



Hierarchical Matching Using Combinatorial Pyramid Framework

Luc Brun, Jean Hugues Pruvot

► To cite this version:

Luc Brun, Jean Hugues Pruvot. Hierarchical Matching Using Combinatorial Pyramid Framework. ICISP 2008, 2008, Cherbourg, France. pp.346-355, <10.1007/978-3-540-69905-7_40>. <hal-00820636>

HAL Id: hal-00820636

<https://hal.science/hal-00820636v1>

Submitted on 6 May 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Hierarchical Matching Using Combinatorial Pyramid Framework

Luc Brun and Jean-Hugues Pruvot

Université de Caen Basse-Normandie, GREYC CNRS UMR 6072, Image Team
6, Boulevard Maréchal Juin – 14050 Caen Cedex France
{luc.brun,jhpruvot}@greyc.ensicaen.fr

Abstract. A string matching approach is proposed to find a region correspondence between two images. Regions and their spatial relationships are represented by two combinatorial pyramids encoding two segmentation hierarchies. Our matching algorithm is decomposed in two steps: We first require that the features of the two matched regions be similar. This threshold on the similarity of the regions to be matched is used as a pruning step. We secondly require that at least one cut may be determined in each hierarchy such that the cyclic sequence of neighbors of the two matched regions have similar features. This distance is based on a circular string matching algorithm which uses both the orientability of the plane and the hierarchical encoding of the two regions to reduce the computational cost of the matching and enforce its robustness.

1 Introduction

Image correspondence plays a major role in many applications like image indexation, tracking or 3D reconstruction. The correspondence between both images is achieved by matching various primitives of dimension 0 (point), 1 (edge) or 2 (region). The detection of point and edge [1] primitives depends on local information and is thus sensible to noise and missing data. Moreover, the features attached to these primitives only provide local information. On the other hand, segmentation algorithms provide a partition of the image into regions, each region being associated to an area of the image on which complex photometric, geometrical and topological features may be computed.

Feature based region matching algorithms associate to each region a vector based on its photometric or geometric [2] features. Such algorithms neglect the neighbourhood of the two regions which encodes the context of the two regions being matched. These neighbourhoods may be taken into account by using inexact graph matching algorithms [3,4] on the graphs encoding the adjacency relationships between regions in both partitions. However, the application of these algorithms have been limited by the following two reasons:

1. An image, considered as a part of a plane, is an oriented surface. Moreover, the adjacency relationships between the regions of the partition define a planar graph. Most of graph matching algorithms neglect these two properties of a partition which constraint the matching and reduce its time complexity.

2. Any object within an image may be described at different scales. A matching algorithm may thus have to match one region of one image with several regions of the other image, both sets of regions describing a same object.

Wang [3] takes into account the planar structure of a partition by formulating the region correspondence into an inexact matching of two attributed planar graphs. He uses some properties of the planar graph and the topological relationships between regions to find a maximal inexact subgraph matching. However, this method doesn't take advantage of the plane orientation.

The orientation is used by Neuhauss [5] and Llados [4] which cast the problem of region correspondence between two images into a problem of circular string matching. The two strings being matched encode the boundaries of the matched regions. This explicit use of the orientation drastically reduces the combinatorial complexity of the match. However, both algorithms are sensible to an over segmentation of one of the two images which may split one of the region being matched or one of its neighbor into several sub regions.

This last drawback is addressed by the method proposed by Glantz [6]. This method builds an irregular pyramid from each image and creates one graph from the higher levels of each hierarchy. Each graph is built by connecting the planar graphs encoding the higher levels of the hierarchies by vertical edges. These edges encode the topological relationships between each vertex of one pyramid and its child in the level bellow. The use of hierarchical segmentation algorithms reduces the influence of the over/under segmentation problem and thus increases the number and the quality of the matches. However, the vertical links between the planar graphs destroy the planar structure of the two graphs being matched and thus forbids any use of the good properties of planar graphs. Moreover, the graphs deduced from each pyramid are defined from the connection of several planar graphs and are thus usually large.

The basic idea of our method is to combine the approaches of Glantz and Neuhauss or Llados. Following the method of Glantz, we first build two irregular pyramids on the two images to be matched. However, instead of building a large graph from each pyramid, we initiate a match by searching in the two hierarchies the couple of regions whose features are similar up to a threshold. Following the method of Neuhauss and Llados, the boundaries of two candidate regions are matched using a circular string matching algorithm. We take into account both hierarchies by considering the different rewritings of the boundary of each region in its hierarchy. The resulting algorithm is thus a circular string matching algorithm with rewriting rules. The two regions whose features are similar up to the threshold and whose boundary's distance is minimal are considered as good candidates for a larger match.

This paper describes the first step of this matching process which consists to select the two candidate regions for a global match. We first present the combinatorial pyramid framework used to encode both hierarchies in Section 2. The selection of the candidate regions based on their features and our circular string matching algorithm are presented in Section 3. Finally, experiments and results are discussed in Section 4.

2 The Combinatorial Pyramid Framework

A 2D combinatorial map may be understood as a particular encoding of a planar graph where each edge is split into two half-edges called darts. Since each edge of a graph connects two vertices, each dart belongs to only one edge and one vertex. A 2D combinatorial map is formally defined by the triplet $G = (\mathcal{D}, \sigma, \alpha)$ where \mathcal{D} represents the set of darts and σ is a permutation on \mathcal{D} whose cycles correspond to the sequence of darts encountered when turning counter-clockwise around each vertex. Finally α is an involution on \mathcal{D} which maps each of the two darts of one edge to the other one. Given a combinatorial map $G = (\mathcal{D}, \sigma, \alpha)$, its dual is defined by $\bar{G} = (\mathcal{D}, \varphi, \alpha)$ with $\varphi = \sigma \circ \alpha$. The cycles of permutation φ encode the faces of the combinatorial map. In what follows, the cycles of α , σ and φ containing a dart d will be respectively denoted by $\alpha^*(d)$, $\sigma^*(d)$ and $\varphi^*(d)$.

If a combinatorial map $G = (\mathcal{D}, \sigma, \alpha)$ is used to encode a 2D partition, each boundary between two regions is encoded by an edge. Such an edge (say $\alpha^*(d) = (d, \alpha(d))$) encodes an adjacency relationship between $\sigma^*(d)$ and $\sigma^*(\alpha(d))$. The two darts d and $\alpha(d)$ encode the two opposite orientations along the boundary of $\alpha^*(d)$. A σ -cycle $\sigma^*(d) = (d_1, \dots, d_n)$ may thus be understood in two different ways:

- Firstly, since each dart encodes an adjacency relationship, the sequence $(d_i)_{i \in \{1, \dots, n\}}$ determines the cyclic sequence $(\sigma^*(\alpha(d_1)), \dots, \sigma^*(\alpha(d_n)))$ of vertices encountered when turning counter-clockwise around $\sigma^*(d)$. Within a segmentation scheme, this sequence of vertices corresponds to the sequence of regions encountered when turning counter-clockwise around the central region defined by $\sigma^*(d)$.
- Secondly, if we consider that each dart encodes an oriented boundary between two regions, the sequence $\sigma^*(d) = (d_1, \dots, d_n)$ may be considered as the concatenation of the oriented boundaries associated to the darts $(d_i)_{i \in \{1, \dots, n\}}$. This concatenation encodes a counter-clockwise traversal of the boundary of the region encoded by $\sigma^*(d)$.

For example, the two darts of the edge $\alpha_2^*(5) = (5, -9)$ in Fig. 1 (c) encode two opposite orientations of the boundary between the regions R_1 and R_2 respectively encoded by the vertices $\sigma_2^*(5) = (5, 16)$ and $\sigma_2^*(-9) = (-9, 20)$. We may associate to the vertex $\sigma_2^*(5)$ the sequence $(\sigma_2^*(9), \sigma_2^*(-21))$ encoding the sequence of regions encountered when turning counter-clockwise around R_1 . Note that in this example, $\sigma_2^*(-21)$ encodes the background of the image.

A combinatorial pyramid is defined by an initial combinatorial map successively reduced by a sequence of contraction or removal operations. Contraction operations are encoded by contraction kernels. A contraction kernel is defined as a forest of the current combinatorial map, each tree of this forest encodes the contraction of a connected set of vertices into a single vertex of the reduced combinatorial map. Contraction kernels may create redundant edges such as empty-self loops and double edges. These redundant edges are removed by

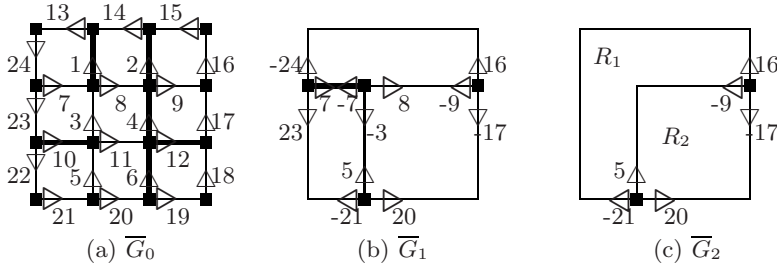


Fig. 1. The first three dual combinatorial map $\overline{G}_0, \overline{G}_1, \overline{G}_2$ of a combinatorial pyramid. The initial map G_0 encodes a 3×3 grid of pixels. Contracted darts are represented by bold lines in (a) and (b).

using respectively empty self loop and double edge removal kernels [7]. These removal kernels are applied successively and are defined as a forest of the dual current combinatorial map. Further details about the construction scheme of a combinatorial pyramid may be found in [7].

A combinatorial pyramid may thus be encoded by a sequence of successively reduced combinatorial maps $P = (G_0, \dots, G_n)$ where G_0 encodes the 4 connected grid of pixels or any other initial partition. Let us consider a level $G_i = (\mathcal{D}_i, \sigma_i, \alpha_i)$ of the pyramid P and a dart d belonging to \mathcal{D}_i . The receptive field of the dart d in G_0 is called the connecting dart sequence of d . Intuitively, this sequence encodes the darts of G_0 combined between levels 0 and i to form the dart d at level i . Since d encodes an oriented boundary between two regions at level i , the embedding of this boundary within G_0 may be retrieved from the connecting dart sequence of d . This sequence called the sequence of boundary darts of d at level i and denoted $SBD_i(d) = d_1 \dots d_p$ is defined by:

$$d_1 = d, d_{j+1} = \varphi_0^{m_j}(\alpha_0(d_j)) \text{ and } \alpha_0(d_p) = \alpha_i(d). \quad (1)$$

where $\overline{G}_0 = (\mathcal{D}_0, \varphi_0, \alpha_0)$ is the dual of the initial combinatorial map and m_j is the minimal integer q such that $\varphi_0^q(\alpha_0(d_j))$ survives at level i or belongs to a double edge kernel. This last condition is tested in constant time using the implicit encoding of combinatorial pyramids [7].

The sequence of boundary darts of the dart 20 at level 2 in Fig. 1 (c), is for example equal to $SBD_2(20) = 20.19.18.17$ (Fig. 1 (a)). The whole boundary of the vertex $\sigma_2^*(20) = (20, -9)$, encoding the four bottom-right pixels of the 3×3 grid at level 2, is defined as $SBD_2(20)SBD_2(-9) = 20.19.18.17. -9. -8. -3. -5$.

3 Hierarchical Matching

Let us consider two pyramids $P = (G_0, \dots, G_N)$ and $P' = (G'_0, \dots, G'_{N'})$. Each combinatorial map G_i of P contains a set of vertices. We denote by \mathcal{V}_P the union of all the vertices of P defined between levels 0 and N . In the same way, the set of vertices of P' is denoted $\mathcal{V}_{P'}$. As mentioned in Section 1, our hierarchical matching algorithm aims at finding two vertices in \mathcal{V}_P and $\mathcal{V}_{P'}$ encoding regions

with close features and a minimal distance between their neighbourhoods. We thus want to determine the couple of vertices $(v, v') \in \mathcal{V}_P \times \mathcal{V}_{P'}$ such that:

$$(v, v') = \underset{(w, w') \in \mathcal{C}}{\operatorname{argmin}} \Delta_N(w, w') \text{ with} \quad (2)$$

$$\mathcal{C} = \{(w, w') \in \mathcal{V}_P \times \mathcal{V}_{P'} \mid F(w, w') = \text{true}\} \quad (3)$$

where $\Delta_N(v, v')$ represents the distances between the neighbourhoods of v and v' , and $F(w, w')$ is a boolean condition on w and w' .

The set \mathcal{C} represents our set of candidates for the match. Within our framework Equation 3 is used as a filtering step. The advantages of this filtering step are twofold: Firstly, our criterion F insures that two matched vertices have similar features independently of the distance between their neighbourhoods. Secondly, this filtering reduces the number of couples of vertices on which Equation 2 will be evaluated. Our filtering step and our neighbourhood's distance are respectively described in Section 3.1 and Section 3.2.

3.1 Filtering

Let us associate to each vertex v of \mathcal{V}_P a vector $Feat(v)$ of M features. Let us suppose that v is defined at level $i \geq 1$ in P and let us consider the set of vertices $RW_i(v) = \{v_1, \dots, v_p\}$ whose contraction in G_{i-1} defines v at level i . The set $RW_i(v)$, called the reduction window of v , is defined as the set of vertices incident to one of the trees of the contraction kernel which builds G_i from G_{i-1} (Section 2). The function *Father* is defined on \mathcal{V}_P as $Father(w) = v$ for any $w \in RW_i(v)$. By convention any vertex of the top level combinatorial map is its own father. We say that one coordinate j of our vector of feature is increasing iff for any vertex $v \in \mathcal{V}_P$ defined at level $i \geq 1$ in P we have:

$$\forall w \in RW_i(v), \quad Feat_j(v) \geq Feat_j(w)$$

Note that any cumulative moment defines an increasing feature along the pyramid.

Let us suppose that the first coordinate $Feat_1(v)$ of our vector of features is increasing. We first reduce the number of candidates for the match by selecting the set of couples $(v, v') \in \mathcal{V}_P \mathcal{V}_{P'}$ such that:

$$\Delta_{F_1}(v, v') = |1 - \frac{Feat_1(v')}{Feat_1(v)}| \leq \epsilon_1 \quad (4)$$

where ϵ_1 is an user defined threshold.

For each vertex $v \in \mathcal{V}_P$ this first filtering step is equivalent to select all vertices $v' \in \mathcal{V}_{P'}$ such that $Feat_1(v') \in [(1 - \epsilon_1)Feat_1(v), (1 + \epsilon_1)Feat_1(v)]$. This filtering step may be achieved efficiently using the increasing property of the first feature's coordinate and the hierarchical relationships of P' encoded by the reduction window and father functions.

The first filtering step defined by equation 4 reduces the number of potential candidates for the match based on only one feature. We refine this first filtering

by using the remaining features of each vertex. However, since the different features associated to a vertex vary within different intervals, a normalisation step is required before computing the distance between the features of two vertices. Given two vertices v and v' we define the distance between $Feat(v)$ and $Feat(v')$ as the infinite norm of a vector f defined as:

$$\forall i \in \{1, \dots, M\} \quad f_i(v, v') = 1 - \frac{\min(Feat_i(v), Feat_i(v'))}{\max(Feat_i(v), Feat_i(v'))} \quad (5)$$

Our final boolean criterion encoding the set \mathcal{C} of filtered couple of vertices (equation 3) is thus equal to:

$$F(w, w') = \Delta_{F_1}(v, v') \leq \epsilon_1 \text{ and } \|f\|_\infty \leq \epsilon_1.$$

Both tests being applied sequentially.

Note that if the first feature $Feat_1(v)$ is positive then: $\Delta_{F_1}(v, v') \leq \epsilon_1$ implies that $f_1(v, v') \leq \epsilon_1$. Our first filtering step ($\Delta_{F_1}(v, v') \leq \epsilon_1$) may in this case be interpreted as a restriction of our second test ($\|f\|_\infty \leq \epsilon_1$) to the first coordinate of f . Such an interpretation is loss if we use the Euclidean norm rather than the infinite one. The Euclidean norm may however be preferred if one wants to allow a phenomenon of compensation between the feature's distances f_i .

3.2 Distance between Hierarchical Neighbourhoods

Given a pyramid P , let us consider a vertex $v \in \mathcal{V}_P$ which survives up to level l_v . Let us denote by $\sigma_{l_v}^*(d) = (d_1, \dots, d_q)$ the σ -cycle of G_{l_v} associated to v . The embedding of the boundary of v within G_0 , denoted B_v is defined as the concatenation of the sequence of boundary darts $(SBD_{l_v}(d_j))_{j \in \{1, \dots, q\}}$ (Section 2). Since all the darts of a sequence of boundary darts belong to $G_0 = (\mathcal{D}_0, \sigma_0, \alpha_0)$, any vertex of the hierarchy may be considered as a word built on the alphabet \mathcal{D}_0 . The neighbourhood of v in G_0 is defined as the set of vertices of G_0 adjacent to the eventual over-segmentation of v . This set, denoted $N^0(v)$ may be formally defined by: $N^0(v) = \{\sigma_0^*(\alpha_0(d)), d \in B_v\}$.

Let us consider two pyramids P, P' and two vertices $(v, v') \in \mathcal{V}_P \times \mathcal{V}_{P'}$ with $B_v = d_1 \dots d_n$ and $B_{v'} = d'_1 \dots d'_{n'}$. Using the orientation of $N^0(v)$ and $N^0(v')$ in G_0 and G'_0 , the neighbourhood's distance between v and v' at level 0 is defined as the edit distance between B_v and $B_{v'}$. If q denotes the number of symbols matched between B_v and $B_{v'}$, the total cost of the match between v and v' is defined as:

$$\Delta(B_v, B_{v'}) = \sum_{i=0}^q \delta(d_{\phi_1(i)}, d'_{\phi_2(i)}) + (n + n' - 2q)K \quad (6)$$

where $\delta(\cdot)$ is a distance function between darts and K is the default cost for a removal operation within either B_v or $B_{v'}$. The values of $\phi_1(i)$ and $\phi_2(i)$ correspond to the indexes within B_v and $B_{v'}$ of the two i^{th} darts being matched.

As mentioned in Section 2, each dart of a combinatorial map may be interpreted both as an encoding of the adjacency between two regions and as a boundary between the same two regions. Therefore, if we consider two darts $d \in G_i$ and $d' \in G'_j$ of P and P' , the distance $\delta(d, d')$ between these two darts may incorporate features of the vertices $\{\sigma_i^*(d), \sigma_i^*(\alpha_i(d)), \sigma_j'^*(d'), \sigma_j'^*(\alpha_j(d'))\}$ and geometrical features of the boundaries associated to the darts d and d' at levels i and j . Within our framework, since the distance between the features of v and v' has been tested during the filtering step we define $\delta(d, d')$ from the features of the vertices $\sigma_i^*(\alpha_i(d))$, $\sigma_j'^*(\alpha_j(d'))$ and the features of the two boundaries encoded by d and d' at level i and j (Section 4).

All the darts of the sequences B_v and $B_{v'}$ are defined in G_0 . Therefore, the distances $\delta(.,.)$ in Equation 6 are evaluated at level 0 and the whole comparison of the neighbourhoods of v and v' is made in G_0 and G'_0 . As mentioned in Section 1, the neighbourhoods of v and v' may not correspond within the two initial partitions encoded by G_0 and G'_0 . However, given the two pyramids P and P' respectively built on G_0 and G'_0 , we may consider the set of cuts of P (resp. P') which pass below v (resp. v'), i.e. such that v (resp. v') is either a single region or is over-segmented within the cut. The basic assumption of this paper is that if v and v' correspond to a same “ideal” region there should be at least one such cut for each pyramid in which the neighbourhoods of v and v' are similar.

Let us consider a sub sequence $d_l \dots d_m$ of B_v such that the set of vertices $\{\sigma_0^*(\alpha_0(d_l)), \dots, \sigma_0^*(\alpha_0(d_m))\}$ is merged into a single vertex at a level $l_1 \leq l_v$. All the darts encoding the adjacency relationships between $\{\sigma_0^*(\alpha_0(d_l)), \dots, \sigma_0^*(\alpha_0(d_m))\}$ should thus be either contracted or removed as empty self loop at level l_1 . Moreover, the set of vertices $\{\sigma_0^*(d_l), \dots, \sigma_0^*(d_m)\}$ defines an over-segmentation of v and should be merged into a single vertex at a level $l_2 \leq l_v$. If we consider the sequence $d_l \dots d_m$ at the level $L = \max(l_1, l_2)$, the darts $d_l \dots d_m$ encode a contiguous sequence of boundaries between two regions and may thus be interpreted as double edges [8]. These darts should thus belong to the sequence of boundary dart $SBD_L(d_k)$ of a dart $d_k \in B_v$. Conversely, a sequence of boundary darts $SBD_L(d_k) = d_l \dots d_m$ with $L \leq l_v$ and $d_k \in B_v$ encodes a sequence of double edge darts which may be produced only by the merge of the vertices $\{\sigma_0^*(d_l), \dots, \sigma_0^*(d_m)\}$ and $\{\sigma_0^*(\alpha_0(d_l)), \dots, \sigma_0^*(\alpha_0(d_m))\}$. The sequence of boundary darts allows thus to retrieve all the merge operations between the vertices of $N^0(v)$ defined before the level l_v . Since two cuts producing the same merge operations on $N^0(v)$ produce the same local configuration in the neighbourhood of v , the sequence of boundary darts $SBD_i(d)$ with $i \leq l_v$ and $d \in B_v$ allows us to retrieve all the neighbourhoods of v produced by the different cuts of P .

For any sequence of boundary dart $SBD_i(d_j) = d_j \dots d_k$ with $i \leq l_v$ and $d_j \in B_v$ let us consider the rewriting rule $\psi_{j,k}^i : d_j \dots d_k \mapsto d_j$ which replaces in B_v , the sequence of darts $d_j \dots d_k$ defined in G_0 by the single dart d_j considered as a dart of G_i . In order to avoid confusions between the dart d_j considered in G_0 and in G_i , we use the same notation for the name of the rule $\psi_{j,k}^i$ and its result d_j

in G_i . We have thus: $\psi_{j,k}^i : d_j \dots d_k \mapsto \psi_{j,k}^i$. From a geometrical point of view, the oriented boundary encoded by the dart $\psi_{j,k}^i$ is defined as the concatenation of the oriented boundaries encoded by $d_j \dots d_k$ at level 0. Let us consider the set Σ of such rules defined on B_v and $\Sigma(B_v)$ the set of rewritings of B_v . In the same way, let us consider the set of rewritings of $B_{v'}$ denoted as $\Sigma'(B_{v'})$. Both $\Sigma(B_v)$ and $\Sigma'(B_{v'})$ encode the neighbourhoods of v and v' among the different cuts of P and P' . The neighbourhood's distance between v and v' is thus defined as:

$$\Delta_N(v, v') = \min_{(m, m') \in \text{Rot}(\Sigma(B_v)) \times \Sigma'(B_{v'})} \Delta(m, m')$$

where Δ is defined according to Equation 6 and $\text{Rot}(\Sigma(B_v))$ represents all the circular permutations of the strings contained in $\Sigma(B_v)$.

Given a rotation, such a distance may be computed efficiently [9] using the following recursive equation:

$$\Delta(0, 0) = 0, \text{ and } \Delta(i, j) = \min \left(\begin{array}{l} \Delta(i-1, j-1) + \delta(d_i, d_j), \\ \min_{\substack{\psi_{k,i}^l \in \Sigma, \psi_{k',j}^{l'} \in \Sigma'}} \Delta(k-1, k'-1) + \delta(\psi_{k,i}^l, \psi_{k',j}^{l'}), \\ \Delta(i-1, j) + K, \\ \Delta(i, j-1) + K \end{array} \right)$$

where $\psi_{k,i}^l$ and $\psi_{k',j}^{l'}$ denote two rewriting rules of Σ and Σ' . The indexes k and k' are respectively lower than i and j and the levels l and l' must be respectively lower than l_v and $l_{v'}$. Note that the distance $\delta(d_i, d_j)$ in the above equation is evaluated at level 0.

4 Experiments

Given a pyramid P and a vertex $v \in \mathcal{V}_P$, let us denote by R_v the geometrical region associated to v and by ∂R_v the boundary of R_v . The vector of features $\text{Feat}(v)$ (Section 3.1) used in our experiments is composed of $M = 6$ features.

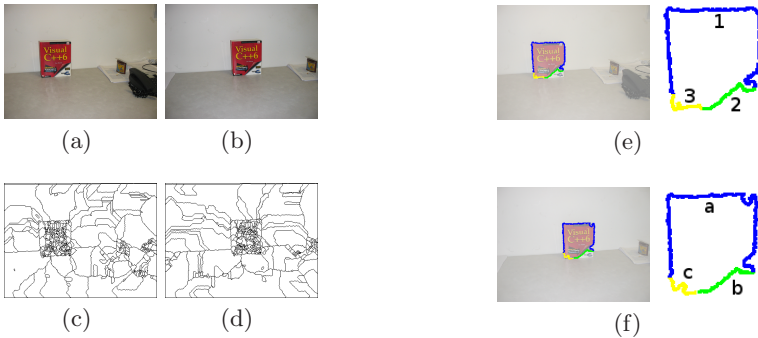


Fig. 2. Two matched boundaries (e) and (f) determined from two pyramids obtained from the initial partitions (c) and (d) and the images (a) and (b).

$Feat_1(v)$ encodes the size of R_v while $Feat_2(v)$ to $Feat_4(v)$ encode the three colour channels of the mean colour of R_v . The two last features $Feat_5(v)$ and $Feat_6(v)$ respectively encode the size of ∂R_v and the mean value of the colour gradient computed along it.

Given a dart d defined at level i , we characterise the shape of the oriented boundary associated to d by a function which maps each discrete point of the boundary to the curvature [10] of the boundary at this point. We obtain thus a vector of curvature points that we compress into 8 features using the Legendre's moments [11]. We add to these 8 geometrical features 3 colour features corresponding to the mean colour of the vertex $\sigma_i^*(\alpha_i(d))$ (Section 3.2). The distance function δ (Section 3.2) between two darts is then defined as the Euclidean norm of a vector f using a normalisation step equivalent to the one used in Equation 5 but applied on the features of the two darts.

One result of our algorithm on a couple of real images representing a same scene with different view points is presented in Fig. 2. The base level combinatorial maps of the two pyramids $P = (G_0, \dots, G_n)$ and $P' = (G'_0, \dots, G'_{n'})$ encode a watershed of the input images [12]. The construction scheme of the remaining levels of the pyramids is described in [13]. The two matched vertices v and v' encode the upper left part of the book in both images. Using our rewriting rules, the sequences B_v and $B_{v'}$ encoding the shape of the book in the base level combinatorial maps (Fig. 2 (c), (d)) have been grouped into only 3 darts (Fig. 2 (e), (f)): The darts, separating the book and the wall ($1 \leftrightarrow a$), the white and the red part of the book ($2 \leftrightarrow b$) and the book and the table ($3 \leftrightarrow c$). All these darts are matched at different levels in the pyramids P and P' . For example, the dart 1 belongs to the 838^{th} level of P while a belong to the 961^{th} level of P' .

Fig. 3 illustrates the behavior of our algorithm on a more complex example. The sequences B_v and $B_{v'}$ encoding the shape of the cars in the base level combinatorial maps (Fig. 3 (c), (d)) have been grouped into 4 darts (Fig. 3 (e), (f)). We used the same covention for the matches than in Figure 2: (a,1),(b,2)...

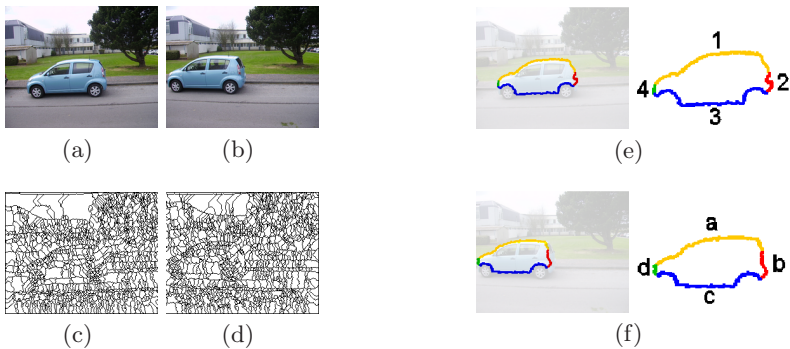


Fig. 3. Two matched boundaries (e) and (f) determined from two pyramids obtained from the initial partitions (c) and (d) and the images (a) and (b).

5 Conclusion

We have presented in this paper a method which matches two similar regions R and R' in two hierarchies P and P' . This method uses the features of R and R' and their oriented neighbourhoods. The distance between the oriented neighbourhoods of R and R' is defined as the minimal distance between the neighbourhoods of R and R' in all the cuts of P and P' which include them. The result of this new notion of similarity is a match between two regions together with a grouping of their boundary's basic elements into larger groups. The resulting boundaries are finally aligned. This last features should allows us to expand the match in order to find the largest common sub-map, within two hierarchies.

References

1. Kumar, S., Sallam, M., Goldgof, D.: Matching point features under small nonrigid motion. *Pattern Recognition* 34, 2353–2365 (2001)
2. Kaygin, S., Bulut, M.M.: Shape recognition using attributed string matching with polygons vertices as the primitives. *Pattern Recognition Letters* 23, 287–294 (2002)
3. Caihua, W., Keiichi, A.: Region correspondence by inexact attributed planar graph matching. In: 5th International Conference on Computer Vision, June 1995, pp. 440–447 (1995)
4. Llados, J., Marti, E., Villanueva, J.: Symbol recognition by error-tolerant subgraph matching between region adjacency graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23, 1137–1143 (2001)
5. Neuhaus, M., Bunke, H.: An error-tolerant approximate matching algorithm for attributed planar graphs and its application to fingerprint classification. In: Fred, A., Caelli, L.M., Duin, R.P.W., Campilho, A.C., de Ridder, D. (eds.) *SSPR&SPR 2004*. LNCS, vol. 3138, pp. 180–189. Springer, Heidelberg (2004)
6. Glantz, R., Pelillo, M., Kropatsch, W.G.: Hierarchical matching of panoramic images. In: *ICIAP 2003*, p. 328. IEEE Computer Society, Los Alamitos (2003)
7. Brun, L., Kropatsch, W.: Construction of combinatorial pyramids. In: Hancock, E., Vento, M. (eds.) *GbrPR 2003*. LNCS, vol. 2726, pp. 1–12. Springer, Heidelberg (2003)
8. Brun, L.: *Traitement d'images couleur et pyramides combinatoires*. Habilitation à diriger des recherches, Université de Reims (2002)
9. Gdalyahu, Y., Weinshall, D.: Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21, 1312–1328 (1999)
10. Malgouyres, R., Brunet, F., Fourey, S.: Binomial convolutions and derivatives estimation from noisy discretizations. In: Coeurjolly, D. (ed.) *Proceedings of DGCI 2008*. LNCS, vol. 4992, Springer, Heidelberg (2008)
11. Shen, J., Shen, D.: Orthogonal legendre moments and their calculation. In: *ICPR 1996*, vol. 2, p. 241 (1996)
12. Brun, L., Mokhtari, M., Meyer, F.: Hierarchical watersheds within the combinatorial pyramid framework. In: Andrès, É., Damiand, G., Lienhardt, P. (eds.) *DGCI 2005*. LNCS, vol. 3429, pp. 34–44. Springer, Heidelberg (2005)
13. Pruvot, J.H., Brun, L.: Scale set representation for image segmentation. In: Escolano, F., Vento, M. (eds.) *GbrPR*. LNCS, vol. 4538, pp. 126–137. Springer, Heidelberg (2007)