An Energy Minimization Method for Matching and Comparing Structured Object Representations*

Robert Azencott and Laurent Younes

Sudimage Research Laboratory
and
CMLA, ENS Cachan, CNRS URA 1611
61 av. du Président Wilson, 94235 Cachan CEDEX, France
email: younes@cmla.ens-cachan.fr

Abstract. We present a general method for matching segmented parts of objects by energy minimization. The energy is designed in order to cope with possible imperfections of the compared segmentations (merged, or missing regions), and relies on the comparison of shape and positional descriptors. The minimization of the energy is performed by a simulated annealing procedure

1 Introduction

Structural representations describe objects by a list of elementary parts (or components, or primitives) and relations between them. Such representations may be used for a large range of objects. One-dimensional representations describe curves as objects outlines ; it may be a polygonal approximation of the curve, a list of feature points, the sequence of its concave/convex parts ([11]). Many segmentation procedures have been designed to provide 2D and 3D decomposition. Representation by line segments or feature points are also commonly used. Recognition is then performed by comparing these representations, and graph-matching paradigms and algorithms have been designed by many authors.

In any case, when comparing two objects, at least one representation is directly extracted from observed data. In fact, most of the methods dedicated to object recognition assume the existence of of prototypes, for which a presumably perfect representation is computed once and for all, and consider that an instance of one of them is observed and must be recognized. The protype which provides the best match to the observation is selected. However, the extraction of the structural representation from real data is a difficult task. It is, for example, hardly possible to design a perfectly reliable image segmentation procedure, because of the inherent ambiguities of 2-D images (shadows, specularities, occlusions...). Thus, at least one of the compared structures will be imperfect, and

* This work partially comes from a project on object recognition, supported by the D.R.E.T, helded at SUDIMAGE research laboratory. Publication is courtesy of the D.R.E.T.

sometimes both, since an exact description of the prototypes need not be always available. Matching (and comparison) has to cope with these imprecisions. A general approach for inexact graph matching has been presented in [9]. In this reference, trangressions of relational constraints between the matched primitives are allowed up to a certain amount. However, for many representations, typical variations may strongly affect the structures. The case of segmented images is illustrative : one of the most frequent perturbations is oversegmentation, in which case the graph structure may be drastically changed. This corresponds to a union of nodes in the graph, which cannot be ignored by the matching process. In [12], a solution is provided by the use of augmented association graphs. Our approach in the present work aims at the same objective, that is comparing structured representations in which similar objects may yield significant differences in the structures, albeit with substantial differences.

A first difference with the most frequent approaches is that our primary goal is not explicitely recognition, but merely comparison. We want to design a method to decide whether two representations are similar or not, and incidentally quantify this similarity (of course, recognition would be the goal of a second stage). In addition, we do not assume that one of the representation is perfect, so that both objects will be treated in a symmetric way.

So, our matching paradigm is not one-to-one but many-to-many, in order to handle possible unions of components in one representation before matching the other. Aggregating primitives yields a new, simpler, but less informative, representation of the object. It is always possible to simplify in that way the original representations so that they become similar, or at least comparable. This provides the principles of our method : object views will be considered as similar if slight simplifications of their representation can be matched so that they have a similar structure.

A large part of the paper deals with the problem with a more or less general formulation. We assume a decomposition into components, but we do not explicitely place a graph structure on the representation, the relationships being described by a family of descriptors, which are real valued functions of several components. A simplification of this structure is formalized as an operation which permits to agglomerate parts, or to discard them. In such a context, we design a method for matching, then comparing representations. Our approach is variational and we design a cost function which is small for a correct matching. The construction of the cost function is based a general framework, in which we include unspecified features which depend on the application.

In the second part of the paper, we complete the cost function in order to deal with the particular case of comparing segmented views of objects.This application is a part of a global project on multi-view object recognition.

2 General formulation

Let an object representation, $\mathcal{R}$, be composed with a certain number, N, of parts (N depending on the object), which we shall write $\mathcal{R} = (R_1, \ldots, R_N)$. The representation $\mathcal{R}$ is characterized by a family of *relational descriptors*, which describes the relationship between the elements of $\mathcal{R}$: let q be a positive integer, the order of the description, and $\mathcal{P}_q$ be the collection of all the subsets of $\{1, \dots, N\}$ with less than q elements : for each $C \in \mathcal{P}_q$, denote by $\mathcal{R}_C$ the collection $(R_i, i \in C)$, and the descriptor associated to C is a feature $\lambda_C(\mathcal{R})$ computed from $\mathcal{R}_C$. Note that λ_C can be multi-dimensional (even infinite dimensional : for example the regions boundaries in the case of segmented images of objects). In our experiments, the order q is limited to $q = 2$. One basic principle is that two representations $\mathcal{R}$ and $\mathcal{R}'$ will be considered to be similar if their descriptors are close.

When given two object representations, one must determine compatible orderings before comparing, that is, the parts in each representation have to be matched one to another. Moreover, since the number of components in the representations of two different objects need not be equal, some parts in one representation may not find any related parts in the other one. Finally, we may also be in a situation in which a component R_k is divided into several parts in the other representation. This is likely to happen when the representation is extracted from observed data (as in our study, in which we use image segmentation algorithms), but this can also correspond to a case in which the second object is similar, but simpler than the first one. Thus, some transformations of the representations must be allowed before the matching: a) discarding components from any of the two representations; b) grouping several parts together. For the last one, we assume that some aggregation operation is available, which associates to two components R_1 and R_2 their union which will be denoted $R_1 \cup R_2$. We furthermore assume that this operation is commutative and associative. In our application, components are regions in the image plane, and aggregation simply is set union. These operations have the effect to simplify a representation ; we do not allow for the possibility to divide a component into several ones, which would induce a complexification of the representation, and would require some extraneous information which is not necessarily available. Thus, the matching problem is to determine compatible simplified representations of the objects from the original representations. To summarize, its solution requires to

a– Discard some components from each representation (those which cannot be correctly matched)

b– Aggregate some components in each representation

c– Find compatible orderings of the aggregated components

It is clear that there is no reason for which any of these three steps could be performed before the other two : it is impossible to decide whether a component in the first representation should rather be discarded or aggregated to others unless one has tried to match the aggregated group to some other group in the second representation. These operations must in fact be performed simultaneously. For this reason, we minimize a single criterion combining several cost functions, each of which being concerned with one step (a– to c–) of the procedure.

3 Notation for the matching problem

We fix some notation. Let $\mathcal{R} = (R_1, \ldots, R_N)$ ans $\mathcal{R}' = (R'_1, \ldots, R'_M)$ be the two representations to match. In order to indicate that two groups of components are matched, it suffices to mark each element of these groups with a common label. Denote by $\{1, \ldots, L\}$ a set of labels, to be used to mark matched components, and add to it a new label, 0, to mark the components which have not been matched. Specifying a common labelling of the representations boils down to defining two mappings, ϕ and ψ, which respectively provide the labels of the components R_i and R'_i:

$$\phi : \{1, \ldots, N\} \longrightarrow \{0, \ldots, L\} \; , \; \psi : \{1, \ldots, M\} \longrightarrow \{0, \ldots, L\} \, .$$

We do not know the exact number of labels in the matching, that is L is unknown, but it can be bounded (for example, $L \leq L_0 = \min(N, M))$). Labels in $\{1, \ldots, L\}$ being reserved for matched components, introduce the notation (for $k \in \{0, \ldots, L\}$) $\Sigma_k = \cup_{\phi(i)=k} R_i$ and $\Sigma'_k = \cup_{\psi(i')=k} R'_{i'}$. The discarded parts, Σ_0 and Σ'_0 may be empty, but, for $k = 1, \ldots, L$, we impose that Σ_k and $\Sigma'_k \neq \emptyset$.

Thus, another way to formulate the problem is that we are looking for two simplified representations $\mathcal{S} = (\Sigma_0, \ldots, \Sigma_L)$, $\mathcal{S}' = (\Sigma'_0, \ldots, \Sigma'_L)$, with the convention that Σ_0 and Σ'_0 are the (possibly empty) aggregation of discarded components, and that the components Σ_k and Σ'_k are matched together, for $k \in \{1, \ldots, L\}$. The representations $\mathcal{S}$ and $\mathcal{S}'$ are called *simplifications* of the original representations $\mathcal{R}$ and $\mathcal{R}'$.

We assume that, for any object representation $\mathcal{R} = (R_1, \ldots, R_N)$, one can compute relational descriptors $\lambda_C(\mathcal{R})$, where C are subsets of $\{1, \ldots, N\}$, which describes some relationship between the R_i, $i \in C$. We assume that λ_C only depends on $R_i, i \in C$ and on $\mathcal{U}(\mathcal{R}) = \cup_{k=1}^{N} R_k$ (this last dependence holding to allow global normalisation of the descriptors). Note that if $\mathcal{S}$ is a simplification of $\mathcal{R}$, we set $\mathcal{U}(\mathcal{S}) = \mathcal{U}(\mathcal{R})$ (that is we include the discarded regions to compute the global properties of $\mathcal{S}$), so that $\mathcal{U}(\mathcal{R})$ and $\mathcal{U}(\mathcal{R}')$ are invariant of the matching process. Thus, our problem is to determine $\mathcal{S}$ and $\mathcal{S}'$ so that $\lambda_C(\mathcal{S}) \simeq \lambda_C(\mathcal{S}')$.

Together with the similarity of the simplifications of $\mathcal{R}$ and $\mathcal{R}'$, we add, in order to evaluate the matching, a parcimony constraint to ensure that the representations are not over- simplified : there should not be too many discarded regions (otherwise, there will be nothing left to compare) and the aggregation process should be limited, in order to keep as much as possible of the information contained in the original representations. The cost function we built takes into account the previous constraints as a sum of penalty terms.

4 Quantitative evaluation of the quality of the matching

4.1 General principle

We follow a variational approach and define a cost function which will be small when the matching is adequate (according to the previous qualitative criteria).

The cost function will be the sum of several terms, each of which being designed in order to constrain a particular behaviour. Since we have selected three criteria, there will be three terms, each of them respectively aiming at

1- Similarity of the descriptors λ_C computed on the simplifications $\mathcal{S}$ and $\mathcal{S}'$
2- Restriction of the sizes of the sets Σ_0 and Σ_0' (unmatched regions)
3- Limitation of the aggregation process : Σ_k and Σ_k' should not be composed with too many regions of the original segmentations

In some applications, one can imagine some hard constraints imposed on the aggregation process. For example, connectivity of the aggregates may be enforced, or aggregation of some uncompatible components may be forbidden.

4.2 Cost function

In this section, we only give the general form of the cost function, leaving the detailed description to the next sections. This function is of the kind

$$E(\phi,\psi) = E_1(\phi,\psi) + E_2(\phi,\psi) + E_3(\phi,\psi)$$

each of these terms corresponding to one of the criteria 1 to 3 above.

In order to estimate the importance of the components (for example to quantify the second criterion), we assume that we can compute, for each part R of a representation, a measure of size, which we shall denote by $\mathcal{A}(R)$. In addition, to compare the descriptors, we assume that, for all $k \leq q$ (q being the order of the description), we have designed a measure of the difference between two descriptors λ_C and λ_C' for $|C| = k$, which will be denoted $\Delta_k(\lambda_C, \lambda_C')$. We shall put, writing, for short, $\lambda_C = \lambda_C(\mathcal{S})$ and $\lambda_C' = \lambda_C(\mathcal{S}')$ for subsets C of $\{1,\ldots,L\}$,

$$E_1(\phi,\psi) = \sum_{p=1}^{q} \sum_{C,|C|=p} \Delta_p(\lambda_C, \lambda_C')\mu_p(C),$$

where μ_p is a weight which depends on the sizes of the sets Σ_k and Σ_k' for $k \in C$.

For the second criterion, we simply set $E_2(\phi,\psi) = \mathcal{A}(\Sigma_0) + \mathcal{A}(\Sigma_0')$.

Finally, to define the cost associated to point 3, we assume a dispersion measure for the representation $\mathcal{R}$, denoted Γ, which can be computed on any family $R_i, i \in V$ (with $V \subset \{1,\ldots,N\}$), which is large when the sets R_i are (in a sense to be defined) far apart one from each other. Similarly, a dispersion measure Γ' in $\mathcal{R}'$ is defined. Let us put, for short, $\Gamma_k = \Gamma(R_i, i \in \phi^{-1}(k))$ and $\Gamma_k' = \Gamma'(R_i', i \in \psi^{-1}(k))$

$$E_3(\phi,\psi) = \sum_{k=1}^{L} \nu(\Sigma_k)\Gamma_k + \sum_{k=1}^{L} \nu(\Sigma_k')\Gamma_k'.$$

There again, ν is a weight, depending on the sizes of the components Σ_k and Σ_k'.

A good choice of the weights μ_k and ν is decisive for the success of the method. They can be calibrated by analyzing the variations of the cost function under simple transformations of the matching. This is a general method (cf [1]) which ensures that the weights are calibrated in order to provide a correct matching at least for particular cases. In order to carry on the analysis, we make some additional assumptions which will be satisfied in the application below.

The first one is that the comparators Δ_k and the dispersion measures Γ and Γ' are normalized so that their typical values are near the unity. The second one is that the size measure is additive, that is $\mathcal{A}(R_1 \cup R_2) = \mathcal{A}(R_1) + \mathcal{A}(R_2)$. Under these hypothesis, let us consider the following case. Start with simplifications $\mathcal{S}$ and $\mathcal{S}'$ with L labels, and consider the variation in which all the components which form Σ_L and Σ'_L are discarded and added to Σ_0 and Σ'_0. Because of the additivity assumption, the variation of E_2 would be $\Delta E_2 = \mathcal{A}(\Sigma_L) + \mathcal{A}(\Sigma'_L)$.

The variation of E_3 is $\Delta E_3 = -\nu(\Sigma_L)\Gamma_L - \nu(\Sigma_L)\Gamma'_L$, and for E_1, it is

$$\Delta E_1 = -\sum_{p=1}^{q} \sum_{C,|C|=p,L\in C} \Delta_p(\lambda_C, \lambda'_C)\mu_p(\Sigma_i, i \in C).$$

Without any knowledge about the regions which have been discarded, there is no reason to priviledge any of the terms of the cost function. Thus, the weights should be tuned in order that each of the terms have comparable size for "average" values of Δ_p, Γ_k (according to our hypothesis, these average values are 1). The analysis will also provide "average values" for the weights (this is why we speak of "calibration" of the weights). It appears however that the weights which are provided by such rough computations are sufficiently well fitted to yield good results, and that slight variations around these values provide matchings of comparable quality. If needed, further variations of the same analysis can provide additional constraints which would induce some more acute information on the weights.

Thus, $\mathcal{A}(\Sigma_L) + \mathcal{A}(\Sigma'_L)$ should have the same size as $\nu(\Sigma_L) + \nu(\Sigma'_L)$ which naturally leads to set $\nu(\Sigma) = \mathcal{A}(\Sigma)$. This should also be the size of

$$\sum_{p=1}^{q} \sum_{C,|C|=p,L\in C} \mu_p(\Sigma_i, i \in C),$$

and we assume that each term of this sum has the same size, so that relationships of all orders have the same influence. The first term (for $p = 1$) is $\mu_1(L)$, and it is natural to set $\mu_1(L) = \mathcal{A}(\Sigma_L) + \mathcal{A}(\Sigma'_L)$.

Now, for $p = 2$, the term is $\sum_{k \neq L} \mu_2(k, L)$ which should have the same size as $\mu_1(L)$. One possibility is to put $\mu_2(k, L) = \mu_1(k)\mu_1(L)/\sum_l \mu_1(l)$, with the assumption that $\mu_1(L)$ is small compared to $\sum_{k\neq L} \mu_1(k)$. Terms of order larger than 2 can be handled similarly.

5 Minimization Procedure

The discrete minimization problem, in its full generality is a hard problem. It requires to find partitions of the sets $\{1, \ldots, N\}$ and $\{1, \ldots, M\}$, and the best matching between them. The size of all acceptable matchings is quite large (about 10^{12} for $M = N = 10$, 10^{31} for $M = N = 15$) so that the optimal matching cannot be determined by systematic exploration. In some cases, for example, when dealing with curves or acyclic graphs, global optimization algorithms, such as dynamic programming, may be devised. In all cases, simulated annealing is a good general procedure for massive discrete optimization. This is the one we have used in our application. In order to determine the labels ϕ and ψ, the algorithm works as follows. At each stage, it proposes a small modification of the current ϕ and ψ, which induces some variation ΔE of the cost function. The modification may be refused, and this is done with probability $\max(0, 1 - \exp(-\Delta E/T))$, T being a factor which slowly decreases to 0 during the procedure. If the elementary modifications are suitably designed, and the decreasing of T is slow enough, the algorithm provides the global minima of E.

Besides the theoretical slow decreasing rate of T (which is practically unachievable, and replaced by an exponentially fast decreasing rate — cf. [4] for a justification of this choice in the case of finite horizon annealing processes), the other condition for a good behaviour of this minimization algorithm holds on the choice of the elementary transition at each time. Assume that, when the current state is (ϕ, ψ), the new proposal is taken at random in a set $A(\phi, \psi)$ (which may vary with time). Then, sufficient conditions for convergence are :

if $(\phi', \psi') \in A(\phi, \psi)$, *then* $(\phi, \psi) \in A(\phi', \psi')$ *and both sets have the same cardinality.*

there exists a fixed integer n such that, a transition between any (ϕ, ψ) *and* (ϕ', ψ') *is possible, with positive probability, in n steps.*

Note that, when modifying ϕ and ψ, we must take care that the constraint that no label can be used for a representation and not for the other, is satisfied. To simplify the implementation, we fix the number L_0 of labels and allow for the possibility of unused labels. If $L_0 = \min(M, N)$, this does not affect the generality of the search. The constraint is then : for all $k \in \{1, \ldots, L_0\}$, $\phi^{-1}(k) = \emptyset \Leftrightarrow \psi^{-1}(k) = \emptyset$.

The sets $A(\phi, \psi)$ that we propose may be of two kinds. The first one contains transformations which simultaneously modify the values of $\phi(i)$ and $\psi(i')$ among the family of all admissible new labels. The second one contains transformations which exchange the values of $\phi(i)$ and $\phi(j)$ (or $\psi(i')$ and $\psi(j')$) if these values are different, and different from 0. Both types satisfy the conditions above, and they are alternated during the procedure.

Remark : The updating phase may become computationaly costly when the order of the description increases. For $q > 2$, it becomes necessary to define λ_C only for a restricted family of sets C with cardinality q, so that, for any i, the number of C with cardinality q for which λ_C is modified remains bounded independently of q. For example, the restriction may be to sets C for which all components are large enough, or close enough one to each other. But, this notion of size, or nearness, depend on the components whose labels are in C, and maybe also on other global properties of the representations : one consequence of it is that, under such a framework, when comparing two segmentations, some

λ_C could be defined in one case, and not in the other. This may be bypassed by letting $\lambda_C = K$, a constant, if C is not admissible, so that, effective computation of λ_C still is restricted to admissible C. Choosing K large enough is a way to forbid matching in which admissible C are matched to non admissible C.

6 Application : comparison of segmentations

6.1 Introduction

We now particularize the above approach to the problem of comparing segmentations. Given an image of an object, trying to separate it into functional parts is attractive, but a genuine functional decomposition is hardly feasible without high-level information on the observed object. A less ambitious program is to use the decomposition given by low-level image segmentation algorithms which separate the picture into homogeneous parts, based on features related to gray-level or color distribution. This representation often provides substantial information on the object, each homogeneous region in the image being most of the time associated to a single functional part of the object. However, starting at low-level, one has to cope with the usual drawbacks of image acquisition. Light variations, shadows, specularities, are elements which may cause errors and biase the results, and it is hopeless to expect that any segmentation procedure would provide outcomes bypassing these problems. Some enhancement, more robustness may be obtained by carefully selecting the algorithm, and it is an important, still largely open, issue in image processing to design efficient, robust, using minimal a priori information, low-level segmentation methods. However, when passing to comparison, the possibility of having to deal with over-segmentations, or strong variations in the shapes of regions, must be kept in mind, and this is precisely what is handled by our matching method.

Thus from now on, our representation is a family $\mathcal{R} = (R_1, \ldots, R_N)$ of regions of the image plane. Simplifications are obtained by discarding regions and aggregation by set union. Before completing the concepts presented in the previous paragraphs, we start with a brief discussion of the segmentation algorithm.

6.2 Segmentation algorithm

We just say a few words about the way we segment images. It is not in our intent to give a precise description of the method, which would be too long and out of the scope of the paper. The aim is, given a 2-D view of an object, to provide a partition of the image into regions $R_1, \ldots, R_N$ which corresponds to homogeneous parts of the picture relatively to a chosen criterium. In the present study, segmentation is based on colour. Once this characteristic is fixed, the procedure is entirely unsupervised, with respect to the number of regions, which is unknown, or to the various parameters (α, λ_1 and λ_2 below) which are estimated on line. The final segmentation is obtained by minimizing a discrete cost function, the general form of which being

$$E = \sum_i V(R_i) + \sum_{i \neq j} \sum_{(s,t) \in \partial_{ij}} [\lambda_1 - \lambda_2 \Delta_{st}] .$$

where : $V(R_i)$ measures how much colour varies in region R_i; for $i \neq j$, ∂_{ij} is the (possibly empty) common boundary of regions R_i and R_j, composed with couples of pixels (s,t) such that $s \in R_i$, $t \in R_j$, and s and t are nearest neighbours on the image grid; Δ_{st} is an indicator function, equal to 1 if the difference of the colours at pixels s and t is larger than a threshold α and 0 if not.

We assume that the object is completely included in the picture, and we discard from the segmentation all the regions which meet the image frame. Assuming that the background is more or less homogeneous, this will discard most of the parts of the picture which do not belong to the object. Some piece of background may however be still present in the final segmentation, which provide a new kind of perturbation which must be handled by the matching procedure.

6.3 Descriptors

Generalities There are some desireable properties which may be expected from the descriptors. A first property is that they are rich enough to characterize the view of the object with satisfying accuracy. A second one comes from the fact that the object characterization must hold up to some parasit rigid transformation, since the relative positions of the compared objects are unknown, but should not influence the matching. The minimal rigid invariance which should be imposed are scale invariance and rotation invariance in the image plane. These are the invariance which will be explicitely addressed in this work. Affine invariance can also be required, to cope with variations in the angle of view of the objects. This seems too be less important than rotation and scaling, especially for complex objects, since variations of the angle of view are likely to yield appearance of occluded parts which cannot be modeled by affine transformation and would rather require a multi-view approach. Affine invariance however brings more robustness, and we will give some indication on how this can be achieved. Note that the use of relational descriptors gives much more latitude for the construction of invariant features, since there are much more invariant functions of several variables than of only one.

A last property which has to be aimed at by the descriptors is computational. Indeed, during the matching, the compared descriptors depend on the simplifications $\mathcal{S}$ and $\mathcal{S}'$ which are unknown. Given the combinatorial structure of the matching algorithm, it is essential that the calculation of the descriptors could be simple enough to avoid prohibitive computer time. At least, their updating after the changes which are proposed in the annealing algorithm of section 5 should not consume too much time. This is a strong limitation to the range of acceptable descriptors, and comes somewhat in contradiction with the first requirement on the accuracy of the description, but this is essential for practical use. It seems, however, that the constraints imposed by relational descriptors of several variables (even simple ones) are restrictive enough to yield good performance of the matching without harming too much the computation time.

We now pass to the explicit presentation of the descriptors which are used. They are of two kinds : a) positional, which depend on the centers of gravity of the regions, and b) relative to shape, which will be based on (rough) descriptions of the outline of the regions. The order of the representation is $q = 2$, so that we only have unary and binary descriptors. We thus assume that we have two segmentations $\mathcal{R} = (R_1, \ldots, R_N)$ and $\mathcal{R}' = (R'_1, \ldots, R'_M)$ and try to find two matched simplifications $\mathcal{S} = (\Sigma_0, \Sigma_1, \ldots, \Sigma_L)$ and $\mathcal{S}' = (\Sigma'_0, \Sigma'_1, \ldots, \Sigma'_L)$.

Importance evaluation To measure the size of a region Σ in a simplification $\mathcal{S}$, we simply use its relative area:

$$\mathcal{A}(\Sigma) = \frac{\text{area}(\Sigma)}{\text{area}[\mathcal{U}(\mathcal{S})]}$$

where $\mathcal{U}(\mathcal{S})$ is the aggregate of all the components in $\mathcal{S}$.

This measure is used for the weights μ_k and ν, and also for the shape descriptors below. It is translation, rotation and scale invariant (in fact, it is affine invariant).

Positional descriptors The position of a region in the image plane is represented by its center of gravity, ie the mean position of the pixels which are contained in the region. In order to obtain translation invariance for unary descriptors, we use their relative position to the center of gravity of the complete object. Thus, we denote by G (resp. G') the center of gravity of $\mathcal{U} = \mathcal{U}(\mathcal{S}) = \Sigma_0 \cup \cdots \cup \Sigma_L$ (resp. $\mathcal{U}' = \Sigma'_0 \cup \cdots \cup \Sigma'_L$), which are constant during the matching. We let G_k (resp. G'_k) be the center of gravity of Σ_k (resp. Σ'_k). To induce rotation invariance, we only use the Euclidean distances between these points, letting our unary descriptors be the distance between G and G_k, denoted GG_k, and the binary descriptors be the collection of all G_kG_l, for $k \neq l$ larger than 1. Finally, in order to also obtain scale invariance, we use a unit length which depends on the total area of the segmentation : letting $A_{tot} = \sum_k \text{area}(R_k)$, we measure the lengths in terms of multiples of $1/\sqrt{A_{tot}}$.

Thus we have obtained unary and binary positional descriptors which are translation, rotation and scale invariant. If affine invariance were required, unary positional descriptors of the previous kind have to be dropped. Concerning binary descriptors, the area of the triangle (G, G_k, G_l) is an example of an affine invariant descriptor.

Shape descriptors To shorten notation, we let $A_k = \mathcal{A}(\Sigma_k)$ be the relative area of Σ_k in $\mathcal{S}$ (and $A'_k = \mathcal{A}(\Sigma'_k)$): this forms our first shape descriptor. The second one is the ellipse of inertia of the region Σ_k, which we denote by $\mathcal{E}_k$. If I_k is the matrix of inertia of Σ_k, ie

$$I_k = \int_{\Sigma_k} \overrightarrow{G_kX}\,\overrightarrow{G_kX}^t\, dX\,,$$

the ellipse of inertia (up to scaling) is defined by $\overrightarrow{G_kX}^t I^{-1} \overrightarrow{G_kX} = \text{cte}$. On the computational level, A_k, G_k and I_k can be very efficiently obtained by incremental formulae when $\mathcal{S}$ varies.

Similarly, we denote by A'_k and $\mathcal{E}'_k$ the area and ellipse of inertia of Σ'_k.

Since they are centered at G_k, the ellipses of inertia are translation invariant, but they are not rotation nor scale invariant. Therefore, the unary descriptors can only depend on the excentricities of the ellipses. The binary descriptors will be based on the comparison of the relative positions of two ellipses.

We shall in fact use two distances in order to compare two ellipses $\mathcal{E}$, $\mathcal{E}'$; we denote them by $d_0(\mathcal{E},\mathcal{E}')$ and $d_1(\mathcal{E},\mathcal{E}')$. The first one is invariant by scaling and rotation of any of the ellipses $\mathcal{E}$ and $\mathcal{E}'$, so that it only depends on the excentricities of the ellipses and will be used for unary descriptors. The second one is scale invariant, and invariant by simultaneous rotation of the ellipses (with a common angle), it thus depends on the relative positions of the ellipses. If we are only interested in comparing ellipses, there are many ways to define such distances. However once the matching will be computed, we want to use, for comparison, richer information than the ellipses of inertia, and use distances which have been designed to compare arbitrary plane curves (cf. [13]). They are computed (once the curves have been rescaled to have length 1), on the basis of the functions which give the orientations of the tangent vectors to the curves in function of the arc-length. An optimal matching is computed between these functions (denoted by θ and $\tilde{\theta}$), letting

$$d_1 = \inf_g \left\{ \arccos \int_0^1 \sqrt{g_s} \left| \cos\left(\frac{\tilde{\theta} \circ g(s) - \theta(s)}{2} \right) \right| ds \right\}$$

g being a diffeomorphism of $[0,1]$, and

$$d_0 = \inf_{g,c} \left\{ \arccos \int_0^1 \sqrt{g_s} \left| \cos\left(\frac{\tilde{\theta} \circ g(s) - \theta(s) - c}{2} \right) \right| ds \right\}$$

g being a diffeomorphism of $[0,1]$, and c being a number in $[0, 2\pi]$.

We have also used these distances to compare the ellipses. In order to reduce computation time, their values have been discretized off-line and stored in a look-up table.

We use two binary descriptors : first, the relative areas of Σ_k and Σ_l, $\frac{A_k}{A_l}$, and second, the distance $d_1(\mathcal{E}_k, \mathcal{E}_l)$. We compute in fact a single number, which is

$$B_{kl} = \frac{1}{2} \left| \log \frac{A_k}{A_l} \right| + d_1(\mathcal{E}_k, \mathcal{E}_l)$$

and, similarly for $\mathcal{S}'$, $B'_{kl} = \frac{1}{2}\left|\log \frac{A'_k}{A'_l}\right| + d_1(\mathcal{E}'_k, \mathcal{E}'_l)$.

Once again, these descriptors are not affine invariant. An affine invariant matching could de based on higher order moment-based invariant of the shape (the area of the ellipse of inertia being the only moment invariant of order 2).

6.4 Comparison of the descriptors

We now define the functions Δ_1 and Δ_2 which are used to compare unary and binary descriptors. Note that, in order that our discussion on the calibration of the weights be valid, their values much be properly normalized to have a typical range arount the unity. We set

$$\Delta_1(k) = \max\left\{\frac{1}{2}\left|\log \frac{A_k}{A'_k}\right| + d_0(\mathcal{E}_k, \mathcal{E}'_k), 2\frac{|GG_k - G'G'_k|}{GG_k + G'G'_k}\right\}. \tag{1}$$

and

$$\Delta_2(k,l) = \max\left\{2\frac{|B_{kl} - B'_{kl}|}{B_{kl} + B'_{kl}}, 2\frac{|G_kG_l - G'_kG'_l|}{G_kG_l + G'_kG'_l}\right\} \tag{2}$$

6.5 Measure of dispersion

The last point to describe is the measure of dispersion used in the cost term E_3, which has been denoted $\Gamma_k = \Gamma(R_i, i \in \phi^{-1}(k))$ and $\Gamma'_k = \Gamma'(R'_i, i \in \psi^{-1}(k))$. Note that this is the only term which refers to the original segmentation, the other ones depending only on the matched simplifications $\mathcal{S}$ and $\mathcal{S}'$. Assume that a distance D_{ij} is defined between regions R_i and R_j. We let

$$\Gamma_k = \frac{1}{A_k^2} \sum_{i,j\in\phi^{-1}(k)} D_{ij}\text{area}(R_i)\text{area}(R_j),$$

The term Γ'_k being similarly defined for the segmentation $\mathcal{R}'$.

To define the internal distance D_{ij} we take into account the topological structure of the segmentation $\mathcal{R}$. For each pair of regions R_i and R_j, we let ∂_{ij} be the possibly empty common boundary of R_i and R_j. Let a path from i to j in the set $\{1, \dots, N\}$ be a sequence $i_0 = i, i_1, \dots, i_p, i_{p+1} = j$. We define the length of such a path by a formula of the kind

$$L = \sum_{q=0}^{p} F(R_{i_q}, R_{i_{q+1}}) + \sum_{q=1}^{p} G(R_{i_{q-1}}, R_{i_q} R_{i_{q+1}})$$

where $F(R_i, R_j)$ is a cost associated to the transition between regions R_i and R_j. It is equal to a constant plus the minimum distance between all non empty boundaries ∂_{ik} and ∂_{jl}, for all $k, l \in \{1, \dots, N\}$, which is 0 if $\partial_{ij} \neq \emptyset$; $G(R_i, R_j, R_k)$ is the distance between the closest boundary of R_i to R_j and the closest boundary of R_j to R_k. Thus the length of path is large when two successive regions are not adjacent, and it crosses some large region. The distance between two boundaries ∂ and ∂' is the mean value of $d(i, \partial')$ for $i \in \partial$ plus the mean value of $d(i', \partial)$ for $i' \in \partial'$.

7 Comparison after matching

Once the combinatorial part of the comparison (ie. the maching procedure) has been achieved, it is possible to use richer descriptors to quantify the differences between the objects. We adopt a hierarchical approach, and use successive criteria of increasing complexity to decide whether the viewed objects are similar or not.

The first criterion is based on the sizes of the discarded components Σ_0 and Σ'_0 after the matching. Indeed, if the matching algorithm couldn't do better, while minimizing the cost function, than discarding a large proportion of the original components, this means that the representations were very different and there is no need to push the comparison further. So, we have a stopping crtiterion after matching which is based on

$$\rho = \max(\mathcal{A}(\Sigma_0), \mathcal{A}(\Sigma'_0)) .$$

If ρ ou ρ' is larger than a threshold (we used 0.4) comparison is stopped.

The second criterion compares the centers of gravity G_k and G'_k of Σ_k and Σ'_k. Denote by z_k and z'_k the complex numbers representing the 2D vectors $\overrightarrow{GG_k}$ and $\overrightarrow{G'G'_k}$, where G and G' are the centers of gravity of the aggregation of the components of $\mathcal{R}$ and $\mathcal{R}'$. We let τ be a number in $[0, 2\pi[$ and set

$$d_{pos}(\tau) = 2 \arccos \frac{\text{real part}(\sum_{k=1}^{L} z_k \overline{z'_k} e^{-i\tau})}{\sqrt{\sum_{k=1}^{L} |z_k|^2} \sqrt{\sum_{k=1}^{L} |z'_k|^2}}$$

which, for each τ, is a distance comparing the z_k and $e^{i\tau}.z'_k$ up to scaling. This distance is small if, after a rotation of angle τ, the configurations of complex numbers, (z_k) and (z'_k) are close enough. We compute this value for a discrete family of angles τ, and select those which are below a fixed threshold (we used 0.5). These τ are retained for the last stage, and if one could not find any, the procedure stops, and we conclude to high dissimilarity of the objects. In our experiments, we always found at least one correct τ, sometimes (but quite rarely) two.

The last criterion compares the outlines of the regions Σ_k and Σ'_k. Note that these regions can be very complex, since we have imposed no constraint on the aggregation process : Σ_k need not be convex, can contain holes. In order to obtain a reasonable candidate for the outline, we adopt the following procedure. For all $\theta \in [0, 2\pi[$, we compute the length $r_k(\theta)$ between G_k (the center of gravity of Σ_k) and the furthest point of Σ_k which belongs to the half-line of angle θ starting from G_k. The curve, parametrized in polar coordinates by $\theta \rightarrow r_k(\theta)$ will be denoted C_k and is our definition of the outline of Σ_k.

Given this, we let, for one of the τ selected at the previous stage, r_τ be a rotation of angle τ. We then compute the distance

$$d_{shape}(\tau) = 2 \arccos \frac{\sum_k \sqrt{A_k A'_k} \cos d_1(C_k, r_\tau C'_k)}{\sqrt{\sum_k A_k} \sqrt{\sum_k A'_k}}$$

That this is a distance between the families of matched curves $(C_1, \ldots, C_L)$ and $(C'_1, \ldots, C'_L)$ can be deduced from the results of [13]. We finally select the τ for which $d_{shape}(\tau)$ is minimal, and this forms our final evaluation of the similarity of the views.

8 Experiments

Our experiments use a small database of video color images of toy vehicles. The matching algorithm performs well in finding a good matching when such a matching exists. If the objects differ too much, this is detected by one of our three criteria above.

Each figure describes a matching and is organized as follows. The upper left and upper right pictures provide the contours of the original segmentations $\mathcal{R}$ and $\mathcal{R}'$ which are compared : they provide all the information which is used for comparison and matching. The lower left and lower right pictures provide the obtained matching : the associated regions have the same gey colour and are patched with the same number.

In figures 1 to 3, we compare different views of a truck. In figure 1, d_{shape} is quite large (about 0.4). This is due to the fact that regions labeled 3 includes the lower part of the truck in one segmentation, and not in the other.

Figures 4 and 5 compare different objects, and the difference is well detected.

Finally, figures 6 to 8 provide comparisons of views of a truck, with various degrees of segmentation.

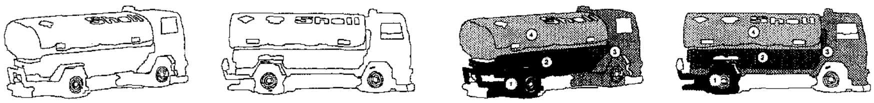

Fig. 1. Comparison of segmentations: truck 1 under different angles ; percentage of matched regions : 89.7% and 80.2%; d_{pos} = 0.15 radian (4 regions); d_{shape} = 0.42 radian

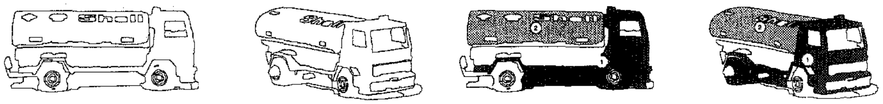

Fig. 2. Comparison of segmentations: truck 1 under different angles ; percentage of matched regions : 65.8% and 64%; d_{pos} = 0.1 radian (2 regions); d_{shape} = 0.35 radian

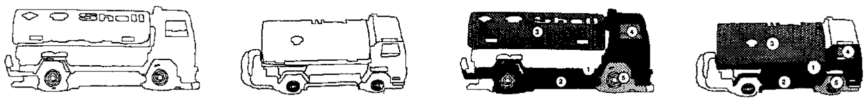

Fig. 3. Comparison of segmentations: truck 1 under different angles ; percentage of matched regions : 78% and 77.3%; $d_{pos} = 0.27$ radian (5 regions); $d_{shape} = 0.32$ radian

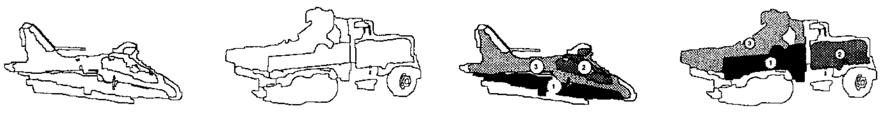

Fig. 4. Comparison of segmentations: plane and truck 2 ; percentage of matched regions : 75.3% and 57.9%; no computed distances

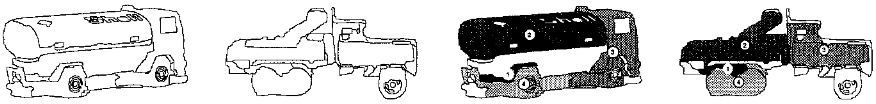

Fig. 5. Comparison of segmentations: truck 1 and truck 2 ; percentage of matched regions : 75.2% and 79.5%; $d_{pos} = 0.1$ radian (4 regions); $d_{shape} = 0.53$ radian

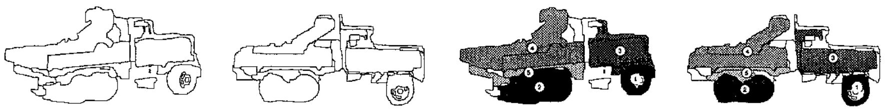

Fig. 6. Comparison of segmentations: truck 2 under different angles ; percentage of matched regions : 92% and 80.9%; $d_{pos} = 0.12$ radian (5 regions); $d_{shape} = 0.31$ radian

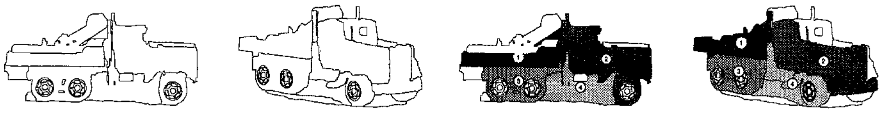

Fig. 7. Comparison of segmentations: truck 2 under different angles ; percentage of matched regions : 96.2% and 81.7%; $d_{pos} = 0.24$ radian (5 regions); $d_{shape} = 0.37$ radian

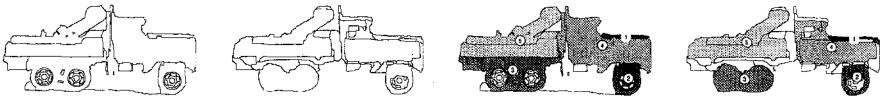

Fig. 8. Comparison of segmentations: truck 2 under the same angle (different segmentation) ; percentage of matched regions : 80% and 78.2%; $d_{pos} = 0.06$ radian (5 regions); $d_{shape} = 0.27$ radian

References

1. R. Azencott (1987): Image analysis and Markov random fields. Proc. of the int. Conf. on Ind. and Appl. Math. SIAM, Paris.
2. J. Ben-Arie and A. Z. Meiri (1987) : 3D Object recognition by optimal search of multinary relation graphs *Comp. Vis. Graph. Im. Proc.* 37, 345-361
3. R. Bergerin and M. Levine (1993) : Generic object recognition: building and matching coarse descriptions from line drawings *IEEE Trans. Pat. Anal. Mach. Intel.* Vol. 15, no 1, 19-36
4. O. Catoni (1990) Rough Large deviation estimates for simulated annealing. Application to exponential cooling schedules *Ann. of Proba.*, 20,1109-1146
5. W. E. L. Grimson and D. P. Huttenlocher (1991) : On the verification of hypothetized matches in model-based recognition *IEEE Trans. Pat. Anal. Mach. Intel.*Vol 13, no 12, 1201-1213
6. A. R. Pope (1994) : Model-based object recognition A survey of recent research Tech. Report 94- 04
7. E. Rivlin, S. J. Dickinson, A. Rosenfeld (1994) Recognition by functional parts. (preprint Center for automation research).
8. L. G. Shapiro (1980) A structural model of shape *IEEE Trans. Pat. Anal. Mach. Intel.* vol 2 no 2 111-126
9. L. G. Shapiro and R. M. Haralik (1981) : Structural description and inexact matching *IEEE Trans. Pat. Anal. Mach. Intel.* vol 3, no 5, 504-519
10. L. G. Shapiro and R. M. Haralik (1985) : A metric for comparing relational descriptors *IEEE Trans. Pat. Anal. Mach. Intel.* vol 7 no 1 90-98
11. N. Ueda and S. Suzuki (1993) : Learning visual models from shape contours using multi-scale convex/concave structure matching *IEEE Trans. Pat. Anal. Mach. Intel.* vol 15 no 4 337-351
12. B. Yang, W. E. Snyder and G. L. Bilbro (1989) : Matching over-segmented 3D images to models using association graphs *Image and vision comp.* Vol 7 no 2 135-143
13. L. Younes (1996) : Computable elastic distances between shapes (preprint).
14. S. Zhang, G. D. Sullivan, K. D. Baker (1993) : The automatic construction of a view independent relational model for 3-D object recognition *IEEE Trans. Pat. Anal. Mach. Intel.* Vol 15 no 6 531-544