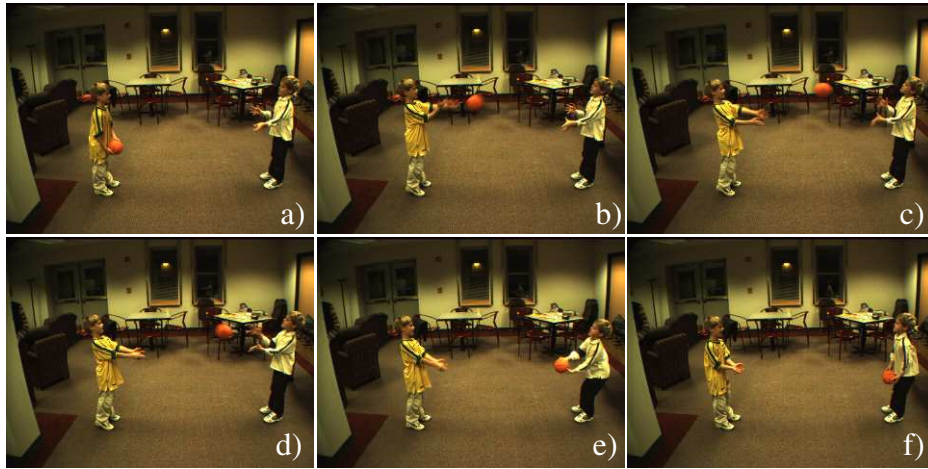


Fig. 2. a) Frame 1. b) Frame 4. c) Frame 5. d) Frame 6. e) Frame 7. f) Frame 14. A set of representative frames of the video sequence used to illustrate the advantages of the approach presented in this work.

In this application each hypothesis about the position of a target is given by a bounding box defined by height, width, and the coordinates of its center. The motion model used for the implementation of the particle filter corresponds to a Gaussian function of zero mean and known diagonal covariance matrix with standard deviations set to 20 for the center of each hypothesis and to 0.5 for its width and height.

The first test considers tracking using visual algorithms based on color and stereovision. In this case, the particle filter used to estimate the pdf of the state of each target considers an adaptive number of particles, but a fixed importance function to generate the samples. Figure 3 shows the tracking results at different time instants. The left figures show all the hypotheses used to estimate the state of each target, while the right side figure shows only the hypothesis with the highest probability (MAP hypothesis).

The system starts the tracking of both targets using the color and stereovision agents. After three frames, the system automatically decides to track the child using only the information provided by the stereovision agent and the ball using only the information provided by the color agent. In the case of the child, lack of hue information in his pants and similarities between the hue of his t-shirt and the carpet make the color information unreliable, and the system decides to use the clear depth features provided by the stereovision agent. In the case of the ball its small size and abrupt motion produce a very poor information from the stereovision agent, and the system decides to use the clear orange color to perform the tracking.

Figure 4 shows the number of particles used at each frame to estimate the posterior distribution of the targets. The number of particles is obtained using a desired error of 0.1 and a confidence level of 95%. In the calculation of the number of particles only the x and y dimensions are considered, assuming independence to facilitate the use of Equation (7). At each iteration, the number of particles selected to run the filter corresponds to the greater value between the estimates obtained in the x and y dimensions. Given that the results in Equation (7) are asymptotic, in practice a minimum number of 1000 samples is always used to ensure that convergence has been achieved.

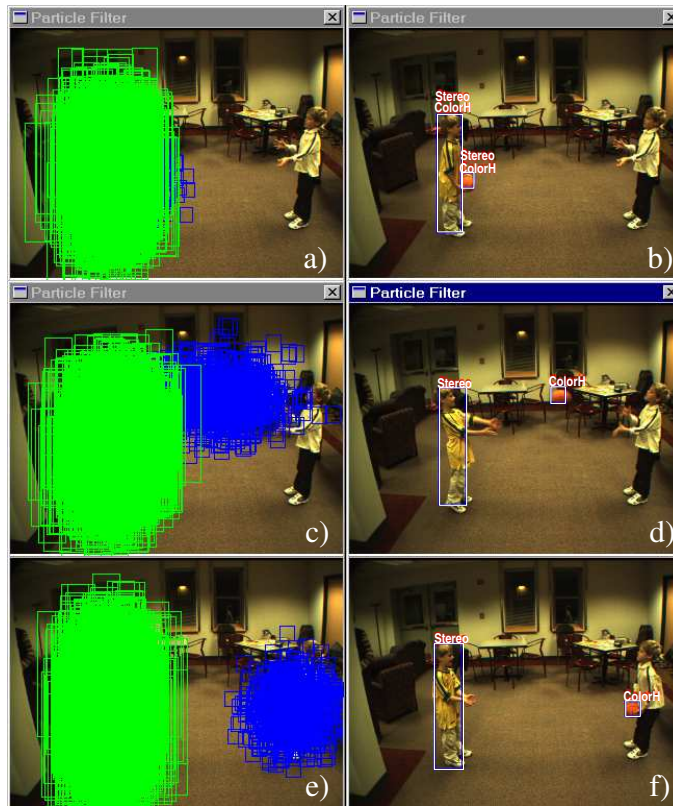


Fig. 3. a-b) Frame 1. c-d) Frame 5. e-f) Frame 14. a-c-e) Hypotheses representing the posterior distribution of each state at different time instants. b-d-f) The MAP hypothesis is marked with a bounding box.

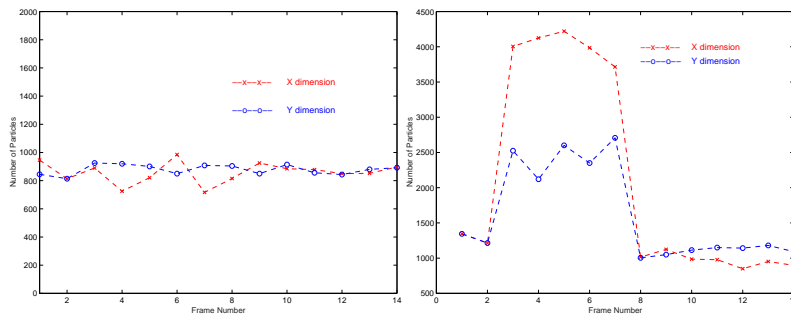


Fig. 4. Number of particles at each iteration of the particle filter using Equation (7) with desired error of 0.1 and a confidence level of 95%. Left: Left side child. Right: Orange Ball

As expected, in the case of the child, the number of particles is roughly constant during the entire sequence. In this case the motion model is highly accurate because the child has only a small and slow motion around a center position. Then, the performance is similar to the case of using a regular particle filter with a constant number of particles. In the case of the ball the motion model is a poor approximation of the real motion, because the ball has a large and fast motion traveling from one child to the other. Then, the number of particles needed to achieve the desired error level has a large increment during this period (Frames 3 to 7). Also, during this period the mismatch between the dynamic prior and the posterior produces an inefficient allocation of the samples. This is clear in Figure 5, which shows the resulting posterior distribution at different time instants. For clarity only the (x, y) coordinates of the center of each hypothesis are shown in the graphs.

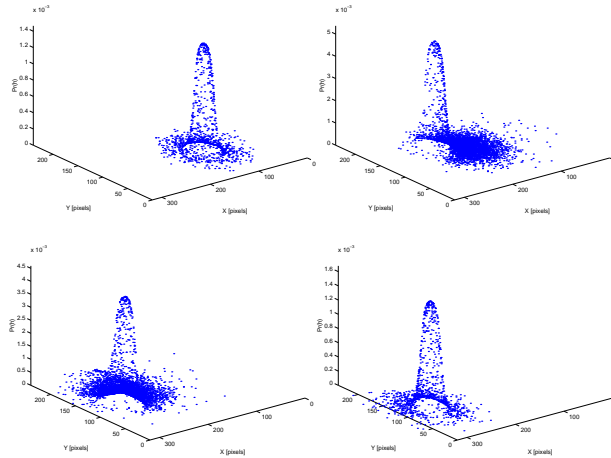


Fig. 5. Estimate of the posterior distribution of the position of the center of the ball at different time instants. From top to bottom: Frames 1, 5, 7, and 12.

In Figure 5, at Frame 1 the ball is mostly static in the hands of the left-side child. As a consequence, the motion model is adequate, and most of the samples from the important function are allocated in areas of high likelihood. In contrast, Figures 5(b), (c), and (d) show the situation when the ball travels from one child to the other. In this case, most of the samples are allocated in the tails of the posterior distribution, having a low likelihood. As a consequence, the estimate needs a larger set of samples to populate the relevant part of the posterior.

In the next test we add to the previous example the algorithm that enables the adaptation of the importance function used by the particle filter. Again, the estimate of the number of particles is performed using Equation (7). The decision to adapt the importance function is based on the estimate of the KL-divergence

between the dynamic prior and the posterior distribution (See [10] for details). Figure 6 shows the evolution of the KL-divergence calculated before adapting the importance function. The Figure shows how the value of the KL-divergence has a large increment during the period that the ball travels from one child to the other. Setting a value of 2 for the threshold on the KL-divergence, the system decides to

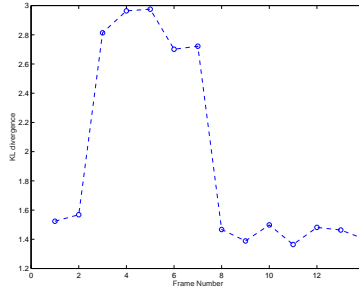


Fig. 6. KL-divergence between the dynamic prior and the posterior distribution.

adapt the importance function at all the frames where the ball travels from one child to the other (Frames 3-7). Figure 7 shows the location of the resulting set of samples used to estimate the posterior distribution. Comparing Figure 7 with Figure 5, it is possible to observe the gain in efficiency by a better allocation of the samples. The update of the importance function by using information from the current observation produces a re-allocation of the samples towards areas of high likelihood, reducing the number of samples needed to estimate the posterior distribution with a desired error level.

Figure 8 shows the number of particles predicted by Equation (7) to estimate the posterior distribution of the position of the ball. During Frames 3 to 7 it is possible to observe a reduction in the number of samples needed to achieve the desired level of accuracy in the estimate. This is the result of the compact representation of the estimate of the posterior that is influenced by accurate new observation given by a peaked likelihood function.

6 Conclusions

Using a synergistic combination of standard and innovative tools from computer vision, probabilistic reasoning, agent technology, and information theory, we presented a computational framework for the adaptive integration of visual information.

By taking into account the uncertainty in the information provided by a set of visual agents, the system was able to take appropriate decisions about the information sources that were worth processing. Furthermore, our enhanced version of the particle filter provided an adaptive mechanism to determine a

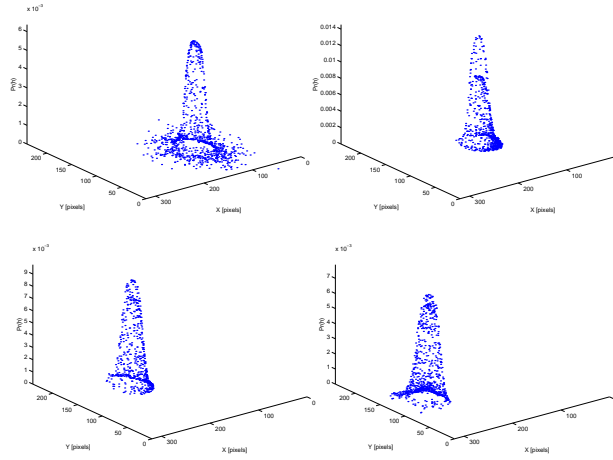


Fig. 7. Estimate of the posterior distribution of the position of the center of the ball at different time instants for the case of using an updated version of the dynamic prior as the importance function.

suitable number of samples and a suitable position for them. Given that the selection of the number of particles and the importance function are one of the main factors that determine the efficiency and accuracy of a particle filter, it is expected that the enhanced version presented in this work has an impact well beyond the application presented here.

The implementation of the system for the case of target tracking showed encouraging results. Using highly general visual algorithms and motion models, the system was able to simultaneously track targets with very different motions and visual features. In this test the application of the enhanced version of the particle filter produced a considerable reduction in the number of samples needed to estimate the state of the system.

References

1. T. Collett. Insect navigation en route to the goal: Multiple strategies for the use of landmarks. *Journal of Experimental Biology*, 235:199–227, 1996.
2. T. Darrell, G. Gordon, M. Harville, and J. Woodfill. Integrated person tracking using stereo, color, and pattern detection. In *IEEE Computer Vision and Pattern Recognition*, pages 601–609, Santa Barbara, June 1998.
3. A. Doucet, N. de Freitas, and N. Gordon. An introduction to sequential Monte Carlo methods. In *Sequential Monte Carlo Methods in Practice*, pages 3–14. Springer, 2001.
4. D. Fox. KLD-Sampling: Adaptive particle filters. In *Advances in Neural Information Processing Systems 14 (NIPS)*, 2001.
5. J. Geweke. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, 57:1317–1339, 1989.

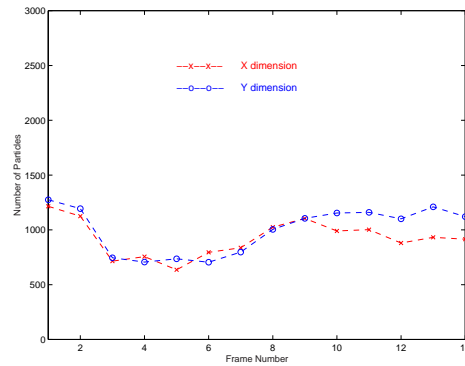


Fig. 8. Number of particles at each iteration of the enhanced version of the particle filter for tracking the orange ball. Between Frames 3 to 7 the importance function is updated including information from the last observation. The number of particles is adaptively selected using Equation (7). In this equation the error is set to 0.1 and the confidence level to 95%.

6. T Kanade, R. Collins, A. Lipton, P. Burt, and L. Wixson. Advances in cooperative multisensor video surveillance. In *Darpa Image Understanding Workshop*, pages 3–24. Morgan Kaufmann, 1998.
7. A. Lipton, H. Fujiyoshi, and R. Patil. Moving target classification and tracking from real-time video. In *Proceedings of the 1998 DARPA Image Understanding Workshop*, November 1998.
8. I. Nourbakhsh, J. Bobenage, S. Grange, R. Lutz, R. Meyer, and A. Soto. An affective mobile educator with a full-time job. *Artificial Intelligence*, 114(1-2):95–124, October 1999.
9. A. Owen. *Empirical Likelihood*. Monographs on Statistics and Applied Probability, Stanford University, California, USA, 2001.
10. A. Soto. A probabilistic approach for the adaptive integration of multiple visual cues using an agent framework. Tech report CMU-RI-TR-02-30, Carnegie Mellon University, 2002.
11. M. Zeki. *A Vision of the Brain*. Oxford, Blackwell scientific publications, 1993.