Bottom-up and top-down object matching using asynchronous agents and a contrario principles

Nicolas Burrus, Thierry Bernard, Jean-Michel Jolion

Bottom-up and top-down object matching using asynchronous agents and *a contrario* principles

Nicolas Burrus[1,2], Thierry M. Bernard[1], and Jean-Michel Jolion[2]

[1] ENSTA - UEI, 32 Boulevard Victor, 75015 Paris, France

[2] Université de Lyon, F-69361 Lyon
INSA de Lyon, F-69621, Villeurbanne
CNRS, LIRIS, UMR5205

Abstract. We experiment a vision architecture for object matching based on a hierarchy of independent agents running asynchronously in parallel. Agents communicate through bidirectional signals, enabling the mix of top-down and bottom-up influences. Following the so-called *a contrario* principle, each signal is given a strength according to the statistical relevance of its associated visual data. By handling most important signals first, the system focuses on most promising hypotheses and provides relevant results as soon as possible. Compared to an equivalent feed-forward and sequential algorithm, our architecture is shown capable of handling more visual data and thus reach higher detection rates in less time.

Keywords: object matching, top-down and bottom-up, *a contrario* reasoning, parallel vision

1 Introduction

We consider the task of 2d object matching: given a database of pictures of objects, we want to detect these objects in new images. Numerous methods have been proposed to solve this task, and most of them are either purely top-down or purely bottom-up. Top-down methods are generally based on template matching and their main limitation is their computational cost when object poses are not constrained in the image and when the database of objects is big. On the other hand, purely bottom-up methods are generally feature-based. Local invariant features are extracted from the image under analysis and matched to the features of database objects. Then concordant features are grouped to make object pose hypotheses. These approaches are much more efficient, but often have less discriminative power. Moreover, to save computation time, weak feature matches may need to be discarded at an early stage, and features actually belonging to an object can be wrongly ignored, thereby reducing detection rates. This motivates a joint use of top-down and bottom-up processing, as done in some recent works [1–4].

In order to break sequentiality in detection algorithms, we use parallelism between processing levels. Then, it becomes possible for strong hypotheses to reach high level analysis and give the first detection results early, much before the complete termination of low level processing. In addition to biological motivations [5], there are several practical interests. First, applications waiting for detection results can be triggered earlier, and the detection process can be constrained to run in a predefined time, and still return useful information (e.g. most contrasted and easy to discriminate objects). Second, since high level steps can be reached before the complete termination of lower level steps, they can influence low level processing at the light of current evidence and accelerate the detection process. Finally, more visual data can be analyzed and thus higher detection rates can be obtained without degrading too much the average detection delays, since promising hypotheses can be handled first.

Enabling a mixed use of top-down and bottom-up processing within a parallel architecture is a difficult algorithmic task. Some systems have already exploited these principles, for example at a very low level in [6, 7] or using complex scheduling strategies in [8, 9]. In this paper, we experiment an original object matching architecture that keeps things simple by using high level independent agents running in parallel without explicit synchronization nor scheduling.

2 Breaking the feed-forward and sequential model

Our starting point is the object matching approach presented in [10]. It will be referred as LBU (Lowe Bottom-Up) in the rest of this paper. It is feed-forward and bottom-up. First, and offline, all the SIFT points of the objects in the database are extracted from their 2d picture and stored. Then, when an image is given for analysis, the following algorithm is applied:

1. Extract the SIFT points of the image.

2. Match each SIFT point of the image to the closest point in the object database. Discard matches whose distance is too high.
3. For the remaining matches, thanks to the location, rotation and scale information embedded in SIFT points, each match results into a full object pose hypothesis. Matches leading to compatible poses are clustered with a Hough transform. Clusters with less than 3 matches are discarded.
4. For the remaining clusters, a probability of object presence is computed and then thresholded to make decisions.

Each step takes decisions using only partial information, and thus can discard relevant hypotheses. We observed that more than 30% of good matches may be lost in step 2 if the object database is quite large and contains rather similar objects. These early decisions also prevent the use of further top-down processing to discriminate weak hypotheses.

We propose to break this sequentiality by using a hierarchy of simple and independent agents. The proposed system is ruled by the following principles:

1. Parallelism. Agents run in parallel and asynchronously in that they run as soon as and as long as they have data to proceed. This allows high level processing to start without waiting for the complete termination of low level processing. For example, it becomes possible to reach step 4 of LBU for some salient hypotheses even if step 2 is not fully completed. To take an even greater advantage of hardware with several processing units, we also enable spatial parallelism by associating a receptive field to each agent.
2. Systematic evaluation of visual data relevance. To handle most promising hypotheses first, we evaluate visual data relevance not only at the final decision level, but also at lower levels when only partial information is available.
3. Bidirectional and asynchronous communications between agents. When an agent has processed its data, it creates a signal holding the result, and sends it to its parent (bottom-up) or to its children (top-down). Each agent keeps a buffer of signals to process so that agents never have to wait for each others. The signal system allows top-down and bottom-up influences to navigate freely through the hierarchy of agents. To ensure that signals containing strong evidence are handled first, principle 2 is used to assign a strength to every signal, and agents process their signal buffer by strength order.

Let us now apply these principles to create a functional object matching system.

3 Object matching with a hierarchy of agents

3.1 Overview

The overall architecture is presented on Figure 1. It is composed of two parts. The first one is based on SIFT matching, and is a pyramid of agents whose levels roughly correspond to LBU's steps. The second one analyzes gray level

histogram similarities and contains only one agent. Finally, the architecture has six different kinds of agents.

The bottom-up workflow in the SIFT part is similar to LBU. The detection system starts with the SiftPointComputer agents, which extract SIFT points in the image under analysis and send one signal to their SiftMatcher parent for each point. These SiftMatcher agents then look for the best SIFT match in the object database and send one signal to their SiftMain parent for each match. The SiftMain agent clusters SIFT matches that vote for similar object poses. When a cluster is updated, the associated pose hypothesis is embedded into a signal and sent to the Main agent.

Top-down signals are sent by the SiftMain and Main agents. According to current hypotheses, the SiftMain agent makes SIFT point predictions in order to accelerate the process. These predictions are then handled by the SiftPointComputer agents. The Main agent also emits top-down signals for the IntensityHistogramComparator agent to request histogram difference analyzes.

3.2 Evaluating visual data relevance

To evaluate the relevance of visual data w.r.t. the object matching task and make decisions, a unified scheme is highly desirable. We propose to rely on so-called *a contrario* statistical methods [11]. They measure the significance of an event from its probability of occurrence under a model of chance. In our application, chance means that no object of the database is present. Thus, the lower the probability of occurrence of an event under the chance model, the higher the confidence that it is associated to an object of the database. The main interest of this approach is that it is generally much easier to estimate accurate probabilities when no object is present, than the contrary. Let w be the visual data associated to a signal. We propose the following procedure to assign a strength to the signal holding w:

1. Determine a discriminative random variable X, such that $X(w_1) < X(w_2)$ implies that the visual data w_1 is *less* likely to be explained by chance than w_2.
2. Compute the cumulative distribution function $P_c(X \leq x)$ under the chance model. Analytical computations are often intractable, but it is possible to get empirical estimations by observing X values obtained on background images without any object of the database, which are simple to collect.
3. The lower $P_c(X \leq X(w))$, the higher the confidence that w is not due to chance. To get a signal strength proportional to the importance of the data, we take the inverse of this probability and set the strength of the signal associated to w to:

$$S(w) = \frac{1}{P_c(X \leq X(w))}$$

This statistical procedure also enables adaptive behaviors. Estimations of P_c are done on images which do not contain the objects to detect. These estimations will depend on the choice of these images: outdoor, indoor, highly textured, etc.

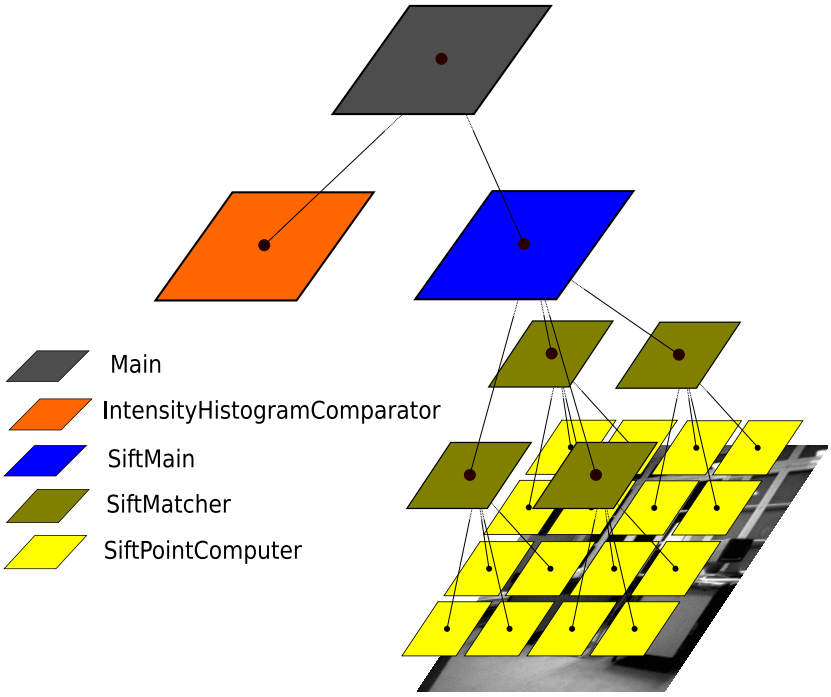

Fig. 1. Our architecture for object matching. At the lowest level, 16 SiftPointComputer agents extract SIFT points from the image and send them to their SiftMatcher parent. SiftMatcher agents then look for the closest SIFT point in the object database. SIFT matches are then handled by the SiftMain agent, which clusters them into compatible object pose hypotheses and send top-down predictions for future SIFT matches. The Main agent gathers SIFT evidence for an object presence, requests top-down histogram analyses from the IntensityHistogramComparator agent, and finally takes decisions.

If there is an *a priori* on the environment, it can be included here. It is also possible to incrementally update these estimations online in a video or robot navigation context.

Thanks to its genericity, this procedure can be used everywhere in the system, including the object presence decision step. Now, the discriminative random variables remain to be defined for each kind of visual data. This is detailed in the next Sections, where agent behaviors are described more precisely.

3.3 SiftPointComputer agents

These agents create one bottom-up signal for each SIFT point within their receptive field on the image under analysis. At this step, only a priori information can be used to evaluate the relevance of SIFT points. We exploit saliency estimates from the saliency maps of [12]. This attentional bias is particularly interesting

for outdoor scenes, as will be shown in Section 4. For these signals, the discriminative random variable X_p associated to a SIFT point p is the opposite value of the normalized saliency map at the location of p. This way, the lower $X_p(p)$, the higher the saliency, and the higher the probability that there is an object of interest.

SiftPointComputer agents are also influenced by top-down predictions emitted by the SiftMain agent. A prediction specifies an object pose hypothesis and a SIFT point of the object for which a corresponding point should be found in the image. One can deduce from the candidate object pose the theoretical location, scale and orientation of the object SIFT point in the image, if the object is present. Thus, SiftPointComputer agents will look for the closest image SIFT point which is compatible with the object pose. Details on the notion of compatibility between poses is given in Section 3.5. A bottom-up signal is created if a compatible point is found. Let $P_\Delta(p, \delta)$ be the probability that a SIFT point p in the image is compatible with pose δ by chance. Definition of P_Δ can be found in [13]. For predictions, we define the discriminative random variable X_{pp} for a point p as the joint probability of having p compatible with the candidate pose and such a high saliency, if there were no object:

$$X_{pp}(p) = P_\Delta(p, \delta) \times P_c(X_p \leq X_p(p))$$

3.4 SiftPointMatcher agents

Given a signal containing an image SIFT point, these agents find its closest match in the object database. They generate one bottom-up signal for each match. In [10], it was shown that the ratio between the distance to the closest point and to the second closest point in the database is very good at discarding false matches. The smaller the ratio, the stronger the match. We also take into account that an object with many SIFT points is more likely to accidentally hold the closest SIFT point than an object with a few SIFT points. Let $D(m)$ be the distance ratio for a SIFT match m and O the object to which the matched point belongs, we define the following discriminative variable for match signals:

$$X_m(m) = P_c(D \leq D(m)) \times \frac{\text{number of SIFT points in } O}{\text{number of SIFT points in the database}}$$

3.5 The SiftMain agent

It receives bottom-up signals emitted by SiftPointMatcher agents containing SIFT point matches. It identifies groups of matches leading to compatible poses using the same Hough-based procedure as in [10].

When it receives a new match, the SiftMain agent either adds it to an already existing hypothesis with the same pose, or creates a new hypothesis. Then it builds a bottom-up signal containing the updated hypothesis. The discriminative random variable for these signals is defined as follows. Let us suppose that we get k matches voting for a pose hypothesis h. Let n be the number

of potential matches one can get for h, i.e. the number of SIFT points lying in the expected bounds of the candidate object in the image under analysis. Let $\{m_1, m_2, \ldots, m_k\}$ denote the set of k SIFT matches, ordered such that: $X_m(m_1) \leq X_m(m_2) \leq \ldots \leq X_m(m_k)$. To measure how unlikely a set of matches is to be accidental, we could evaluate the probability of observing a group of k matches among n voting for the same hypothesis with so low discriminative variables X_m. However this probability is difficult to compute, and following [11], we rely here on an estimation of the expectation of the number of groups of k matches having as low X_m values as h. This expectation acts as a bound on the previous probability, thanks to Markov inequality, and provides our discriminative variable X_s for an hypothesis h:

$$X_s(h) = \binom{n}{k} \times \prod_{i=1}^{k} P_c(X_m \leq X_m(m_k))$$

The SiftMain agent also emits top-down predictions for futures SIFT matches. These predictions contain the current object pose hypothesis and a SIFT point which has not yet been matched. To look for most reliable SIFT points first, a score is given to each SIFT point during an offline learning phase. Each object is artificially added to background images, with various affine transformations and degrees of noise. We then apply our object matching algorithm on theses images, and whenever a SIFT point is correctly matched the distance ratio is added to its score. It is easy to know when a SIFT match is correct since we generated the training images. Then, to make predictions, the SiftMain agent starts with points having the highest scores and thus the highest probability to be present and discriminative.

3.6 The IntensityHistogramComparator agent

The Main agent can send top-down requests for histogram analysis for a given object pose hypothesis. The IntensityHistogramComparator agent computes the difference between the normalized histogram of the candidate object with the normalized histogram of the projected candidate zone in the image and creates a bottom-up signal holding the result . The difference between the two histograms is measured using the χ^2 distance, and this directly gives the discriminative variable for histogram signals X_h.

3.7 The Main agent

It receives bottom-up object pose hypothesis signals from the SiftMain and the IntensityHistogramComparator agents. When histogram information is not available for an hypothesis, it sends a top-down request, which will be handled by the IntensityHistogramComparator agent. Once all the information is collected for an hypothesis h, it decides whether h is due to chance or not, according to the accumulated evidence. The latter is given by the joint probability of having

so many SIFT matches voting for h and having such a low histogram difference when there is no object:

$$X_a(h) = P_c(X_s \leq X_s(h)) \times P_c(X_h \leq X_h(h))$$

X_a then has to be thresholded to take decisions. To get robust and adapted decision thresholds, we follow [11] and compute the expectation of the number of hypothesis with such evidence in the image under analysis, if there were no object. Since only a few matches get clustered when the database is quite big, the number of analyzed hypothesis in an image can be approximated by the number of SIFT points N_s in the image. Thus, we get:

$$\text{NFA}(h) = N_s \times P(X_a \leq X_a(h))$$

If NFA(h) is very low, then the candidate object is certainly present. A threshold on this value has a physical interpretation: if only hypotheses h such that NFA(h) $< \varepsilon$ are accepted, then it is guaranteed that in average, less than ε false detections will be made in an image.

4 Results

We evaluated our approach using the COIL-100 database [14] with a procedure similar to [15] to generate two difficult test sets of cluttered images. Test images were obtained by embedding each of the 100 objects on 640x480 background images. One test set contained only indoor background images, and the other one only outdoor background images. A contrario probability distributions were learned separately for each test set using 10 additional indoor and outdoor background images. Each object was inserted with a planar rotation of 30 degrees, a shear of 0.1, and a scale factor of 0.7. Finally, 5% of noise was added to the final images. Since many objects of COIL-100 are not very textured, this detection task is rather difficult. Processing times were measured on a 2.4 Ghz dual-core processor. It should be noted that since our focus is not on absolute time, no particular optimizations were implemented. In particular, no kd-tree is used for SIFT point matching.

The performance gain of our approach is summarized in Figure 2. These performance profiles show the obtained detection rates as a function of running time. The decision thresholds were chosen so that there were no false alarms on the test set. This figure clearly shows that parallelism, prioritization and top-down processing significantly increase detection rates while drastically reducing execution times. LBU detection times only depend upon the number of SIFT points in the analyzed image, whereas execution times also depend on objects saliency for our system. On indoor images, some of the most salient and textured objects can be detected in less than 20ms, and the maximal detection rate of LBU can be obtained in less than 90ms instead of 500ms.

On outdoor images, saliency maps are very good at predicting object locations, and thus detection times are even more reduced when saliency values are used to set signal strengths in SiftComputer agents. In this case, LBU's maximal detection rate can be achieved in about 40ms.

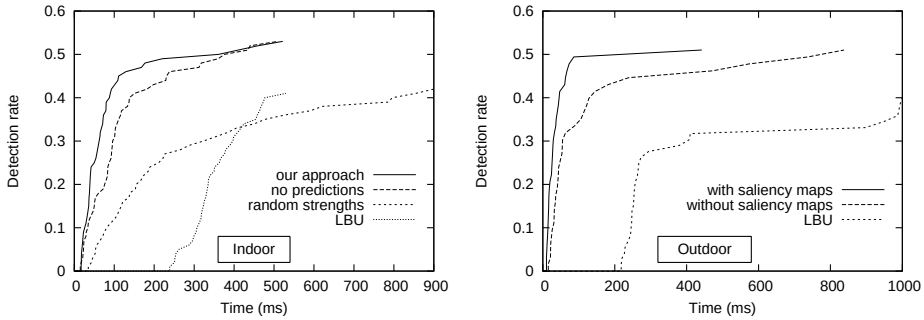

Fig. 2. For each time limit, the detection rate of different algorithms are shown. Decision thresholds were chosen to get no false alarms. The left figure is obtained on a test set with complex indoor images. In this case, the saliency maps in SiftComputer agents were not used. Our approach clearly outperforms LBU both in terms of detection rates and processing time. Since we have much more data to process, prioritization is essential: execution times obtained with random signal strengths are very poor (the curve continues and will reach the optimal detection rate in about 2 seconds). Top-down SIFT point predictions also brings a significant gain on execution time. The right figure is obtained on an outdoor test set and shows the possible speed improvements when using saliency maps in SiftComputer agents.

5 Conclusion

We have experimented an architecture for object matching based on a hierarchy of agents running in parallel. This architecture allows a mixed use of top-down and bottom-up processing. Bottom-up analysis efficiently proposes object pose hypotheses using local feature matching, while top-down influences accelerate the process by predicting future matches and increase detection rates with histogram analysis. To further improve detection rates, early bottom-up thresholds were replaced by a systematic evaluation of the statistical relevance of visual data. Then at any time, the system focuses on most promising hypotheses, and thanks to asynchronism, detection times can be greatly reduced. The system can also be constrained to run in a pre-defined time, in which case only the most salient and textured objects will be detected.

These results are encouraging and motivate further work. The concepts behind our architecture are not limited to object matching, and we are currently investigating the application of the same approach to object class detection. Very good top-down generative models have been proposed for this vision task, and we expect their combination with bottom-up analysis to be fruitful. Moreover, object matching and object class recognition are not incompatible applications, and since our architecture is modular and uses a unified statistical scheme, they could be used jointly and compete to provide the best image interpretation.

References

1. Kokkinos, I., Maragos, P., Yuille, A.: Bottom-Up & Top-down Object Detection using Primal Sketch Features and Graphical Models. Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Volume 2 (2006) 1893–1900
2. Ulusoy, I., Bishop, C.: Generative versus discriminative methods for object recognition. Proc. CVPR **2** (2005) 258–265
3. Lampinen, J.: Sequential Monte Carlo for Bayesian Matching of Objects with Occlusions. IEEE Transactions on Pattern Analysis and Machine Intelligence **28**(6) (2006) 930–941
4. Tu, Z., Chen, X., Yuille, A.L., Zhu, S.: Image parsing: Unifying segmentation, detection, and recognition. International Journal of Computer Vision **63**(2) (2005) 113–140
5. Delorme, A., Rousselet, G.A., Mace, M.J., Fabre-Thorpe, M.: Interaction of top-down and bottom-up processing in the fast visual analysis of natural scenes. Technical report, Cognitive Brain Research 19 (2004)
6. Tsotsos, J.: Toward a computational model of visual attention. Early vision and beyond (1995) 207–218
7. Borowy, M., Jolion, J.: A pyramidal framework for fast feature detection. Proc. of 4th Int. Workshop on Parellel Image Analysis (1995) 193–202
8. Draper, B., Collins, R., Brolio, J., Hanson, A., Riseman, E.: The schema system. International Journal of Computer Vision **2**(3) (1989) 209–250
9. Guhl, P., Shanahan, P.: Machine Perception using a Blackboard Architecture . The 5th International Conference on Computer Vision Systems Conference Paper (2007)
10. Lowe, D.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision **60**(2) (2004) 91–110
11. Desolneux, A., Moisan, L., Morel, J.M.: Maximal meaningful events and applications to image analysis. Annals of Statistics **31**(6) (2003) 1822–1851
12. Itti, L., Koch, C., Niebur, E., et al.: A model of saliency-based visual attention for rapid scene analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence **20**(11) (1998) 1254–1259
13. Lowe, D.: Object recognition from local scale-invariant features. International Conference on Computer Vision **2** (1999)
14. Nene, S., Nayar, S., Murase, H.: Columbia Object Image Library (COIL-100). Techn. Rep. No. CUCS-006-96, dept. Comp. Science, Columbia University (1996)
15. Stein, A., Hebert, M.: Incorporating Background Invariance into Feature-Based Object Recognition. Seventh IEEE Workshop on Applications of Computer Vision (WACV) (2005)