

Optimal approach for fast object-template matching

András Hajdu, and Ioannis Pitas*, *Senior Member, IEEE*

Department of Informatics

Aristotle University of Thessaloniki

Box 451, 54124 Thessaloniki, Greece

Tel: +30-231-099-6361

Fax: +30-231-099-8453

e-mail: {hajdua, pitas@aia.csd.auth.gr}

Abstract—This paper proposes a novel algorithm for an optimal reduction of object description for object matching purposes. Our aim is to decrease the computation needs by considering simplified objects, thus reducing the number of pixels involved in the matching process. We develop the appropriate theoretical background based on centroidal Voronoi tessellations. Its use within the chamfer matching framework is also discussed. We present experimental results regarding the performance of this approach for 2D contour and region-like object matching. As a special case, we investigate how the snake based representation of target objects can be employed in chamfer matching. The experimental results concern the use of object part matching for recognizing humans and show how the proposed simplification leads to valid replacements of the original templates.

Index Terms—chamfer matching, centroidal Voronoi tessellation, distance transformation, object simplification, object recognition.

EDICS Category: MOD-MRPH, SEG-MDFT

I. INTRODUCTION

OBJECT detection and classification have been challenging problems in digital image analysis and computer vision. Usually, some image features (e.g. edges) are extracted and then a matching procedure is applied to find the occurrences of pre-defined templates. Chamfer matching is one of these methods [1]. Here, usually an edge detection step is applied first to obtain a binary image, for which a distance map is calculated [2], [3]. If there are no additional possibilities (e.g. background subtraction [4], [5], intensity-based pre-segmentation [6]) to restrict the search range, a scanning step is applied to the entire image to match pre-defined templates, like in [2], [7].

Chamfer matching is a popular method in real-time object matching applications. Several supplementary approaches have been recommended to speed up its computation. One of them is to simplify the description of the objects to be matched by representing them by a smaller number of points. This can be achieved by a coarse-to-fine approach [8] regarding object resolution, or by selecting some significant points on the object boundary. For the latter case, a natural consideration was proposed in [9], by selecting points with high curvature (e.g. corners). However, actual optimization of such a selection was not given.

In this paper, we propose a novel way of simplifying of the object description for faster matching. Our approach, which

is based on centroidal Voronoi tessellation (CVT) methods can be applied to any dimensions [10], [11]. Furthermore, we show that it is optimal for chamfer matching purposes. We prove some fundamental statements regarding the proposed approach and recommend corresponding algorithms to obtain the optimal set of points.

In one of our human person detection/tracking/recognition systems being currently developed, we use snakes to describe the target objects to be recognized. Recognition is performed by matching contour templates from a database representing whole objects or object parts. Consequently, we shall present our experimental studies for the classical application setup of chamfer matching, when contour object descriptions are matched. Moreover, we make some comments on the natural extension of our approach to any other kind of object representation (e.g. region-like ones). Our simplification approach can be applied in two ways in the matching process. On one hand, the database templates can be simplified for faster performance. On the other hand, we can consider a simpler description of the target objects. For example, in the case of snakes, we can iterate with a smaller number of snake points. Furthermore, we investigate how the matching accuracy drops, when the database templates are simplified and also how the number of snake points affects the reliability of the matching. We also compare our technique with naive representation reduction approaches.

The structure of the paper is as follows. In section II we describe chamfer matching. Then, section III explains how the objects to be matched can be simplified optimally for chamfer matching based on a CVT approach. We also present efficient algorithms for this simplification. Section IV contains our experimental results on the proposed simplification. Here, we also explain how snake representation speeds up chamfer matching techniques and we illustrate the efficiency of the simplification in a simple region-based object matching approach. We end up with conclusions in section V.

II. CHAMFER MATCHING

Chamfer matching was proposed in [1] for measuring the distance between binary images. While Hamming or other distances are used on exactly matched pixel pairs only, chamfer matching is a more robust approach for a less accurate matching. Let S be a binary image ($S \subseteq \mathbb{Z}^m$), and $T = \{\mathbf{t}_i \in$

$\mathbb{Z}^m \mid i = 1, \dots, K; m \in \mathbb{N}; \mathbf{t}_1 = \mathbf{O}$ be a binary template consisting of K ordered points starting at the origin \mathbf{O} of \mathbb{Z}^m . The distance map of S is defined as $d_S : \mathbb{Z}^m \rightarrow \mathbb{R}_{\geq 0}$, such that, for any $\mathbf{x} \in \mathbb{Z}^m$, $d_S(\mathbf{x})$ is the distance of \mathbf{x} and the closest element of S . By definition, $d_S(\mathbf{x}) = 0$ for all $\mathbf{x} \in S$. To create distance maps, either the Euclidean distance [12], [13], or city-block/chessboard/chamfer distances [14], [15] can be used for faster implementation. Moreover, we can find robust variants of distance maps that weigh distance values, like the truncated linear/quadratic [7], [16], or the exponentially decreasing one [17] in order to concentrate more on the regions closer to the edges. Throughout the paper, notation d will refer to the Euclidean distance function.

When the distance map d_S is obtained, a natural way to determine how good the fit is between S and T at $\mathbf{x} \in \mathbb{Z}^m$ is to consider the chamfer distance:

$$d_{\mathbf{x}}(S, T) = \frac{1}{K} \sum_{i=1}^K d_S(\mathbf{x} + \mathbf{t}_i). \quad (1)$$

Obviously, smaller $d_{\mathbf{x}}$ values correspond to better fit between T and S at \mathbf{x} . To get an impression of the basic steps of chamfer matching, see Figure 1.

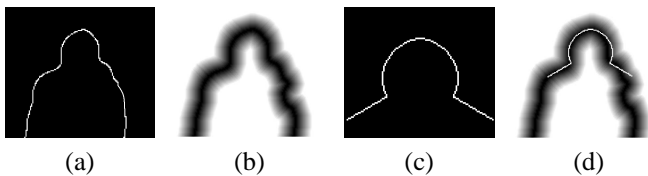


Fig. 1. Basic steps of chamfer matching; (a) extracted binary contour S , (b) the distance map d_S of S , (c) a template T (magnified) to be matched against S , (d) finding the best matching position of T on d_S .

Besides the arithmetic mean in (1), one can use the root mean square average, the median or the maximum of the corresponding distance values, as well [18]. Moreover, to suppress the influence of outliers (due to noise) or missing data (due to occlusion or segmentation errors), the α -trimmed mean (where the large distance values are excluded from the summation) or the truncated mean (where a global threshold is applied for the distance map) can be employed [7], [18], [19]. Note that the chamfer distance is in close relation to the Hausdorff distance [7]. In our experiments, we considered the 2D case, and used the $\langle 3, 4 \rangle$ distance map proposed in [15], which is fast to generate. However, as a comparative analysis shows, other distance maps lead basically to the same results with our approach.

Usually, the target object goes under some geometric transformation (e.g. translation, rotation, scaling) in the image. Thus, in the worst case, we have to search for the optimal parameters for translation, rotation, magnification, that is, for an optimal affine transformation. A direct search for all these transformation parameters is very time-consuming. Thus, algorithmic speed ups have been proposed in the literature. Borgefors [8] suggested a hierarchical edge matching based on a "coarse to fine" spatial resolution approach. However, Huttenlocher et al. [7] warned that this method has some risk of losing information when reducing resolution. They

suggested searching the transformation space using a cell decomposition (divide and conquer) strategy [19]. This method is based on the fact that if a bad match is found for a given transformation, then similar transformations (being close in the parameter space) can be excluded, as well. This idea has become popular in applications. However, as it still needs quite many computations or parallelization, the transformation space is often reduced, like e.g. in [2], using an a priori estimation of the scale parameter.

III. REDUCING THE NUMBER OF OBJECT POINTS INVOLVED IN CHAMFER MATCHING

As mentioned in the introduction, there are two ways to reduce the number of points involved in the chamfer matching procedure. We can adjust both the density of the points describing the target objects and the database templates. The database templates can be reduced in a pre-processing step, as it is independent from the feature extraction procedure. We should remove points from the template in an optimal way, with respect to the distance measure used. To set up our optimality criterion we also note that, in chamfer matching, the target object and the template to be matched can be exchanged. In other words, we follow the same procedure when matching either the template against the target object or the target object against the template. Thus, any optimization approach for the target object must also be valid for the template, and vice versa. It is already clear from our previous discussions that, if the target object description is simplified, the generated distance map should remain similar to the original one. Now, by the invertibility of the matching, to simplify the template, we should keep those points that generate a distance map, which is "close" to the one generated by the original template. Similar goals are set in [9], but no actual optimization is given.

As it was discussed in section II, to solve this simplification problem, there must be a search region defined within which the distance maps are compared. An obvious search region can be a bounding box or some dilated version of the original template. More precisely, we solve the following problem: Reduce a discrete set $A \subseteq B$ with cardinality $|B| = M$ to A' with $A' \subseteq A$, and $|A'| = K \leq M$, so that the distance map generated by A' is closest to the distance map generated by A within B . Figure 2 shows an example [20].

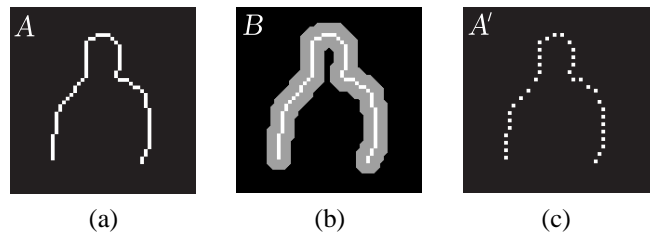


Fig. 2. Simplification of an object for fast matching by keeping its distance map close to the original one; (a) object A to be simplified, (b) the region B (in gray) within which the distance map should be preserved, (c) the simplified object A' .

Distance maps can be compared simply by summing up the differences between their corresponding values within B . The following lemma shows that this comparison can also be

performed by comparing the sum of the values of the distance maps generated by the reduced variants of A .

Lemma 3.1: The discrete set A' is an optimal reduction of A , if we have:

$$\sum_{\mathbf{x} \in B} d_{A'}(\mathbf{x}) = \min_{\substack{A' \subseteq A \\ |A'|=K}} \left(\sum_{\mathbf{x} \in B} d_A(\mathbf{x}) \right). \quad (2)$$

Proof: If we remove points from A , the corresponding distance values in the distance map generated by the remaining subset will be greater than in d_A . Thus for any $A' \subseteq A$ with $|A'| = K$,

$$\sum_{\mathbf{x} \in B} |d_{A'}(\mathbf{x}) - d_A(\mathbf{x})| = \sum_{\mathbf{x} \in B} d_{A'}(\mathbf{x}) - \sum_{\mathbf{x} \in B} d_A(\mathbf{x}) \geq \sum_{\mathbf{x} \in B} d_{A'}(\mathbf{x}) - \sum_{\mathbf{x} \in B} d_A(\mathbf{x}) \quad (3)$$

holds, if A' is defined according to (2). ■

The solution of the previous problem could obviously be found by a very expensive "brute force" algorithm, by checking $\binom{N}{K}$ cases, assuming that $|A| = N$. To find a more robust solution for this problem, let us re-formulate it in the Euclidean space. For a more general setup, we also define a weight function ϱ on B , to keep the relative importance of the different parts of A adjustable for the simplification process. Such weight functions can be probability kernels, like the uniform or Gaussian ones. As the distance values within a distance map are based on the closest points of the generator object, the problem can be interpreted in the continuous case using the elements of the Voronoi tessellation framework:

Let $A \subseteq B \subseteq \mathbb{R}^m$, such that A is compact and convex and B is bounded. Moreover, let $K \in \mathbb{N}$ and $\varrho : B \rightarrow \mathbb{R}_{\geq 0}$. Find the set of points $A' = \{\mathbf{y}_i \in A \mid i = 1, \dots, K\}$, which minimizes:

$$\sum_{i=1}^K \int_{V_i(A')} \varrho(\mathbf{y}) |\mathbf{y} - \mathbf{y}_i|^2 d\mathbf{y}, \quad (4)$$

where $\{V_i(A')\}_{i=1}^K$ denotes the Voronoi tessellation of B generated by A' .

Though it is already suggested in Lemma 3.1, the subsequent construction of the solution to problem (4) will clarify the equivalence of the two problems above. For simplicity, A^K will denote the Cartesian product $\underbrace{A \times \dots \times A}_{K \text{ times}}$. The following

lemma explains why A needs to be convex and compact, while B must obviously be bounded, so that the integral has a finite value in (4).

Lemma 3.2: Let $A \subseteq \mathbb{R}^m$ be compact and convex, and let $\mathbf{x} \in \mathbb{R}^m$ be arbitrarily chosen. Then, there exists a unique $\mathbf{y} \in A$, such that $d(\mathbf{x}, \mathbf{y}) = \min_{\mathbf{z} \in A} d(\mathbf{x}, \mathbf{z})$.

Proof: The compactness (closedness) of A obviously implies the existence of such a $\mathbf{y} \in A$. Now, to show that \mathbf{y} is unique, let us assume that there exist $\mathbf{y}_1, \mathbf{y}_2 \in A$, $\mathbf{y}_1 \neq \mathbf{y}_2$ such that $d(\mathbf{x}, \mathbf{y}_1) = d(\mathbf{x}, \mathbf{y}_2) = \min_{\mathbf{z} \in A} d(\mathbf{x}, \mathbf{z})$. Now, as $d(\mathbf{x}, \frac{\mathbf{y}_1 + \mathbf{y}_2}{2}) < d(\mathbf{x}, \mathbf{y}_1)$, and the convexity of A gives $\frac{\mathbf{y}_1 + \mathbf{y}_2}{2} \in A$, we have a contradiction. ■

We note here that problem (4) has been thoroughly investigated when $A = B$. It has been shown that the optimal solution

should define a centroidal Voronoi tessellation (CVT) on A [10], which means that the generators of the Voronoi cells are also their mass centers (centroids), respectively. Such a distribution of points can be achieved using CVT algorithms [10], [21], [22], [23]. The CVT approach together with the generating algorithms were extended to the tessellation of surfaces by considering constrained CVT (CCVT) [11]. In this paper, we consider the case $A \subseteq B$. Though it has some aspects in common with the existing CCVT techniques, it is a different problem. We will call our approach region-influenced centroidal Voronoi tessellation, or shortly RCVT. Several theoretical results for CVT and CCVT can be shown to hold for RCVT, as well.

We begin by defining the same distortion functional as in [11], [24] for the pair (Y, Z) with $Y, Z \in A^K$ as:

$$\mathcal{E}(Y, Z) = \sum_{i=1}^K \int_{V_i(Y)} \varrho(\mathbf{y}) |\mathbf{y} - \mathbf{z}_i|^2 d\mathbf{y}, \quad (5)$$

with a density function ϱ defined over B . Let us now consider a modification of the fixed point iteration of the so called Lloyd map [10]. Namely, for a set of Voronoi generators $Z \in A^K$, let $T : A^K \rightarrow A^K$, with $T(Z) = Z'$, such that $d(\mathbf{z}'_i, \mathbf{z}_i^*) = \min_{\mathbf{a} \in A} d(\mathbf{a}, \mathbf{z}_i^*)$, for $i = 1, \dots, K$, where:

$$\mathbf{z}_i^* = \frac{\int_{V_i(Z)} \mathbf{z} \varrho(\mathbf{z}) d\mathbf{z}}{\int_{V_i(Z)} \varrho(\mathbf{z}) d\mathbf{z}}. \quad (6)$$

That is, $Z \in B^K$ is the set of mass centers of the corresponding Voronoi cells $\{V_i(Z)\}_{i=1}^K$. In other words, the mapping T moves the current generators Z to those points Z' of A which are closest to the corresponding mass centers Z^* , respectively. The existence of such Z' points is guaranteed by Lemma 3.2. We shall show that the optimally selected points of A regarding problem (4) must be the ones closest to the centroids of the generated Voronoi tessellation. The generators of such a tessellation obviously form a fixed point of the mapping T , which (as we shall show) can be reached via a fixed point iteration of T :

$$Z_n = T(Z_{n-1}), \quad n \geq 1. \quad (7)$$

The above iteration process can be interpreted as a modification of the Lloyd algorithm [22], like in the CCVT case [24]. Thus we can summarize our proposed algorithm in the following way:

Algorithm 3.3: Lloyd algorithm for computing RCVT

Input:

$B \subseteq \mathbb{R}^m$, selected search region;
 $A \subseteq \mathbb{R}^m$, the set to be simplified (with $A \subseteq B$);
 $K \in \mathbb{N}$, the number of generators;
 ϱ , a density function over B ;
 $Z \in A^K$, an initial set of generators;

Output:

$\{V_i\}_{i=1}^K$, a RCVT with K generators $Z \in A^K$;

Iteration:

1. Construct the Voronoi tessellation $\{V_i\}_{i=1}^K$ of B with generators $Z \in A^K$;
2. Define the new set of generators as the points of A closest to the centroids of $\{V_i\}_{i=1}^K$;
3. Repeat steps 1 and 2 until some stopping criterion is met.

Now we show that some basic properties of CVT [24] remain valid also for RCVT.

Lemma 3.4: Let ϱ be a positive and smooth density function defined on a smooth bounded domain B . Then:

- 1) \mathcal{E} is continuous and differentiable in $\overline{B}^K \times \overline{B}^K$ (where \overline{B} stands for the closure of B);
- 2) $\mathcal{E}(Z, T(Z)) = \min_{Y \in \overline{B}^K} \mathcal{E}(Z, Y)$;
- 3) $\mathcal{E}(Z, Z) = \min_{Y \in \overline{B}^K} \mathcal{E}(Y, Z)$.

Proof: The proof of the first statement of Lemma 3.4 is similar to that of the CVT case [24].

The proof of the second statement is based on the fact that the distortion value \mathcal{E} increases with shoving off the mass centers of the $\{V_i(Z)\}_{i=1}^K$ tessellation. Let us fix an $i \in \{1, \dots, K\}$. For simpler calculations, without loss of generality, we may assume that $\mathbf{z}_i^* = \mathbf{O}$. Let us choose now $\mathbf{z}'_i, \mathbf{z}''_i \in V_i(Z)$ such that $|\mathbf{z}'_i| \leq |\mathbf{z}''_i|$. Then:

$$\int_{V_i(Z)} \varrho(\mathbf{y}) |\mathbf{y} - \mathbf{z}'_i|^2 d\mathbf{y} - \int_{V_i(Z)} \varrho(\mathbf{y}) |\mathbf{y} - \mathbf{z}''_i|^2 d\mathbf{y} = (|\mathbf{z}'_i|^2 - |\mathbf{z}''_i|^2) \int_{V_i(Z)} \varrho(\mathbf{y}) d\mathbf{y} \leq 0. \quad (8)$$

Thus, for any $Z' = \{\mathbf{z}'_i\}_{i=1}^K$, $Z'' = \{\mathbf{z}''_i\}_{i=1}^K \in A^K$ with $|\mathbf{z}'_i| \leq |\mathbf{z}''_i|$ for all $i \in \{1, \dots, K\}$, we have $\mathcal{E}(Z, Z') \leq \mathcal{E}(Z, Z'')$, which completes the proof of the second statement of the lemma.

Finally, note, that the third statement of the lemma has already been proved for the CVT case in [24] without the extra restriction $T_i(Z) \in A$. ■

Given these basic properties, we can formulate the important monotone decreasing behavior of the distortion functional \mathcal{E} for the proposed iterative algorithm.

Lemma 3.5: Let $\{Z_n\}_{n=1}^\infty$ be the sequence of generating sets produced by Algorithm 3.3. Then:

- 1) $Z_n = T(Z_{n-1})$;
- 2) $\mathcal{E}(Z_n, Z_n) \leq \mathcal{E}(Z_{n-1}, Z_{n-1})$.

Proof: The first statement of the lemma is obvious, as it is the formal description of how Algorithm 3.3 operates through T .

As for the second statement, by Lemma 3.4, we have

$$\mathcal{E}(Z_n, Z_n) = \min_{Y \in \overline{B}^K} \mathcal{E}(Y, Z_n) \leq \mathcal{E}(Z_{n-1}, Z_n) = \min_{Y \in \overline{B}^K} \mathcal{E}(Z_{n-1}, Y) \leq \mathcal{E}(Z_{n-1}, Z_{n-1}). \quad (9)$$

It has been shown in [24] for the CVT case that, if the density function is positive except on a set of measure zero,

stationary points of the distortion \mathcal{E} are given by fixed points of the Lloyd map T . The result below justifies that fixed points are attainable as a limit of Lloyd iterations in our case, as well.

Theorem 3.6: Any limit set Z of Algorithm 3.3 is a fixed point of the Lloyd map, and thus, (Z, Z) is a critical point of \mathcal{E} . Moreover, for an iteration starting from a given point, all elements in the set of its limit points share the same distortion value \mathcal{E} .

Proof: See Appendix. ■

As an immediate consequence of Lemma 3.5, the points solving problem (4) can be selected through Algorithm 3.3.

Corollary 3.7: Let $Z \in A^K$ be a point set that optimizes problem (4). Then $T(Z) = Z$.

The Lloyd algorithm has the disadvantage that the Voronoi regions must be computed. So, equivalent iterative statistical methods based on random sampling were proposed [21], [24]. They are initialized with a random selection of K points. Then, in every iteration step, a Monte-Carlo sampling is executed to update the centroids of the cells. These methods can easily be adopted for our task, as well.

Algorithm 3.8: Random sampling algorithm for computing RCVT

1. Choose a $q \in \mathbb{N}$ and constants $\alpha_1, \alpha_2, \beta_1, \beta_2$, such that $\alpha_2, \beta_2 > 0$, $\alpha_1 + \alpha_2 = \beta_1 + \beta_2 = 1$; choose an initial set of K points $\mathbf{z}_1, \dots, \mathbf{z}_K$ in A , e.g. by using a Monte Carlo method; set $j_i = 1$ for $i = 1, \dots, K$;
2. Choose q points $\mathbf{y}_1, \dots, \mathbf{y}_q$ in B at random, e.g. by a Monte Carlo method, according to some probability density function (uniform one in this paper);
3. For $i = 1, \dots, K$, collect in the set W_i all sampling points \mathbf{y}_r closest to \mathbf{z}_i among $\mathbf{z}_1, \dots, \mathbf{z}_K$, i.e. the ones lying in the Voronoi region of \mathbf{z}_i w.r.t. B ; if the set W_i is empty, do nothing; otherwise, compute the \mathbf{u}_i average of W_i and set:

$$\mathbf{z}_i \leftarrow \frac{(\alpha_1 j_i + \beta_1) \mathbf{z}_i + (\alpha_2 j_i + \beta_2) \mathbf{u}_i}{j_i + 1}, \quad j_i \leftarrow j_i + 1. \quad (10)$$

The new set of $\mathbf{z}_1, \dots, \mathbf{z}_K$ along with the unchanged \mathbf{z}_j 's (i.e. when W_i is empty), form the new set of points $\mathbf{z}_1, \dots, \mathbf{z}_K$;

4. If for some i , $\mathbf{z}_i \notin A$, then $\mathbf{z}_i \leftarrow \mathbf{z}$, where $d(\mathbf{z}_i, \mathbf{z}) = \min_{\mathbf{y} \in A} \{d(\mathbf{z}_i, \mathbf{y})\}$, i.e., \mathbf{z} is the nearest point to \mathbf{z}_i in A .

5. If the new points meet some convergence criterion, terminate; otherwise, return to step 1.

The actual modification of the basic CVT algorithm [10] is done in step 4, where the centroids are mapped into A , in a similar way as in the corresponding CCVT algorithm [11]. Note that it would not be sufficient to apply this step only once in the end, as the above iteration also influences the positions of the centroids better inside A . Figure 3 shows how the final positions of the centroids change according to the radius of the disc A centered within the unit square B .

An example of the reduction of a contour like object is shown in Figure 4. Here, the object A to be simplified is a circle of radius 0.45. Figure 4a shows the result of the CCVT algorithm, while Figure 4b depicts the outcome of the proposed RCVT method, when B is chosen to be the unit square $B \supset A$. In the CCVT case, equidistant points are

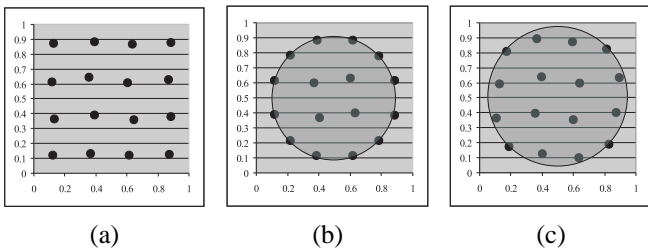


Fig. 3. The result of the modified CVT algorithm when B is the unit square; (a) $A = B$, (b) $A \subset B$ is a disc of radius 0.4 centered at $(0.5, 0.5)$, (c) $A \subset B$ is a disc of radius 0.45 centered at $(0.5, 0.5)$.

preserved, since no outside region is considered there. For RCVT, it can be observed that more points are selected closer to the directions $\pm \frac{\pi}{2}$, $\pm \frac{3\pi}{4}$, since the square B has a larger spread along them. Consequently, more points of A are needed to represent these larger diagonal zones.

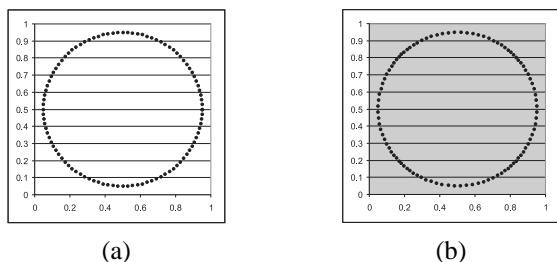


Fig. 4. Simplification of the circle A having radius 0.45 centered at $(0.5, 0.5)$ to 100 of its points; (a) the result of the CCVT algorithm, (b) the result of the RCVT algorithm, when B is the unit square.

The global convergence of the corresponding versions of Algorithms 3.3 and 3.8 were thoroughly investigated only in 1D for the CVT/CCVT case. Regarding CVT, the global convergence is shown for a closed interval [10], while for CCVT, it is shown for smooth bounded curves [11], in the case of any positive and smooth density function. Higher dimensional convergence is still an open issue, and so it is for RCVT, as well.

IV. EXPERIMENTAL RESULTS

In our experiments we considered the discrete object simplification problem on digital images. In the discrete case, we consider the following variant of the continuous distortion functional (5):

$$\mathcal{E}(Y, Z) = \sum_{i=1}^K \sum_{\mathbf{y} \in V_i(Y)} \varrho(\mathbf{y}) |\mathbf{y} - \mathbf{z}_i|^2, \quad (11)$$

with $B = \cup_{i=1}^K V_i(Y)$ being a set of M points in \mathbb{R}^m , and $Y, Z \in A^K$. Following a similar argument to that given in [10] for the CVT case, it can be shown that the above distortion can be minimized by a Voronoi tessellation generated by its centroids Z . Similarly to the CVT case, as the number of sample points M tends to infinity, the convergence of the energy [23] and of the centroids [25] can be proven under certain conditions.

During the RCVT iteration, the generators were supposed to have real coordinates. After the last iteration step they were rounded to the closest point of object A , when they were lying outside A . Additionally, for a more flexible application, we ignored the convexity criteria for the objects without encountering problems in our tests. To overcome possible concavity problems in our implementation, we initialize the RCVT algorithm to start from a point set obtained by uniform sampling of A . In this way, if A has a relatively smooth boundary, the closest point of object A was found within the same Voronoi cell. If possible infinite loops caused by concavity does not let the iteration converge, repetitions can be expected for the alternating closest point assignment caused by concavity. Though we did not need to do so, in case of more severe concavity problems, a promising approach might be the subdivision of A into convex (or smoother) subsets, with separate optimization. However, in this case, the borders of the subsets must be handled with additional care.

We present two experimental setups with the corresponding results of our optimization approach in chamfer matching. First, we show our results in template matching on snake representations for contour simplification. Then we also consider the possible simplification of region object representations.

A. Simplifying object contour representations

In the case of a simple one-pixel wide digital template A , when the search set B is chosen to be at its vicinity, possible naive alternatives can keep randomly selected or equidistant points instead of applying the RCVT algorithm. For a practical example, see Figure 5 for a head template (set A) used in our system, where we applied various percentages of contour point reduction. However, note that equidistant simplification is hard to interpret for non-contour objects, while RCVT can be applied for arbitrary sets without problems. As for random selections, we considered the average performance of some random sampling in our experiments.

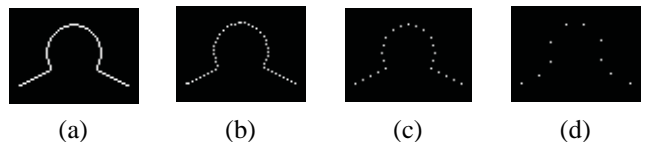


Fig. 5. Reduction of the number of template points for chamfer matching; (a) original template, (b) 50%, (c) 75%, (d) 90% reduction of the template points.

The search regions (sets B), for checking the change in the distance map, can be obtained e.g. by successive dilations of the original template A . Here we used the 3×3 square structuring element C [26]. Figure 6 shows the result of some dilations.

Figure 7 depicts some quantitative results to compare the accuracy of the random, equidistant and RCVT based reduction. In this experiment 25% of the template points were kept to form a new template A' . The calculation was made for dilations $B = A \oplus nC$ of the original head template for several $n \in \mathbb{N}$. The horizontal axis is the number of dilation steps n , whereas the vertical axis shows the distance map error $E =$

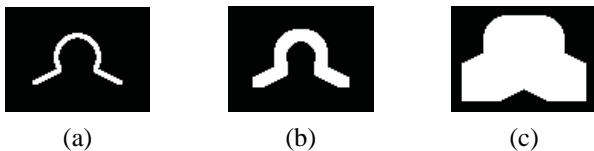


Fig. 6. Dilations of the head template to create the search region $B = A \oplus nC$ for (a) $n = 1$, (b) $n = 3$, (c) $n = 12$.

$\sum_{\mathbf{x} \in B} (d_{A'}(\mathbf{x}) - d_A(\mathbf{x}))$. We can see that the proposed RCVT approach gives remarkably better distance map approximations in the case of larger search regions. Besides the $\langle 3, 4 \rangle$ distance map considered primarily in the paper, we include the results corresponding to the distance map $\langle 5, 7, 11 \rangle$, which is known to be computationally more demanding, but a more accurate approximation of the Euclidean distance [15]. According to this test, we can expect to derive basically the same results in case of any other approximation of the Euclidean distance.

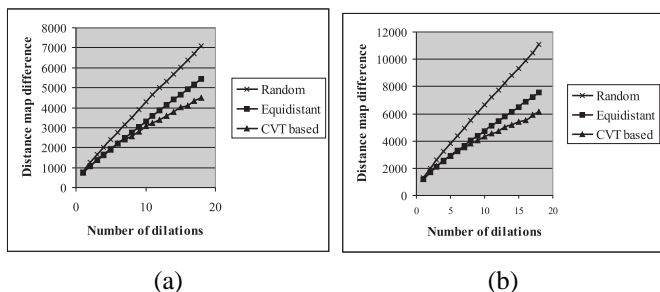


Fig. 7. The comparison of the equidistant and RCVT based reduction of one-pixel wide objects for chamfer matching for different distance maps: (a) $\langle 3, 4 \rangle$ distance map, (b) $\langle 5, 7, 11 \rangle$ distance map.

Another important issue is to investigate how the distance map error E depends on the simplification of the template, in order to determine the acceptable reduction level. According to the results of our experiments shown in Figure 8, we can conclude that the accuracy falls exponentially with the percentage of the retained template points. Severe inaccuracy can be experienced in the case of excessive (80%, 90%) simplification.

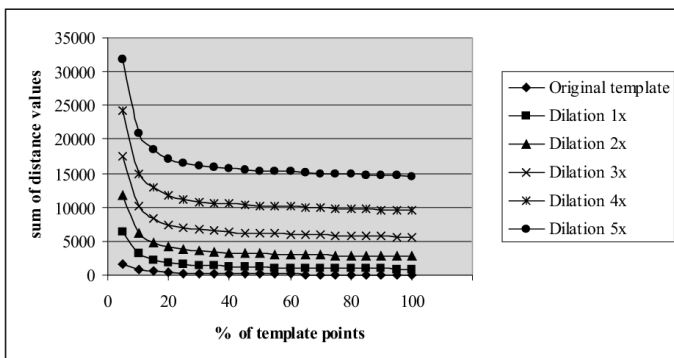


Fig. 8. The change of the distance map error at different levels of reduction of the points of the head template.

1) *Chamfer matching with Gradient Vector Flow (GVF) snakes:* In one of our currently developed person detec-

tion/tracking/recognition systems, we analyze infrared images captured in a fire scene, like the ones shown in Figure 9.



Fig. 9. Thermal images captured under varying temperature conditions.

We cannot use background subtraction here (no pre-recorded background data exists), and hardly the infrared intensity data (due to varying temperature values) to locate objects, e.g. humans. Accordingly, as a robust active contour [27] technique, the GVF snake has been chosen to extract object boundaries [28]. A very useful outcome of the snake algorithm is that, in the case of a closed snake, we have the snake points in an indexed sequence $[\underline{s}_1, \dots, \underline{s}_p]$, with $\underline{s}_p = \underline{s}_1$. The process for evolving the GVF snake considers two additional parameters for the density of the points composing the snake. Namely, d_{\max} and d_{\min} denote the maximum and minimum distance allowed between two snake points, respectively. It can be easily seen that by requiring $d_{\max} < 1$, we can guarantee an 8-connected snake. The main disadvantage of considering small d_{\max} values (a dense snake) is that the iterative process can become very time-consuming. Therefore, an important point in our approach is to investigate how dense the snake points can be for a reliable chamfer matching.

To recognize the object represented by the snake, we match whole object contours or parts of them. For example, for human body detection, the object can be classified as a human, based either on successful whole human contour or on head and limb matching.

2) *Matching along the snake:* In section II, we discussed the difficulties rising from the necessary geometric transformations between the target object and database object representations. This usually results in an exhaustive search for the appropriate transformation parameters. Using snakes, we are in quite a comfortable position to make obvious restrictions to this parameter space. First of all, we can avoid translating the template "blindly" over the entire image, as we adjust its origin to snake points. After translation, we can utilize the sequentiality of the snake points in finding the suitable rotation angle. Namely, we can consider consequent snake points to estimate the direction of the snake by comparing a given snake point with some subsequent ones. Naturally, this method can be adopted easily only for templates having a straight starting segment, in which case the straight segment can be aligned to the estimated (or close) angle. Otherwise, the divide and conquer strategy can be used here, for rotation. The magnification parameter can be bounded easily by adjusting it, regarding the spatial "size" (e.g. perimeter, area, bounding box) of the snake.

Moreover, as now we have a closed boundary with no outliers, it is less important to involve edge direction information

[29] in measuring how good the fit is. The "blind" methods also suffer from the problem of selecting suitable threshold(s) for (e.g. the Canny) edge detection and giving many false negatives in the case of a cluttered/noisy scene. Taking all these factors into consideration, we can perform the matching steps in an obviously shorter time than in the general case. Thus, considering the geometric transformations summarized in section II regarding scaling, translating and rotating, now the best match at $\underline{s}_i \in S$ can be defined as:

$$d_{\underline{s}_i}(\widehat{S}, T) = \min_{\substack{\lambda \in \Lambda \\ \Theta \in [0, 2\pi[}} (d_{\underline{s}_i}(S, \lambda T_{\Theta})), \quad (12)$$

where Λ is a set of possible scaling values, and T_{Θ} denotes the set T rotated around its origin by Θ . Consequently, the best matching value can be given as

$$d(S, T) = \min_{\underline{s}_i \in S} (d_{\underline{s}_i}(\widehat{S}, T)), \quad (13)$$

and the best matching position is the snake point where (13) is taken.

3) *Matching human body parts:* In this section, we present some experimental results regarding matching human body parts, like the head and limbs. Our matching process is based on the above described steps, and we tested how the density of the snake S and the reduction of the templates affected matching reliability. The templates T_j (head and leg) and their simplified versions were matched along the snake, as it was described in the previous section. Figure 10 shows examples for matching the head and leg template.



Fig. 10. Best fitting positions of templates for a human silhouette represented by a snake; (a) head, (b) leg.

We experienced that simplifying the template and considering a less dense snake representation speed up matching computations. Figure 11 shows $d_{\underline{s}_i}(\widehat{S}, T)$ for various densities of the snake and head template points. The correct position for the template (shown in Figure 10a) was found in all these cases according to the minimal distance value (normalized to 1) indicated by an arrow in Figure 11. However, the reliability of matching naturally deteriorates when less snake/template data are used, as it can be checked in Figure 12, as well. Here the corresponding Receiver Operating Characteristic (ROC) curves [30] for the cases shown in Figure 11a-d are presented, respectively. The curves show how true (acceptable matching positions) and false positives are found by raising the threshold value for the distance map error. For this experiment the snake points were manually pre-classified as acceptable/unacceptable matching positions, that is, true/false positives.

Moreover, in order to have an experimental comparison between the RCVT and the naive contour reducing approaches

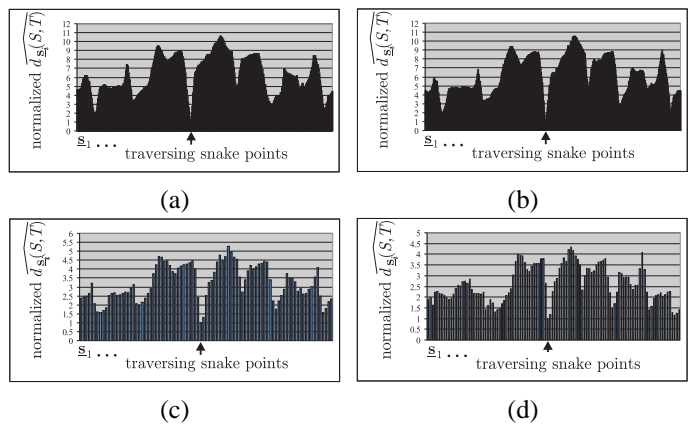


Fig. 11. Point-wise goodness of fit distance profile for matching the head template against the body contour at different levels of template simplification and snake density. Best match is found at the normalized sum of distance values 1; (a) 100% retained template points, dense snake ($d_{\max} < 1$), (b) 100% retained template points, less dense snake ($d_{\max} < 4$), (c) 25% retained template points, dense snake ($d_{\max} < 1$), (d) 25% retained template points, less dense snake ($d_{\max} < 4$).

(random and equidistant sampling), we set up a test environment. In this experiment, we performed the chamfer matching of head templates against a database of 60 elements containing the original head template distorted by several geometric transformations (stretching and skewing) together with some head silhouettes extracted from real videos. Besides making a test without any reductions, we considered random, equidistant, and RCVT-based simplification of the head template. Naturally, in case of a simplification, the same number of points were retained.

Our main aim here was to experimentally validate the assumption that the original head template can be replaced more reliably with applying RCVT instead of some other naive simplification approach. As an easily obtainable result for the simplified objects, first we checked the deviation of the best matching positions of the simplified templates from the best matching position of the original head template. The deviation was calculated considering all the database elements as the sum of the squared distance between the best matching positions found for the original (non-simplified), and simplified templates, respectively. This analysis gives a preliminary impression on which simplification approach can lead to the most valid replacement of the original template. In the way discussed before, we considered RCVT simplification of the head template regarding several search regions, which naturally had no effect on the performance of the equidistant and random sampling. The deviations are shown in Figure 13 for this experiment.

We can see that RCVT provides improvement regarding both the random and equidistant sampling. We note two things here. On one hand, this analysis gives a quick impression about the possible improvement obtainable by RCVT. On the other hand, we have to keep in mind that the selection of a larger search region does not lead necessarily to better matching performance. With the selection of the search region size for RCVT, we can respond to the expected spatial deviation of the template from the object. In other words, if less precise

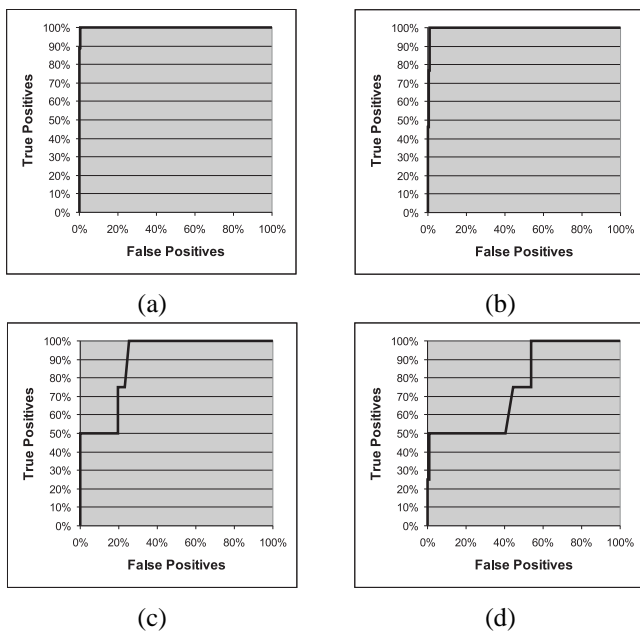


Fig. 12. ROC curves for matching the head template against the body contour at different levels of template simplification and snake density; (a) 100% retained template points, dense snake ($d_{\max} < 1$), (b) 100% retained template points, less dense snake ($d_{\max} < 4$), (c) 25% retained template points, dense snake ($d_{\max} < 1$), (d) 25% retained template points, less dense snake ($d_{\max} < 4$).

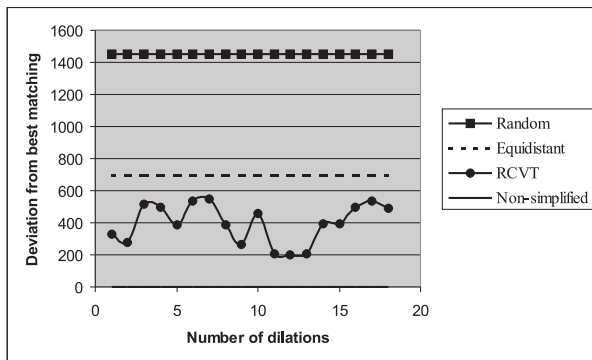


Fig. 13. Comparing the performance of simplified head templates with considering the sum of the distances between their best matching positions from those of the original head template for a data set of head silhouettes.

matching is expected, a larger search region can be used to try to cover a larger area.

To confirm this hypothesis and obtain more detailed comparative results, we performed a second test using the test database. Namely, a *good match* region was defined as a neighborhood of the best matching position of the original template. This way, we can create ROC curves to see how similarly the simplified templates behave compared to the original one. To perform this analysis and also to validate Figure 13, we considered RCVT simplifications belonging search regions of 5, 12, 18 dilations, respectively. The results are shown in Figure 14.

As it can be seen in Figure 14, the results suggested by Figure 13 are confirmed corresponding to the deviation from

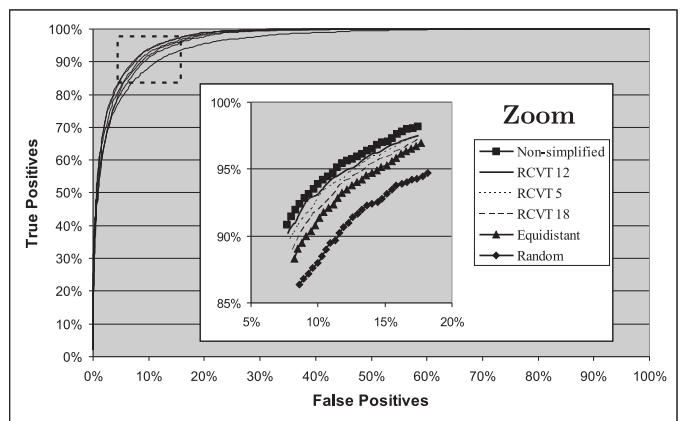


Fig. 14. ROC curves to measure the performance of the simplification methods applied to the head template on an experimental data set of target head objects. The numbers assigned to the RCVT simplifications (5, 12, 18) refer to the size of the search region in terms of dilation steps.

the best matching positions. For simplicity, in this approach we excluded all geometric distortion and snake-based matching issues, and performed pure chamfer matching with the translation of the head template on the input distance maps.

The computation time increases linearly with the percentage of the template points retained, since the same family of operations should be performed for a larger point set. Our experiments also reflected this behavior, as it can be seen in Figure 15 for the head template. Similar results were found for the leg templates.

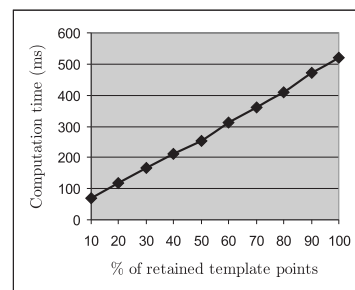


Fig. 15. Computation time of object matching vs. percentage of retained points of the original head template.

B. Simplifying region object representation

Though chamfer matching is usually employed to match object contour representations, it is also possible to apply it to region matching, e.g. for human body part matching [31]. After applying RCVT optimization to simplify regions, we obtained similar experimental results to the contour case described in section IV-A. Such an example for a human body silhouette is shown in Figure 16.

If the regions to be matched are approximately of the same size (whole shape matching), then it can be sufficient to execute the matching as we discussed for the contour objects. However, if we want to match object parts, as we discussed it in section IV-A.3, then we need to modify our approach.

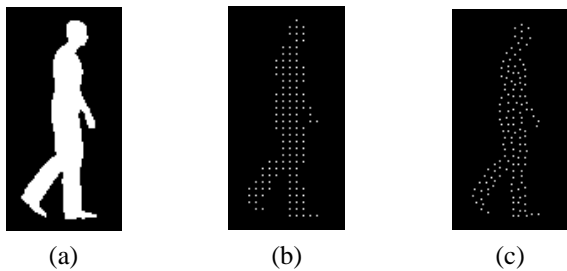


Fig. 16. The result of the RCVT algorithm in 2D; (a) the object A to be simplified (walking human), (b) the initial generators chosen by naive equidistant sampling, (c) the generators achieved by the RCVT algorithm for $A = B$.

Because, with the same setup, the object part would give very good matches anywhere inside the target object, as the distance values are 0 there (or close to 0 in case of simplification). To overcome this difficulty, we can consider a subregion \hat{A} of the complement A^c of the object part (its background). A natural selection for \hat{A} is $\hat{A} = (A \oplus nC) \cap W$ for some $n \in \mathbb{N}$, and $W \subseteq A^c$, as shown in Figure 17b. The role of W is to define an area around the object part A within its background. If desired, we can simplify A and \hat{A} using the RCVT approach, obtaining A' and \hat{A}' , respectively.

Furthermore, we determine the distance maps both for the target object S (see Figure 17a) and for A (see Figure 17b). To determine how good the fit is at a point $\underline{x} \in \mathbb{Z}^2$, we calculate:

$$\sum_{\underline{y} \in A' \cup \hat{A}'} |d_S(\underline{x} + \underline{y}) - d_A(\underline{y})|. \quad (14)$$

The best match is found when this sum is minimal. Note that, in this example, we ignored all the geometric transformation issues discussed in section II. To match body parts based on regions, another approach could consider signed distance maps [32].

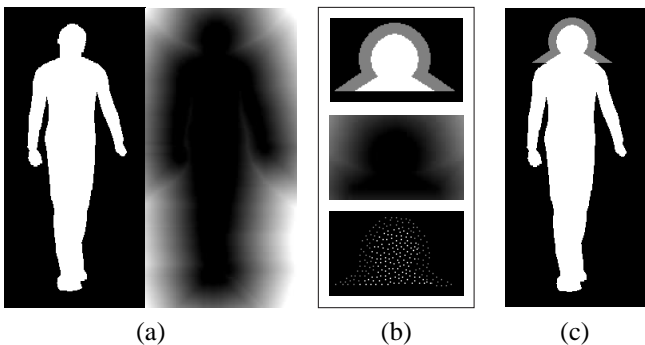


Fig. 17. Chamfer matching of object parts; (a) target object, and its distance map, (b) from top to bottom: head template region A (white) with a subset \hat{A} of its complement (gray), the distance map of A , and the simplification of A and \hat{A} , (c) best matching position for the simplified template.

V. CONCLUSIONS

This paper investigates the simplification of object representation and its effects on the accuracy of the general chamfer matching approach. We propose a novel method for optimal simplification, while keeping the generated distance map close

to the original one concerning different search regions. To reach our aims, we incorporated several well-founded results from the field of centroidal Voronoi tessellations. We found that object simplification is possible in matching body parts for human detection and recognition. We also explained how the snake representation for the target object speeds up chamfer matching by getting rid of the laborious search of the transformation parameter space. We proposed some application schemes to demonstrate the usability of the simplification of region-based representations of objects, as well.

One of our future research aims within this topic is to investigate other application fields, where the proposed method to reduce the number of points can be useful. Such a promising field might be the fast registration of point clouds using the Iterative Closest Point (ICP) algorithm. On the other hand, we also plan to take better advantage of the flexibility that our approach offers for silhouette matching. One possibility is to focus more on some object regions during its simplification with defining a specific weight function accordingly. For example, larger uncertainty closer to the boundary of the object can be compensated by a weight function concentrating to the skeleton. It is also clear that a parallel implementation of the proposed method would improve the performance of chamfer matching.

ACKNOWLEDGEMENT

This work was partially supported by the project SHARE: Mobile Support for Rescue Forces, Integrating Multiple Modes of Interaction, EU FP6 Information Society Technologies, Contract Number FP6-004218. The authors are also grateful to the reviewers for their thorough work to improve the content of the paper.

APPENDIX

PROOF OF THEOREM 3.6

Algorithm 3.3 produces a bounded sequence $\{Z_n\}_{n=1}^{\infty}$ in \overline{B}^K and thus it has a convergent subsequence. Let Z be a limit point of $\{Z_n\}_{n=1}^{\infty}$ with a subsequence $\{Z_{n_j}\}_{j=1}^{\infty}$ such that $\lim_{n_j \rightarrow \infty} Z_{n_j} = Z$. Since the distortion values are monotone decreasing, all the limit points have the same distortion value:

$$\mathcal{E}(Z, Z) = \lim_{n_j \rightarrow \infty} \mathcal{E}(Z_{n_j}, Z_{n_j}) = \inf_n \mathcal{E}(Z_n, Z_n). \quad (15)$$

Let \mathcal{E}_1 and \mathcal{E}_2 denote the the partial derivatives with respect to all the variables of the first and second argument of \mathcal{E} , respectively. Then Lemma 3.4 implies:

$$\mathcal{E}_1(U, Z_n)|_{U=Z_n} = 0, \quad (16)$$

and by continuity:

$$\mathcal{E}_1(Z, Z) = 0. \quad (17)$$

Now if $\mathcal{E}_2(Z, U)|_{U=Z} = 0$, then (Z, Z) is a critical point of \mathcal{E} and the proof is complete. Otherwise, there exists a Y such that:

$$\mathcal{E}(Z, Y) < \mathcal{E}(Z, Z). \quad (18)$$

Then, for every sufficiently small δ , there exists an index n_j such that

$$\begin{aligned} \mathcal{E}(Z_{n_j}, Y) &< \mathcal{E}(Z, Y) + \delta < \mathcal{E}(Z, Z) \leq \\ \mathcal{E}(Z_{n_j+1}, Z_{n_j+1}) &\leq \mathcal{E}(Z_{n_j}, Z_{n_j+1}), \end{aligned} \quad (19)$$

contradicting the fact that

$$\mathcal{E}(Z_{n_j}, Z_{n_j+1}) = \min_{Y \in \overline{B}^K} \mathcal{E}(Z_{n_j}, Y). \quad (20)$$

■

REFERENCES

- [1] H.G. Barrow, J.M. Tenenbaum, R.C. Bolles, and H.C. Wolf, "Parametric correspondence and chamfer matching: Two new techniques for image matching." in *Proc. 5th Int. Joint Conf. Artificial Intelligence*, Cambridge, MA, 1977, pp. 659–663.
- [2] D.M. Gavrilu, and V. Philomin, "Real-time object detection for smart vehicles." in *Proc. of IEEE International Conference on Computer Vision*, Kerkyra, Greece, 1999, pp. 61–71.
- [3] M. Kumar, P.H.S. Torr, and A. Zisserman, "Extending pictorial structures for object recognition." in *Proc. of British Machine Vision Conference*, London, 2004.
- [4] D. Huttenlocher, and P. Felzenszwalb, "Pictorial structures for object recognition." *Intl. Journal of Computer Vision*, vol. 61, no. 1, pp. 55–79, 2005.
- [5] T. Zhao, and R. Nevatia, "Stochastic human segmentation from a static camera." in *Proc. of IEEE Workshop on Motion and Video Computing*, Orlando, Florida, USA, 2002, pp. 9–14.
- [6] F. Xu, and K. Fujimura, "Pedestrian detection and tracking with night vision." in *Proc. of IEEE Intelligent Vehicle Symposium*, Versailles, France, 2002, pp. 21–30.
- [7] D. Huttenlocher, G. Klanderman, and W. Rucklidge, "Comparing images using the Hausdorff distance." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, pp. 850–863, 1993.
- [8] G. Borgefors, "Hierarchical chamfer matching: A parametric edge matching algorithm." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 6, pp. 849–865, 1988.
- [9] J. You, W. Zhu, E. Pissaloux, and H. Cohen, "Hierarchical image matching: A chamfer matching algorithm using interesting points." *International Journal of Real-time Imaging*, vol. 1, pp. 245–259, 1995.
- [10] Q. Du, V. Faber, and M. Gunzburger, "Centroidal Voronoi tessellations: Applications and algorithms." *SIAM Review*, vol. 41, pp. 637–676, 1999.
- [11] Q. Du, M. Gunzburger, and L. Ju, "Constrained Centroidal Voronoi tessellations on general surfaces." *SIAM J. Scientific Comp.*, vol. 24, pp. 1488–1506, 2003.
- [12] P.E. Danielsson, "Euclidean distance mapping." *Comput. Graph. Image Process.*, vol. 14, pp. 227–248, 1980.
- [13] T.E. Schouten, and E.L. van den Broek, "Fast exact Euclidean distance (FEED) transformation." in *Proc. of International Conference on Pattern Recognition*, Cambridge, UK, 2004, pp. 594–597.
- [14] A. Rosenfeld, and J. L. Pfaltz, "Distance functions on digital pictures." *Pattern Recognition*, vol. 1, pp. 33–61, 1968.
- [15] G. Borgefors, "Distance transformations in digital images." *Computer Vision, Graphics, and Image Processing*, vol. 34, no. 3, pp. 344–371, 1986.
- [16] K. Toyama, and A. Blake, "Probabilistic tracking with exemplars in a metric space." *Intl. Journal of Computer Vision*, vol. 48, pp. 9–19, 2002.
- [17] A. Pinz, M. Prantl, and H. Ganster, "A robust affine matching algorithm using an exponentially decreasing distance function." *Journal of Universal Computer Science*, vol. 1, no. 8, pp. 614–631, 1995.
- [18] I. Pitas, and A.N. Venetsanopoulos, *Nonlinear Digital Filters: Principles and Applications*. Kluwer Academic, Europe, 1990.
- [19] W. Rucklidge, "Locating objects using the Hausdorff distance." in *Proc. of International Conference on Computer Vision*, Boston, USA, 1995, pp. 457–464.
- [20] A. Hajdu, A. Roubies, and I. Pitas, "Optimized chamfer matching for snake-based image contour representations." in *IEEE International Conference on Multimedia & Expo (ICME 2006)*, Toronto, Canada, 2006, to appear.
- [21] L. Ju, Q. Du, and M. Gunzburger, "Probabilistic methods for centroidal Voronoi tessellations and their parallel implementations." *Parallel Computing*, vol. 28, pp. 1477–1500, 2002.
- [22] S. Lloyd, "Least square quantization in PCM." *IEEE Transactions on Information Theory*, vol. 28, pp. 129–137, 1982.
- [23] J. MacQueen, "Some methods for classification and analysis of multivariate observations." in *Proc. of Fifth Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, Berkeley, 1967, vol. I, pp. 281–297.
- [24] Q. Du, M. Emelianenko, and L. Ju, "Convergence properties of the Lloyd algorithm for computing the Centroidal Voronoi Tessellations." *SIAM J. Num. An.*, vol. 44, no. 1, pp. 102–119, 2006.
- [25] D. Pollard, "Strong consistency of k-mean clustering." *Ann. Statist.*, vol. 9, pp. 135–140, 1981.
- [26] I. Pitas, *Digital Image Processing Algorithms*. Prentice Hall, 1993.
- [27] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models." *Int. J. Comput. Vis.*, vol. 1, pp. 321–331, 1987.
- [28] C.Y. Xu, and J.L. Prince, "Snakes, Shapes, and Gradient Vector Flow." *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 359–369, 1998.
- [29] C. Olson, and D. Huttenlocher, "Automatic target recognition by matching oriented edge pixels." *IEEE Transactions on Image Processing*, vol. 6, pp. 103–113, 1997.
- [30] C.E. Metz, "Basic principles of ROC analysis." *Semin Nucl. Med.*, vol. 8, no. 4, pp. 283–98, 1978.
- [31] T. Foures, and P. Joly, "A multi-level model for 2d human motion analysis and description." in *Proc. of SPIE-IS&T Electronic Imaging*, Santa Clara, CA, 2003, pp. 61–71.
- [32] G.Z. Ye, "The signed euclidean distance transform and its applications." in *Proc. 9th International Conference on Pattern Recognition*, Rome, Italy, 1988, vol. I, pp. 495–499.



András Hajdu received his MSc degree in Mathematics from the Lajos Kossuth University, Hungary, in 1996. He obtained his PhD degree in Mathematics and Computer Science from the University of Debrecen, Hungary, in 2003. He worked as a Post Doc researcher for the Artificial Intelligence Information Analysis Laboratory, Dept. of Informatics, Aristotle University of Thessaloniki in 2005–2006. From 2001 he served as Assistant Lecturer and since 2003 he has been an Assistant Professor at the University of Debrecen. He is a member of the Janos Bolyai Mathematical Society, John von Neumann Computer Society (Hungary), Public Body of the Hungarian Academy of Sciences, and the Hungarian Association for Image Analysis and Pattern Recognition. He has authored or co-authored 16 journal papers and 37 conference papers. His main interest lies in discrete mathematics with applications in digital image processing.



Ioannis Pitas received the Diploma of Electrical Engineering in 1980 and the PhD degree in Electrical Engineering in 1985 both from the Aristotle University of Thessaloniki, Greece. Since 1994, he has been a Professor at the Department of Informatics, Aristotle University of Thessaloniki. From 1980 to 1993 he served as Scientific Assistant, Lecturer, Assistant Professor, and Associate Professor in the Department of Electrical and Computer Engineering at the same University. He served as a Visiting Research Associate or Visiting Assistant Professor at several Universities. He has published 140 journal papers, 350 conference papers and contributed in 18 books in his areas of interest and edited or co-authored another 5. He has also been an invited speaker and/or member of the program committee of several scientific conferences and workshops. In the past he served as Associate Editor or co-Editor of four international journals and General or Technical Chair of three international conferences. His current interests are in the areas of digital image and video processing and analysis, multidimensional signal processing, watermarking and computer vision.