MULTISCALE MODEL. SIMUL.
Vol. 1, No. 1, pp. 1–24

ON THE THEORY OF PLANAR SHAPE*

J. L. LISANI[†], L. MOISAN[‡], P. MONASSE[§], AND J. M. MOREL[‡]

Abstract. One of the aims of computer vision in the past 30 years has been to recognize shapes by numerical algorithms. Now, what are the geometric features on which shape recognition can be based? In this paper, we review the mathematical arguments leading to a unique definition of planar *shape elements*. This definition is derived from the invariance requirement to not less than five classes of perturbations, namely noise, affine distortion, contrast changes, occlusion, and background. This leads to a single possibility: shape elements as the normalized, affine smoothed pieces of level lines of the image. As a main possible application, we show the existence of a generic image comparison technique able to find all shape elements common to two images.

Key words. invariant planar shape recognition, generic algorithm

AMS subject classification. 62H35

PII. S1540345902410846

1. What is shape?

"Shape" can have different meanings. Most authors refer to shape as a common denominator between several identical or similar three-dimensional (3D) objects seen from different points of view. The problem of "recognizing a 3D object from a single view" has been extensively studied [22, 26, 34, 41]. When several uncalibrated views of the same 3D object are available, its 3D model can be reconstructed [21, 24].

From another, more restrictive point of view adopted in phenomenology [5, 37], shape means a subset of an image, digital or perceptual, endowed with some qualities permitting its recognition. We call such perceptual objects *planar shapes*. The very notion of shape is linked to the recognition problem. Thus, it is licit to use recognition processes as a way to define classes of shape: one then calls shape any part of an image which can be recognized in another image.

Here again, some restriction must be made. The common use of planar shape always involves a set of possible deformations. The usual ones are translation, rotation, and zoom. Now, authors have explored in recent years more general deformations, particularly elastic ones. In [18, 51], the matching of images or shapes is made up to a flow of diffeomorphisms deforming one of the images (or shapes) onto the other one. These papers, and many references therein, give a well developed mathematical treatment of a problem raised as early as 1983 in [6]: the matching of deformed radiographic images to idealized atlas images.

We therefore have two meanings for shape: one is the 3D bulk of an object and the other one denotes an equivalence class of planar shapes, or images, under a deformation group.

[*] Received by the editors July 9, 2002; accepted for publication (in revised form) October 27, 2002; published electronically January 28, 2003. This work was supported by Office of Naval Research grant N00014-97-1-0839.

http://www.siam.org/journals/mms/1-1/41084.html

[†] Universitat de les Illes Balears, Cra. de Valldemossa, km 7.5, 07071 Palma, Spain (joseluis.lisani@uib.es). This author was partially supported by CICYT project TIC99-0266.

[‡] CMLA, ENS Cachan, 61, av. du Pdt Wilson, 94235 Cachan, France (moisan@cmla.ens-cachan.fr, morel@cmla.ens-cachan.fr). The work of these authors was supported in part by Centre Nationale d'Etudes Spatiales, Centre National de la Recherche Scientifique et Ministère de la Recherche et de la Technologie.

[§] Cognitech Inc., 225 S. Lake Avenue, Pasadena, CA 91101 (monasse@cmla.ens-cachan.fr).

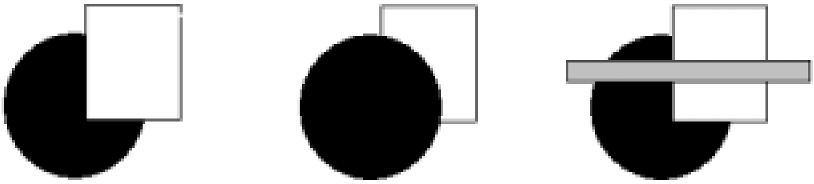

FIG. 1.1. *According to the theory of G. Kanizsa and his school, shapes can be recognized even when they undergo several occlusions. Our perception is trained to recognize shapes which are only seeable in part.*

Shape recognition must be performed in spite of *occlusion.* The phenomenology of occlusion was thoroughly studied by Kanizsa [27] and his school. Figure 1.1 illustrates how invariant our vision is to occlusion. Most viewers describe it as made of circles and rectangles; they do not even notice that such a description is inferential, rectangles and disks simply not being there. Kanizsa argues that occlusion is always present in every day's vision: most objects we see are partially hidden by other ones. Our perception must therefore perform a recognition of partial shapes.

Another structural difficulty arising in shape recognition was first pointed out in perception psychology as the *figure-background problem*, studied by Rubin [45]. It is the other face of the occlusion problem: a shape is superimposed to a background, which can be made of various objects. How does one extract, or single out, the shape from that clutter? This can also be viewed as a dilemma: do we first extract the shape and then recognize it or, conversely, do we extract it *because* we had it recognized?

There are other perturbations affecting the identity of shapes. Shapes are easily recognized in images in spite of a change in the color and luminance scale; shapes are also easily recognized when noise and blur are present. Thus, we can list not less than five kinds of perturbations for planar shapes, which do not affect shape recognition:

- elastic or projective deformations;
- the classical noise and blur, inherent to any perception and any image by Shannon's theory;
- changes of contrast;
- the occlusions;
- the background.

We should notice that the last two mentioned perturbations are of a very different nature: indeed, occlusion is related to a removal of parts of the shape, while the background can lead to additions as well. This means only that we are not able to define the shape as a whole but rather as a conjunction of geometric features, which we shall call *shape elements.* Thus, a natural strategy to the definition of shape has two main steps.

1. Define shape elements as any local, contrast invariant, and affine invariant part of the image.
2. Define shape as a conjunction of shape elements which can be recognized in several different images.

This strategy is classical and adopted (e.g.) in the book [44] and many references therein. We notice that shape elements are then defined by invariance arguments only and are therefore fully accessible to a mathematical discussion.

Shapes, instead, are of empirical nature and will be learned by the recognition of several shape elements. We shall focus first on shape elements and then, only in the

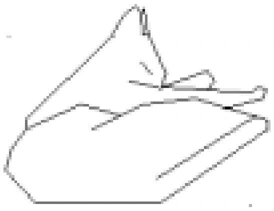

FIG. 1.2. *Sleeping cat? According to the perception psychologist Attneave* [5], *the main information in shapes is contained in the high curvature points: "Common objects may be represented with great economy, and fairly striking fidelity, by copying the points at which their contours change direction maximally, and then connecting these points appropriately with a straightedge."*

final part, outline several uses which can be made of the shape elements in various recognition and image comparison contexts.

Let us end this introduction with some cues about how we shall compute the shape elements. Not later than 1954, a visionary paper [5] by the perception psychologist Attneave outlined most of the shape encoding program we intend to describe. "Information", he wrote, "is concentrated along contours (i.e., regions where color changes abruptly), and is further concentrated at those points on a contour at which its direction changes most rapidly (i.e., at angles or peaks of curvature)" (see Figure 1.2). As we shall see in the citation of section 2.4, he also defined, in literary but accurate terms, how a smoothing of the contours should be performed. His description was shown afterwards to correspond to curvature motion [36, 28].

If we follow step by step the propositions of Attneave, made at a time where no computer vision, and almost no computers, existed, we are led to solve the following problems:

- extract curves as contours of the image;
- smooth such curves by a curvature motion (we shall improve here Attneave's proposition by using affine invariant curve motion);
- encode locally such curves so as to get an encoding robust to occlusion and background. Those invariant codes will be the shape elements.

Our plan follows from the preceding discussion. In section 2, we translate the four invariance arguments into as many geometric and algorithmic requirements for the extraction, filtering, and encoding of shapes in digital images. The affine invariant shape smoothing algorithm is analyzed in detail in sections 3 and 6. We describe the final encoding method, after smoothing of the extracted curves, in section 4. Section 5 shows several image comparison experiments.

2. Invariance arguments: A derivation of shape elements.

In this section, we address the problem of deciding which information in a digital image is relevant to define or recognize shapes. We shall show that invariance arguments enforce the definition of the most basic shape elements. The problematic is illustrated in Figure 2.1, with two different paintings of Georges de la Tour containing similar shapes. Our presentation method here, and in the next sections, follows [33].

2.1. The local contrast invariance argument.

"The concentration of information in contours is illustrated by the remarkable similar appearance of objects alike in contour and different otherwise. The 'same' triangle, for example, may be either white on black or green on white. Even more impressive is the familiar fact that an artist's sketch, in which lines are substituted for sharp color gradients, may constitute

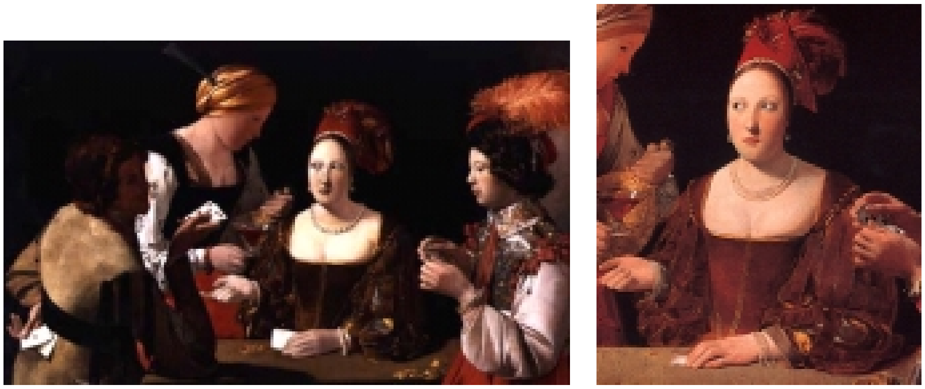

Fig. 2.1. *Original images. These images correspond to different paintings from the French painter Georges de la Tour. The painting on the left is called "Le tricheur à l'as carré" (Kimball Art Museum, Ft. Worth, TX). The painting on the right is a fragment of "Le tricheur à l'as de diamants" (Louvre, Paris, France). Images used with permission from Olga's Gallery—Online Art Museum, www.abcgallery.com.*

a readily identifiable representation of a person or thing." (Attneave [5], 1954).

"I stand at the window and see a house, trees, sky. Theoretically I might say there were 327 *brightnesses and nuances of colour. Do I have* '327'*? No. I have sky, house, and trees. It is impossible to achieve* '327' *as such. And yet even though such droll calculation were possible and implied, say, for the house* 120*, the trees* 90*, the sky* 117*—I should at least have this arrangement and division of the total, and not, say,* 127 *and* 100 *and* 100*; or* 150 *and* 177*."* (Wertheimer [49], 1923).

The above quotes in the founding papers of the gestaltists Attneave and Wertheimer point out the same invariance principle: shape perception is independent of the grey level scale or of the measured colors. This perceptual axiom is easily explained by physical arguments. Indeed, the response of the various captors receiving the light varies according to several unknown factors such as brightness of the sky, aperture of the optical apparatus, and physical sensibility range of the captor. The only reliable information left is actually the order of brightness. As Wertheimer says, we do not see "327," but he noticed that we are at least able to see some contrast between 327 and 150.

We define a digital image as a function $u(x)$, where $u(x)$ represents the grey level or luminance at x.[1] According to the contrast invariance principle, our first task is to extract from the image a topological information fairly independent from the varying and unknown contrast change function of the optical and/or biological apparatus. We can model such a contrast change function as any continuous increasing function g from $\mathbb{R}^+$ to $\mathbb{R}^+$. The real datum when we observe u could be as well any image $g(u)$. This simple argument leads to select the set of level sets of the image [47], or its set of level lines, as a complete contrast invariant image description [10].

Definition 2.1. *The upper topographic map of an image is the family of the connected components of the level sets of* u, $[u \geq \lambda]$, $\lambda \in \mathbb{R}$.

[1] Shape recognition algorithms do not use much color. The geometric information brought by color usually is redundant with respect to luminance [11].

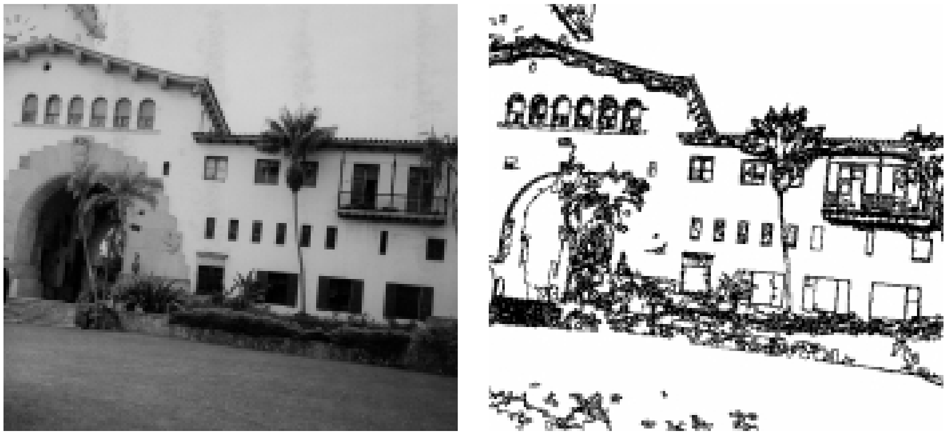

Fig. 2.2. *Left: original image. Right: meaningful level lines. These lines make a sparse nested set of Jordan curves. Each one of the Jordan curves can be used directly for shape based image comparison.*

An image can be reconstructed from its upper level sets by the formula

$$u(x) = \sup\{\lambda, u(x) \geq \lambda\}. \tag{2.1}$$

We define the level lines as the boundaries of the level sets. There are several frameworks to define the level lines: if u is considered to be a function with bounded variation, the level lines can be defined as a set of nested Jordan curves [3]. The set of all level lines is called the *topographic map* of the image.

2.2. The concentration of information argument. The preceding subsection led us to define the set of the image level lines as a complete contrast invariant information. Somewhat in contradiction with this contrast invariance principle, many authors assert, like Attneave, that "Information is concentrated along contours (i.e., regions where color changes abruptly)." One can argue that not all of the level lines are really needed to have a complete description. Some of them are due to noise or to small, hardly noticeable, changes in illumination. Thus, it makes sense to prune the tree of level lines by only keeping a selection of the most contrasted level lines. This is not an essential step, but it permits one to accelerate a lot shape recognition algorithms. A simplification of the tree of level lines can be performed by using the method proposed in [14], which retains roughly all "meaningful" level lines. These level lines are defined by applying Helmholtz's perception principle: an observed geometric structure is perceptually "meaningful" if its number of occurrences would be very small in a random situation. Desolneux, Moisan, and Morel [14] propose a method that, based on contrast measurements, assigns an expectation to each level line. Only those level lines with small expectation are retained, i.e., the ones which are contrasted enough to be "above the noise." We shall show in the experiments only those well-contrasted, "meaningful" level lines. This method usually reduces by a factor 10 the number of level lines and therefore speeds up the recognition method by a 100 factor. Figure 2.2 shows an example of the set of meaningful level lines for a given image.

2.3. The occlusion and figure-background arguments. In section 2.1, we have reduced an image, from the shape parsing viewpoint, to the set of all its level

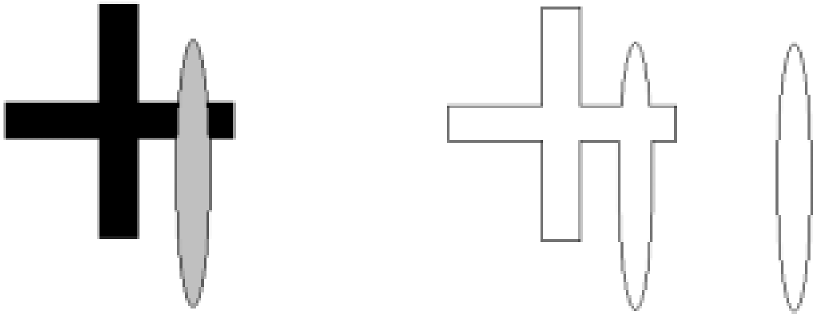

Fig. 2.3. *Left: oval occluding a cross. Right: the level lines of the resulting image. While the oval's boundaries can be recovered as a full level line, the boundary of the cross concatenates with the oval's boundary. Thus recognition cannot be based on level lines, but it can still be based on pieces of level lines.*

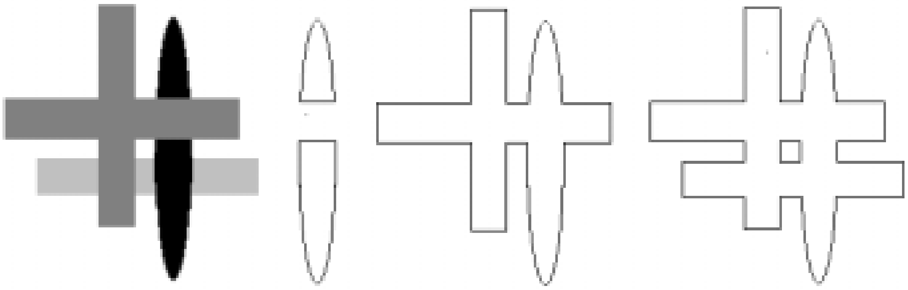

Fig. 2.4. *Left: cross on a background with an oval occluding a rectangle. The cross is wholly in view. All the same, its shape does not appear as a level line because of the background. As in Figure* 2.3*, one sees that the level lines must be broken into pieces to get clues of each single shape.*

lines, otherwise called the *topographic map.* The topographic map is in no way the ultimate description of shape elements. When a shape A partially occludes a shape B, the level lines of the resulting image are a concatenation of pieces of the level lines belonging to A and to B. Thus, the Jordan curves of the topographic map are not simple shape elements and must be further decomposed: a segmentation of them into their parts belonging to different objects is requested. This is shown with a very simple example in Figure 2.3. Thus, we are led to the following definition.

Definition 2.2. *We call* shape element *of an image u any piece of any level line of the image.*

Even if a shape is not occluded but simply occludes its own background, there may be no level line surrounding the whole shape, as we show in Figure 2.4. This situation is quite general: the level lines of the background can be concatenated with the level lines of an object in view.

2.4. The smoothing argument. In order to justify the necessity of a multiscale smoothing for shapes, we can again rely on Attneave:

"It appears, then, that when some portion of the visual field contains a quantity of information grossly in excess of the observer's perceptual capacity, he treats those components of information which do not have redundant representation somewhat as a statistician treats 'error variance,' averaging out particulars and abstracting certain statistical homogeneities. Such an averaging process was involved in drawing the cat

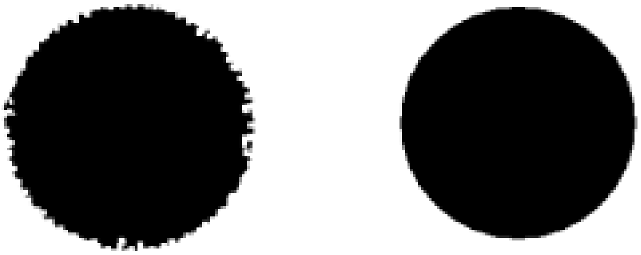

FIG. 2.5. *One can immediately see that those two objects are disks, with approximately the same size. The second one is obtained from the first by the affine curvature equation* [2]. *Such a curvature equation was anticipated by Attneave, who proposed to smooth the silhouette of a cat by* (a) *blurring the cat image and* (b) *enhancing the resulting image to get a smooth silhouette: "somewhat as if the photograph of the object were blurred and then printed on high-contrast paper."*

for Figure 1.2. *It was said earlier that the points of the drawing corresponded to places of maximum curvature on the contour of the cat, but this was not strictly correct; if the principle had been followed rigidly, it would have been necessary to represent the ends of individual hairs by points. In observing a cat, however, one does not ordinarily perceive its hairs as individual entities; instead one perceives that the cat is furry. (...) The perceived contour of a cat (e.g., the contour from which the points of Figure* 1.2 *were taken) is the resultant of an orthogonal averaging process in which texture is eliminated or smoothed out almost entirely, somewhat as if* **a photograph of the object were blurred and then printed on high-contrast paper***."*

A correct shape encoding, which does not get lost in textural details, asks for a previous blurring. The process suggested (in bold in the citation) is to convolve the silhouette of the object ("blurred") and then to enhance the result, which amounts to thresholding the image ("high-contrast paper") (see Figure 2.5). This algorithm is by now known as the Bence, Merriman, and Osher algorithm [36] (an early version of the algorithm is due to Koenderink and van Doorn [28]) and was proved to be equivalent to the curvature motion [7, 19]:

$$\frac{\partial u}{\partial t} = |Du|\mathrm{curv}(u). \tag{2.2}$$

3. Affine invariant mathematical morphology and affine scale space. The general process by which an image or a shape is smoothed at several scales in order to eliminate spurious or textural details and extract its main features is called "scale space." The main developments of scale space theory in the past 10 years involve invariance arguments: indeed, a scale space will be useful to compute invariant information only if it is itself invariant. Let us summarize a series of arguments given (e.g.) in [2]: a scale space computing contrast invariant information must in fact deal directly with the image level lines; in order to be local (not dependent upon occlusions), it must in fact be a PDE. In order to be a smoothing, this PDE must be of a parabolic kind. Then, the further affine invariance requirement and the invariance with respect to reverse contrast (if we want a self-dual operator, in the mathematical morphology terminology [47]) lead to a single PDE,

$$\frac{\partial u}{\partial t} = |Du|\mathrm{curv}(u)^{\frac{1}{3}}, \tag{3.1}$$

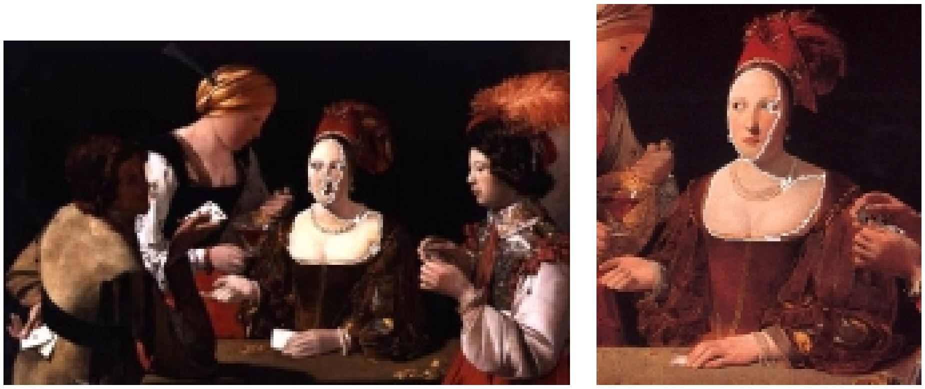

FIG. 3.1. *Two similar level lines (in white) in the paintings of Figure* 2.1. *How similar they locally are will be decided by an algorithm performing affine scale space and local affine invariant comparison.*

where Du denotes the gradient of the image, $\mathrm{curv}(u)$ denotes the curvature of the level line, t denotes the scale parameter, and the power $\frac{1}{3}$ is signed, i.e., $s^{\frac{1}{3}} = sign(s)|s|^{\frac{1}{3}}$. This equation is equivalent to the "affine curve shortening" [46] of all of the level lines of the image, given by the equation

$$\frac{\partial x}{\partial t} = |\mathrm{Curv}(x)|^{\frac{1}{3}}\vec{n}, \tag{3.2}$$

where x denotes a point of a level line, $\mathrm{Curv}(x)$ denotes its curvature, and $\vec{n}$ denotes the signed normal to the curve, always pointing towards the concavity.

This equation is the only possible smoothing under the invariance requirements mentioned above. This gives a helpless bottleneck to the local shape recognition problem, since it is easily checked [2] that no further invariance requirement is possible: (3.1) is the only affine invariant local contrast invariant smoothing. In particular, despite some interesting attempts [20], there is no practical way to define a projective invariant local smoothing.[2]

The use of curvature-based smoothing for shape analysis is well established. Founding papers are [4, 39, 17]. These authors define a multiscale curvature which is similarity invariant but not affine invariant. Abbasi and Mokhtarian [1] used the curvature motion (2.2) and an affine length parameterization of the boundary of the shapes in order to get an affine shape encoding. Of course, such a coding cannot be fully affine invariant, since the curvature motion is not affine invariant. It is much more natural to use directly (3.1).

An algorithm performing affine scale space and local affine invariant comparison will allow us to measure the similarity between any two given curves, like the ones displayed in Figure 3.1.

3.1. Affine erosions and dilations. In this subsection we shall describe a practical derivation of the affine invariant smoothing, which is quite useful for the design

[2] A recent paper [15] proposes, however, a new model for the motion in a planar image which is consistent with the displacement of a pinhole camera. By using the 3D representation of a projective deformation and the reciprocity principle, a new group with six parameters (the registration group) is proposed instead of the projective group (eight parameters) for which multiscale smoothing can be performed.

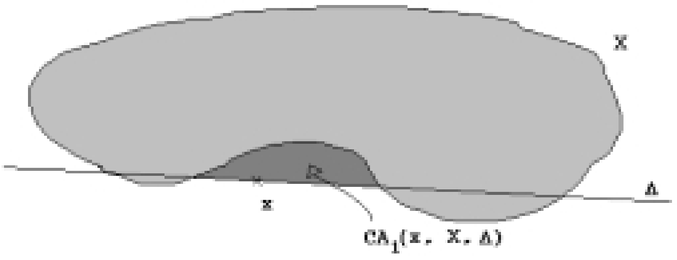

FIG. 3.2. *Affine distance.*

of a fast algorithm. In order to do so, we go back to the mathematical morphology formalism [47, 35] and define first an affine distance and then affine invariant set erosions and dilations. We shall prove how from a very simple (probably the simplest possible) definition of affine invariant distance of a point to a set we are in a position to design fast affine invariant set erosions and dilations. These filters are consistent with (3.1) and yield a natural formal derivation for a fast algorithm introduced by Moisan [38] and described in section 6. This section follows the general line of the book in preparation [23].

We consider ways to erode or dilate a shape in an affine invariant way.

Let $A = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)$ be an arbitrary matrix such that $\det A = ad - cb = 1$. The set of such matrices is the so-called special linear group, $SL(\mathbb{R}^2)$. We say that an operator T is special affine invariant if T commutes with A for every A in $SL(\mathbb{R}^2)$: $AT = TA$.

We first define an "affine invariant distance" which will be a substitute to the classical euclidean one. We consider shapes X, that is, in whole generality, closed nonempty subsets of $\mathbb{R}^2$. Let $x \in \mathbb{R}^2$, and let Δ be an arbitrary straight line passing by x. We consider all connected components of $\mathbb{R}^2 \setminus (X \cup \Delta)$. If $x \notin X$, two and only two of them contain x in their boundary. We denote them by $\mathrm{CA}_1(x, \Delta, X)$, $\mathrm{CA}_2(x, \Delta, X)$ (see Figure 3.2). We call these sets the "chord-arc sets" defined by x, Δ, and X, and we order them so that $\mathrm{area}(\mathrm{CA}_1(x, \Delta, X)) \leq \mathrm{area}(\mathrm{CA}_2(x, \Delta, X))$.

DEFINITION 3.1. *Let X be a "shape" and $x \in \mathbb{R}^2, x \notin X$. We call affine distance of x to X the real number $\delta(x, X) = \inf_\Delta \mathrm{area}(\mathrm{CA}_1(x, \Delta, X))^{1/2}$, $\delta(x, X) = 0$ if $x \in X$.*

Remark. Obviously, we take the power $1/2$ in order that the affine distance be homogeneous to a length. The affine distance can be infinite: take, e.g., a convex set X and x outside X. Then it is easily seen that $\delta(x, X) = +\infty$ because all chord-arc sets defined by all straight lines Δ are unbounded.

DEFINITION 3.2. *Let X be a shape, i.e., a closed nonempty subset of $\mathbb{R}^2$. We call affine a-dilate of a set X the set $\tilde{D}_a X = \{x, \delta(x, X) \leq a^{1/2}\}$. We call affine a-eroded of set X the set $\tilde{E}_a X = \{x, \delta(x, X^c) > a^{1/2}\} = (\tilde{D}_a X^c)^c$.*

PROPOSITION 3.3. *The affine invariant erosions and dilations $\tilde{E}_a$ and $\tilde{D}_a$ are special affine invariant monotone operators.*

Proof. It is easily seen that if $X \subset Y$, then for every x, $\delta(x, X) \geq \delta(x, Y)$. From

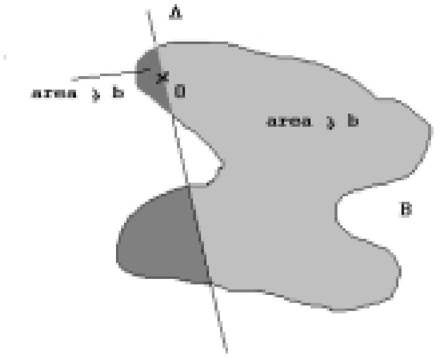

Fig. 3.3. *An affine structuring element: all lines passing by* 0 *divide* B *into several connected components. All of them which contain* 0 *in their boundary have an area larger than or equal to* b.

this, we deduce that $X \subset Y \Rightarrow \tilde{D}_a X \subset \tilde{D}_a Y$. The monotonicity of $\tilde{E}_a$ follows by the duality relation $\tilde{E}_a X = (\tilde{D}_a X^c)^c$. The special affine invariance of $\tilde{D}_a$ and $\tilde{E}_a$ follows from the fact that if $\det A = 1$, then $\mathrm{area}(X) = \mathrm{area}(AX)$. □

We shall now use Matheron theorem (Theorem 6.2 in [23]) in order to give a standard form to $\tilde{E}_a$ and $\tilde{D}_a$.

Theorem 3.4 (Matheron). *Let* T *be a translation invariant monotone operator acting on a set of subsets of* $\mathbb{R}^N$. *Then, there exists a family of sets* $\mathbb{B} \subset \mathcal{P}(\mathbb{R}^N)$, *which can be defined as* $\mathbb{B} = \{X, 0 \in T(X)\}$, *such that*

$$T(X) = \bigcup_{B \in \mathbb{B}} \bigcap_{y \in B} X - y = \{x, \exists B \in \mathbb{B}, x + B \subset X\}.$$

The sets $B \in \mathbb{B}$ are called "structuring elements" of the operator T. Let us start by defining structuring elements adapted to $\tilde{E}_a$.

Definition 3.5. *We say that* B *is an affine structuring element if* B *is a set whose interior contains* 0 *and if there is some* $b > 1$ *such that for every line* Δ *containing* 0 *both connected components of* $B \setminus \Delta$ *containing* 0 *in their boundary have an area larger or equal to* b *(see Figure* 3.3*). We denote the set of affine structuring elements by* $\mathbb{B}_{\mathrm{aff}}$.

Proposition 3.6. *For every set* X,

$$\tilde{E}_a X = \bigcup_{B \in \mathbb{B}_{\mathrm{aff}}} \bigcap_{y \in a^{1/2} B} X - y = \{x, \exists B \in \mathbb{B}_{\mathrm{aff}}, x + a^{1/2} B \subset X\}.$$

Proof. We simply apply Theorem 3.4. The set of structuring elements associated with $\tilde{E}_a$ is $\mathbb{B} = \{X, \tilde{E}_a X \ni 0\}$. Now,

$$\tilde{E}_a X \ni 0 \Leftrightarrow \delta(0, X^c) > a^{1/2} \Leftrightarrow \inf_\Delta \mathrm{area}(\mathrm{CA}_1(0, \Delta, X))^{1/2} > a^{1/2}.$$

This means that for every Δ, both connected components of $X \setminus \Delta$ containing 0 have an area larger than some number $b > a$. Thus, X belongs to $a^{1/2}\mathbb{B}_{\mathrm{aff}}$ by the definition of $\mathbb{B}_{\mathrm{aff}}$. □

By Proposition 3.6, x belongs to $\tilde{E}_a X$ if and only if for every straight line Δ chord-arc sets containing x have an area strictly larger than a.

Conversely we can state the following corollary.

Corollary 3.7. *$\tilde{E}_a X$ can be obtained from X by removing, for every straight line Δ, all chord-arc sets contained in X which have an area smaller than or equal to a.*

Hence, Definitions 3.2 and 3.5 are equivalent to Definition 2 in [38]. As we shall see in the next subsection, alternating affine erosions and dilations on the set X surrounded by a Jordan curve yields a numerical scheme that computes the affine shortening (3.2) of the curve. In section 6, we shall also explain how this scheme is implemented.

3.2. Consistency of the affine erosion-dilation scheme with the affine invariant PDE. The main difficulty for showing the consistency of a fully affine invariant scheme with a PDE lies in its nonlocality. Indeed, the set of affine structuring elements contains stretched sets of any size. The arguments we now develop permit us to localize affine invariant sets of structuring elements. This localization is the key point to prove the announced consistency.

Definition and Proposition 3.8. *We shall say that $\mathbb{B}$ is localizable if it is made of compact connected sets containing 0 and if there exists a constant $c > 0$ such that for every $\rho > c$ we can assert that $\forall B \in \mathbb{B}, \exists B' \in \mathbb{B}, B' \subset D(0,\rho)$, and $B' \subset D_{\frac{c}{\rho}}(B) = \{x, d(x,B) \leq \frac{c}{\rho}\}$, where d denotes the euclidean distance, $d(x,B) = \inf_{y \in B} d(x,y)$.*

As a consequence, if we define $\mathbb{B}_s = \{s^{1/2}B, B \in \mathbb{B}\}$, we also have $\exists c > 0, \forall s \leq c^{-1}r^2, \forall B \in \mathbb{B}_s, \exists B' \in \mathbb{B}_s, B' \subset D(0,r)$, and $B' \subset D_{\frac{cs}{r}}(B)$.

Proof. In order to deduce the second relation from the first, we simply set $r = \rho s^{1/2}$. We have $B \in \mathbb{B}$ if and only if $s^{1/2}B \in \mathbb{B}_s$. We therefore replace B by $s^{1/2}B$ and B' by $s^{1/2}B'$ and we get for the new B and B' in $\mathbb{B}_s$

$$d(B,B') \leq \frac{cs^{1/2}}{\rho} = \frac{cs}{r},$$

provided $\rho > c$, i.e., $r > cs^{1/2}$ or $s < c^{-2}r^2$. □

Proposition 3.9. *Let $\mathbb{B}$ be made of subsets of $\mathbb{R}^2$ containing 0. Assume that there exists $c > 0$ such that if $B \in \mathbb{B}$ and $r > c$, then the connected component of $D_{\frac{1}{r}}(B) \cap D(0,r)$ containing 0 is in $\mathbb{B}$ (resp., contains an element of $\mathbb{B}$). Then $\mathbb{B}$ is localizable (we denote by $D_{\frac{1}{r}}(B)$ the dilate of B, $\{x, d(x,B) \leq \frac{1}{r}\}$).*

Proof. For any B in $\mathbb{B}$, we consider B', the connected component of $D_{\frac{1}{r}}(B) \cap D(0,r)$ containing 0. In the second case, we consider an element B' of $\mathbb{B}$ contained in this connected component. □

Proposition 3.10. *If $\mathbb{B} = \mathbb{B}_{\text{aff}}$ is the set of all chord-arc sets, then $\mathbb{B}$ is localizable.*

Proof. We want to apply Proposition 3.9. Let B' be the connected component of $D_{\frac{2b}{r}}(B) \cap D(0,r)$ containing 0. Let Δ be a line passing by 0 (see Figure 3.4). We consider one of the two connected components $\mathcal{C}'$ of $B' \setminus \Delta$ containing 0 in their boundary. We also consider $\mathcal{C}$ the connected component of $B \setminus \Delta$ containing 0 in their boundary and such that $\mathcal{C} \cap D(0,r) \subset B'$.

Two cases: if $\mathcal{C}$ is contained in $D(0,r)$, then by definition of chord-arc sets the area of $\mathcal{C}$ is larger than b. Therefore, the area of $\mathcal{C}'$ is larger than b.

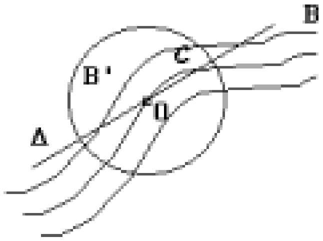

FIG. 3.4. *Proof of Proposition* 3.10.

Second case: if $\mathcal{C}$ is not contained in $D(0,r)$, we consider the connected component $\mathcal{C}_1$ of $\mathcal{C} \setminus \partial D(0, r - \frac{2b}{r})$ which contains 0. $\mathcal{C}_1 \subset \mathcal{C}$ meets $\partial D(0, r - \frac{2b}{r})$ at some points; then, by connectedness, each line orthogonal to $[0,x]$ and passing by tx, with $0 \leq t \leq 1$, meets $\mathcal{C}_1$ at at least one point $x(t) \in D(0, r - \frac{2b}{r})$. Noting $\nu = \frac{x^\perp}{\|x\|}$, a unit vector orthogonal to x, we notice that the interval $[x(t) - \frac{2\nu}{r}, x(t) + \frac{2\nu}{r}]$ is contained in $D_{\frac{2b}{r}}(\mathcal{C}_1) \subset D_{\frac{2b}{r}}(\mathcal{C}) \subset B'$. In addition, at least one half of this interval is contained in $\mathcal{C}'$. Thus, provided $r \geq 2b$, $\text{area}(\mathcal{C}') \geq (r - \frac{2b}{r})\frac{2b}{r} = 2b - \frac{4b^2}{r^2} \geq b$. □

The relevance of Proposition 3.10 is explained by the following theorem (see [23]).

THEOREM 3.11. *Assume that $\mathbb{B}$ is a localizable affine invariant set of subsets of $\mathbb{R}^2$. Set $\mathbb{B}_s = s^{\frac{1}{2}}\mathbb{B}$ its scaled version, with $s \to 0$. Consider the alternate operator IS_sSI_s, where for any real valued image $u(x)$ on the plane*

$$IS_s u(x) = \inf_{B \in \mathbb{B}_s} \sup_{y \in B} u(x+y), \quad SI_s u(x) = \sup_{B \in \mathbb{B}_s} \inf_{y \in B} u(x+y).$$

Then, there exists a constant $c_{\mathbb{B}} \geq 0$ such that for every C^3 function $u(x)$ one has

$$\lim_{s \to 0} \frac{IS_sSI_s u(x) - u(x)}{s^{\frac{2}{3}}} = c_{\mathbb{B}} |Du| (\text{curv}(u)(x))^{\frac{1}{3}}.$$

The relevance of this theorem can be explained as follows. Applying the alternate scheme IS_sSI_s to u is equivalent to applying alternate affine erosions and dilations to each level curve of u. Thus, Theorem 3.11 means that this algorithm moves these curves, as we take s small and iterate the alternate scheme, according to (3.2) (indeed, as we mentioned above, it is equivalent to moving all level curves of u by this equation or to move u by (3.1)) (see Figure 3.5). We have indicated the main steps and refer to [23] for a detailed exposition of the mathematical framework for these results.

Remarks. In this section it has been proved that if we apply (3.1) to an image and then level lines are extracted, we get the same result as when extracting level lines of the original image and then applying the affine erosion-dilation scheme on them. When dealing with digital images, this theoretical equivalence no longer holds due to the sampling effects. All finite difference schemes implementing (3.1) do not commute with contrast changes; they create new grey levels and a blur effect. In consequence, new level lines are created and the blur induces wrong connections between level lines. Now, it is extremely important to have a scheme as accurate and consistent as possible to get a high quality shape matching. Experience proves that this accuracy is obtained only by a fully invariant scheme as the affine erosion-dilation scheme we have

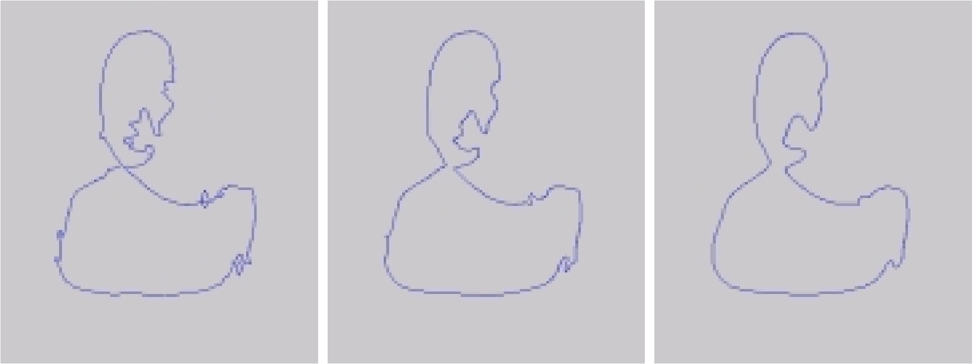

Fig. 3.5. *Evolution of the selected curve in the first image of Figure* 3.1 *under increasing scales of filtering (alternate affine erosions and dilations). From left to right: original curve, eroded area* = 1*, and eroded area* = 5*. This numerical scheme is consistent with the affine shortening equation* (3.2).

introduced. It may be objected that this scheme is computationally heavy. This is not the case; as we comment further in our remarks on complexity at the end of subsection 4.2, the complexity of the filtering algorithm is proportional to the total variation of the image; the complexity of a good difference scheme would be proportional to the image size. The complexity ratio between both is about 10, but this does not matter since the heavy term in the final complexity is due to code matching and not to filtering.

4. Shape local affine invariant encoding.

4.1. From global to local recognition methods. In global recognition methods, the shape is considered as a whole and is described by a sequence of characteristics of the shapes [43, 16], such as perimeter, algebraic moments [48, 25, 40], or Fourier coefficients [29, 32, 42]. A comparison of two shapes is led back to the comparison of the vectors of computed characteristics. Most of the methods we just mentioned are Euclidean but not affine invariant. It is possible, however, to compute affine invariant moments. In that case, we face another difficulty: how do we define the relative weights of each moment in a shape comparison distance? This difficulty is overcome by the more clever *normalization methods* [12]. Normalization methods allow the transformation of any element of an equivalence class of shapes under a group of geometric transforms into a specific one, fixed once forever in each class. A good account of global normalization methods can be found in [43]. Affine normalization [25] can be used as a tool to match both final invariance requirements on shape recognition, namely the locality (robustness to occlusion) and the affine invariance (invariance with respect to orthoprojections). Indeed, the affine invariance entails an affine normalization of the level lines. In addition, the robustness under partial occlusions implies that the normalization of the level lines must be done with respect to several local reference systems. Thus, several pieces of the same level line are normalized by using different reference systems, providing a local and redundant description of the level line.

Affine invariant robust semilocal descriptors are given by the lines which are bitangent to the curve. These descriptors will be used as starting points for an invariant sampling of the set of the tangent lines to the curve. A bitangent to a given curve is

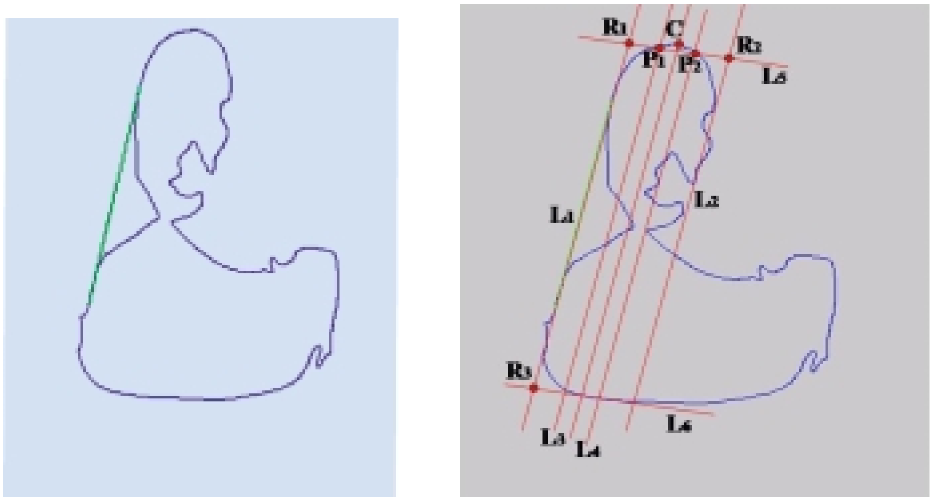

Fig. 4.1. *Affine invariant encoding of a piece of curve. Left: a bitangent found in the selected curve from the first image (after affine erosion with area = 1). Right: description of the encoding method: L_2 is a tangent to the curve parallel to the original bitangent (L_1). If we call d the distance from L_1 to L_2, lines L_3 and L_4 are parallel to L_1 and at distances $d/3$ and $2d/3$ from L_1, respectively. Points P_1 and P_2 are the intersection of L_3 and L_4 with the curve, respectively. L_5 is a line passing through P_1 and P_2, and L_6 is a tangent to the curve parallel to L_5. The reference points R_i for the normalization are the intersections of lines L_1 and L_5 (R_1), L_2 and L_5 (R_2), and L_1 and L_6 (R_3).*

any straight line tangent to the curve at two different points.

From each bitangent, one can define a local system of coordinates. These coordinate systems shall be invariant under affine transforms, and they will be used to normalize a portion of the curve. This method is systematically used in [31] and in the recent shape recognition prototype described in the book [44]. As pointed out in this last reference, an affine normalization of a curve needs three affine invariant reference points (see Figure 4.1). We describe in the following such an algorithm, which we tested extensively [33].

4.2. Local affine normalization and coding algorithm.

1. Orient each level curve in a unambiguous way, e.g., by the Maxwell rule. Thus, given a point x on the curve, we can talk about the "next" point on the curve having some property. For each bitangent L_1 to the curve, we consider the next tangent L_2 to the curve which is parallel to L_1.

2. Call L_3 and L_4 the straight lines parallel to L_1 located between L_1 and L_2, at distances $\frac{1}{3}d$ and $\frac{2}{3}d$ from L_1, d being the distance between L_1 and L_2.

3. Consider the intersection points P_1 and P_2 between L_3 and L_4 and the portion of the curve limited by the tangency points of L_1 and L_2.

4. Call L_5 the line passing through P_1 and P_2 and find the previous tangent to the curve parallel to this line (call it L_6).

5. Find the intersection points between L_1 and L_5, L_2 and L_5, and L_1 and L_6; call them R_1, R_2, and R_3 respectively.

6. Affine normalization: R_1, R_2, and R_3 form an affine reference system since they can be mapped to the triangle $(0,0)-(1,0)-(0,1)$ of the plane. The rest of the curve can be normalized according to this mapping.

7. Encoding of a piece of curve: the portion of the curve to be normalized is proportional to the distance between R_1 and R_2 (the proportionality factor is typically set to 3). The normalized portion of the curve is sampled with a fixed number p of points (typically 9), equidistant along the normalized curve. The central point of the normalization is the intersection point between the piece of curve limited by the tangency points of L_1 and L_2 and the straight line parallel to L_1 and equidistant from L_1 and L_2 (C). We take $\frac{p-1}{2}$ samples on both sides of this central point. The set of $2p$ normalized point coordinates form the affine invariant code of the piece of curve.

To summarize, an algorithm performing shape comparison between two images will proceed with the following steps:

1. Extraction of *all* the level lines for each image.
2. Affine filtering of the extracted level lines at several scales.
3. Local encoding of pieces of level lines after affine normalization.
4. Comparison of the vectors of features of the images.

Since the scale space filtering is affine invariant, the geometric transform between two images can be correctly estimated in spite of the different scales of smoothing which have been performed.

Remarks on the computational cost of the algorithm. We shall give an estimate in the case where all level lines of the image are considered. The number of level lines of the image is finite because of the grey level quantization effect, which allows one to take only integer levels. In that case, the total length of the level lines is equal to the total variation of the image, which we denote in the following by $TV(u) = \int |\nabla u|$. Now, it is always possible to subquantize the image without much decay in recognition performance. We also mentioned a method to extract the most meaningful level lines. In practice, this means that we consider only a proportion $\epsilon_1 TV$ of the whole total variation. Usually, $\epsilon_1 = \frac{1}{10}$. We shall give the complexity in terms of TV, which is usually proportional to the image size (in practice, the total variation of most digital images is about 10 times their size).

We give in the following an estimate for each step of the matching algorithm, assuming that we compare images with the same order of magnitudes for sizes and total variation.

- Extraction of the level lines: $\epsilon_1 TV$.
- Filtering of the level lines: $C\epsilon_1 TV$, where C is proportional to the (usually fixed) smoothing scale and is larger than 1. The order of magnitude of C is 30.
- Level line encoding. The number of codes is proportional to the length of the level lines, which yields another $\epsilon_2 TV$ and ϵ_2 is small with respect to 1 (about $\frac{1}{10}$).
- Comparison of codes. This is proportional to $\epsilon_2^2 \epsilon_1^2 TV^2$.

In conclusion, the complexity of the described algorithm is $O(C\epsilon_1 TV + \epsilon_1^2 \epsilon_2^2 TV^2)$, which yields with realistic values, comparing two images of size 512 × 512:

- $TV = 3.10^6$.
- $C\epsilon_1 TV = 10^7$.
- $\epsilon_1^2 \epsilon_2^2 TV^2 = 10^9$.

Thus the estimated complexity is 10^9 operations, which can be performed in times less than one minute. This time corresponds to the practice with a personal computer having a 1 Gigahertz microprocessor. Let us remark, however, that filtering reduces the number of bitangents in the curve. Thus, higher speed is attained by more filtering, and this is the main reason for the necessity of a filtering step.

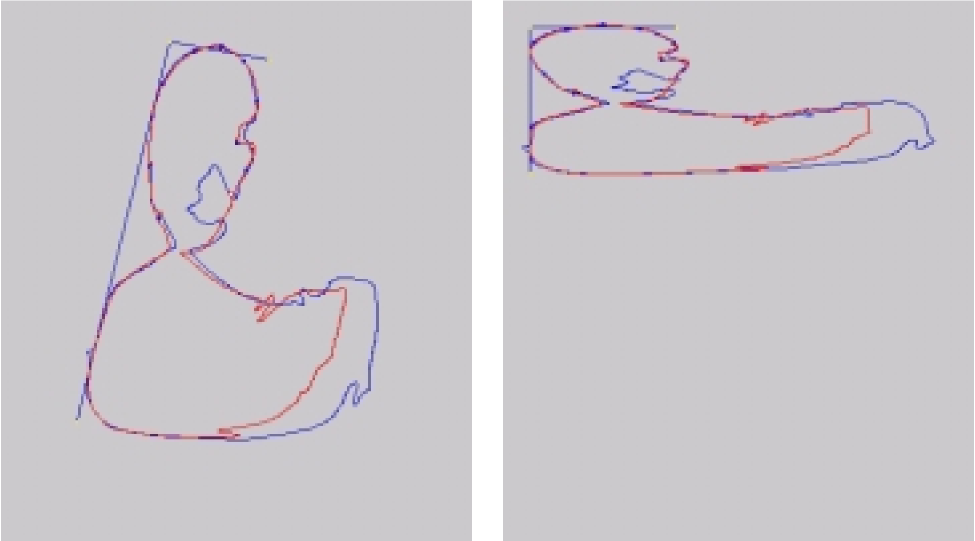

Fig. 5.1. *Matching between the selected curves in Figure* 3.1. *Left: both curves are displayed under the same reference system. Right: curves displayed under the normalized reference system. They coincide very accurately in a long portion.*

5. Experimental results in image comparison. According to the preceding description, a generic image comparison algorithm works as follows. For each quantized code of the first image do the following:

1. Search in turn each value of the code describing the curve. Since codes are ordered, the search is fast and generates pairs of candidate matches.
2. Compute the actual distance between them. Reject the matching if this distance is larger than some threshold (usually one pixel).
3. Extend the matching beyond the initial portion of curves, provided that the distance between the corresponding points in the curves is below the distance threshold.

Figure 5.1 displays an example of the partial matching between two level lines coming from different images (the ones in Figure 3.1).

The information provided by the image comparison algorithm gives a very accurate local estimation of the matching of some level lines of the image (the ones which could be coded). In most cases a dominant motion can be estimated from all the local motions of the matched level lines.

As a first test, we show in Figure 5.2 (in white) all pieces of level lines which matched accurately between both paintings of de la Tour (Figure 2.1). The knowledge of these roughly 50 codes permits us to compute an accurate affine transform between both images.

The question of comparing and registering two images with different illuminations and viewpoints is essential in satellite imaging. Figure 5.3 shows, as an extreme test, two images corresponding to different channels of a satellite image (green and infrared). Both images have a small common part.

From the motion information provided by the found matching pairs of pieces of level curves, a mosaic can be constructed that combines the information of both images (Figure 5.4).

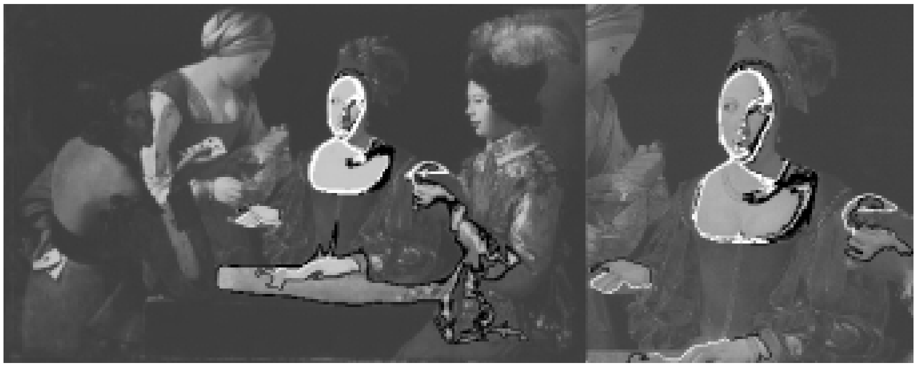

FIG. 5.2. *All pieces of level lines matching accurately between both Georges de la Tour paintings.*

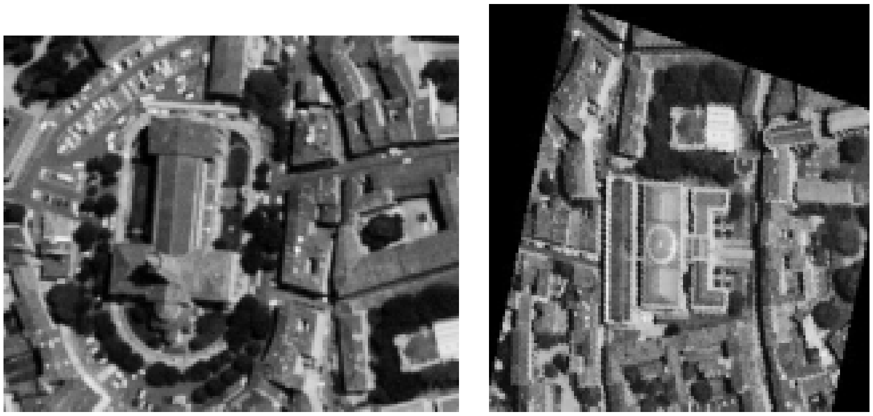

FIG. 5.3. *Satellite images to be registered (CNES database). Remark that there is a common part between the images (the lower right corner of the first image).*

As a third test, we used two very different images that display the same logo (Superman logo, Figure 5.5). One of the images is a drawing of the logo and the second one is a snapshot of an alarm clock with the logo printed on it. Remark that in this second image there is an occlusion effect due to the fingers of the clock and an unknown geometrical transformation between the original logo and the one in the snapshot. The found matchings between both images are shown in Figure 5.6, and the result of applying the estimated affine transform to the second image is displayed in Figure 5.7.

5.1. Conclusions and bibliographical notes. The shape recognition problem has been the subject of years of intense research. As we mentioned in the introduction, the literature on the subject is huge, and it is impossible to give a complete account of it. Thus, our strategy in this paper has been to refer mainly to references proposing general shape recognition principles.

We have focused on invariant planar shape recognition, which is afterall a tiny part of the gigantic problem of 3D shape recognition. Now, on this problem, enough progress has been done, both on the principles and on their applicability to a *generic shape recognition algorithm*. In this paper, we have therefore tried to prove the existence of at least one fully generic shape recognition algorithm. This algorithm is fully

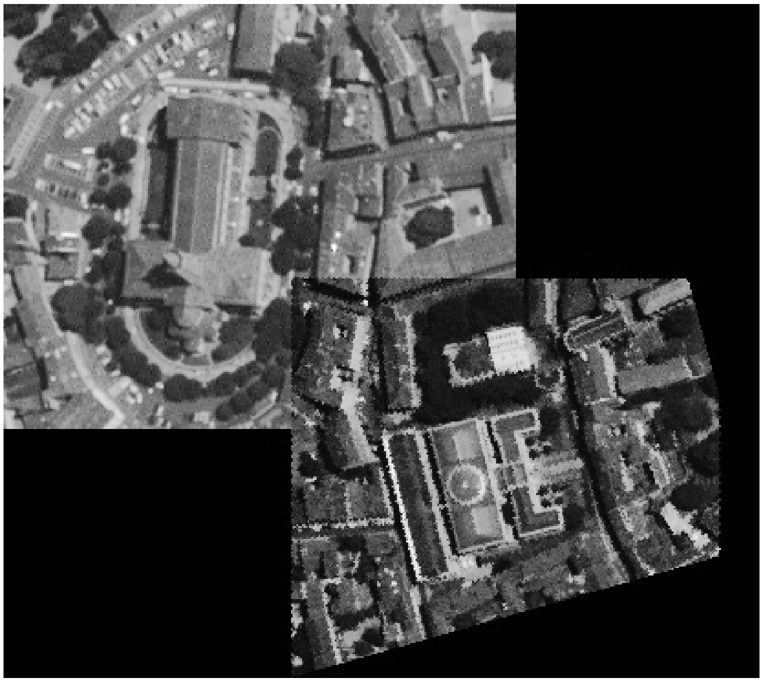

Fig. 5.4. *Mosaic for the images in Figure* 5.3.

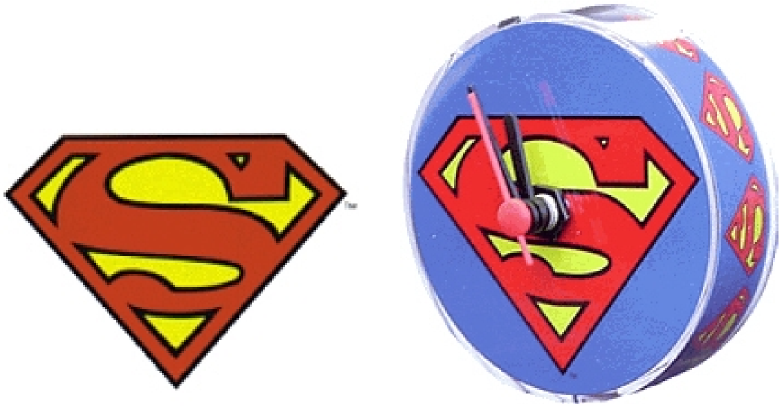

Fig. 5.5. *Original Superman logo images. Superman logo* TM *and* © *DC Comics. All rights reserved. Used with permission.*

"principle-based." To the best of our knowledge, there is no fully generic and fully invariant shape recognition algorithm, either published or disclosed as software. By generic algorithm we mean an algorithm taking any two images and yielding all shapes common to both. This paper describes one such algorithm, yielding for any pair of images a list of matching shapes, depending on a single distance threshold. As we have tried to show, there is no fully new idea or principle involved in such an algorithm. It is the mere concatenation of algorithmic consequences of invariance principles. We have in place quoted as well as we could the discoverers of those principles.

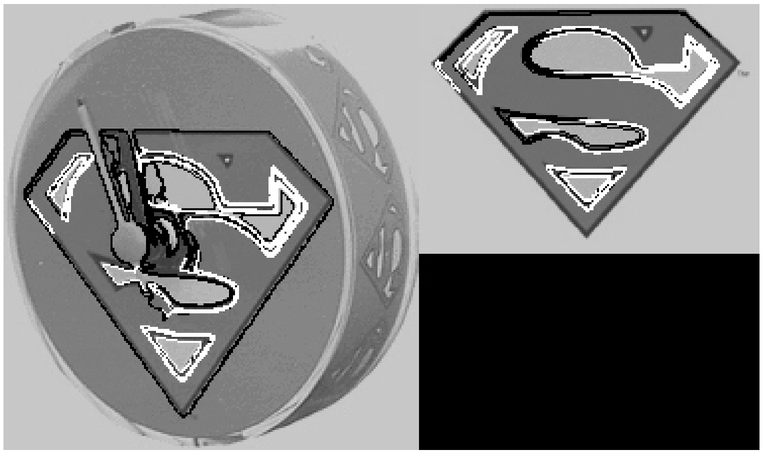

FIG. 5.6. *Matching pieces of level lines in Superman logos.*

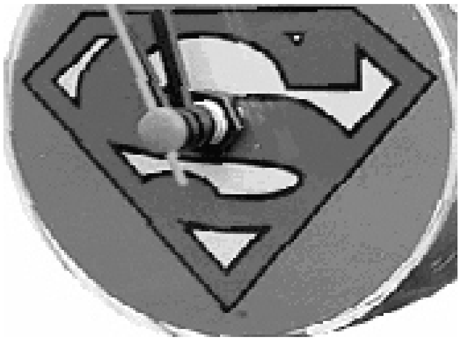

FIG. 5.7. *Figure* 5.5 *right after applying the affine transform estimated from the pairs of matching pieces of curves.*

There are instead many dedicated *object recognition* algorithms, whose aim it is to solve one or the other practical problem. We have presented registration results (Figures 5.3 and 5.4), where registration is obtained as a by-product of shape recognition. Let us mention (e.g.) [9], which gives a survey on image registration. There are indeed other possible, and simpler, registration techniques, since the registration problem amounts to estimating between six and 12 motion parameters. See, e.g., the excellent registration results given in [30] and the David Sarnoff group. These results are not obtained by shape recognition techniques and, actually, the technique giving such remarkable results is not really explained.

As far as we can gather, the technique is not contrast invariant, and neither is it for most of the dedicated or generic algorithms usually proposed. As an example of a finalized algorithm, let us mention the one described in the Rothwell book [44]. His algorithm is very close in all aspects to the one we described here, and it was explicitly designed to the same aim of showing the existence of a finalized shape recognition algorithm. Now, the shapes treated by Rothwell are a few flat tools occluding each other and for which a model is available. Thus, the problem treated is not *shape recognition* but again *object recognition*; thus in many aspects it is specifically designed for the chosen objects.

There are, however, generic shape recognition algorithms, which stem from the founding paper by Wolfson [50] under the generic name of geometric hashing. Let us mention [8] as a very recent extension of this technique. The idea of geometric hashing is this: a shape is described by local descriptors, usually points with an orientation (minutiae in the fingerprint recognition terminology), or corners [13]. Thus, the shape matching problem can be reduced to the problem of finding a given configuration of minutiae, describing a target shape, in a clutter of minutiae computed in another image. In the affine invariant case, the complexity of such a method grows as N^3, where N is the number of minutiae in the explored image. The reason is that no order is given between the minutiae, and any group or subgroup can belong or not belong to a given shape. Our aim here has been to demonstrate that one can single out preformed chains of minutiae in an image, namely the encoded pieces of level lines. The existence of these chains reduces considerably the complexity of a shape search, since a single chain may be enough to characterize a target shape, when it is complex enough.

Most techniques used in *object recognition* are not invariant enough. For instance the very interesting method of [1] includes a fair part of what we presented here (curvature scale space, similarity invariance), but this paper does not explain how to extract the shapes in generic images.

As a first conclusion, we would rather promote *shape recognition* as the main problem, as opposed to the rather infinite *object recognition* problem. While the first one can receive a generic and general answer by invariance arguments, the second relies on specificities of the objects to recognize and leads to as many algorithms as objects. As a second conclusion, the generic shape recognition algorithms like geometric hashing have suffered from their lack of accuracy and invariance in the starting points, namely the minutiae. Thus, the contribution of this survey, if any, is to promote very accurate and invariant steps in the first stages of any recognition algorithm.

6. A fast invariant curve affine erosion-dilation scheme.

A fast algorithm. In general, the affine erosion of X is not simple to compute, because it can be strongly nonlocal. However, if X is convex, then it has been shown in [38] that it can be exactly computed in linear time. In practice, c will be a polygon and the exact affine erosion of X—whose boundary is made of straight segments and pieces of hyperbolae—is not really needed; numerically, a good approximation by a new polygon is enough. Now the point is that we can approximate the combination of an affine erosion plus an affine dilation of X by computing the affine erosion of each *convex component* of c, provided that the erosion-dilation area is small enough.

The algorithm consists of the iteration of a four-step process:

1. *Break the curve into convex components.* This operation permits us to apply the affine erosion to convex pieces of curves, which is much faster (the complexity is linear) and can be done simply in a discrete way. The main point is to take into account the finite precision of the computer in order to avoid spurious (small and almost straight) convex components only due to numerical artifacts.

2. *Sample each component.* At this stage, points are removed or added in order to guarantee an optimal representation of the curve that is preserved by step 3.

3. *Apply discrete affine erosion to each component.*

4. *Concatenate the pieces of curves obtained at step* 3. This way, we obtain a new closed curve on which the whole process can be applied again.

6.1. Extracting convex components. We start with a closed polygonal curve $P_0P_1\dots P_{n-1}$, with the convention that $P_{i+n}=P_i$ and the notation $P_i=(x_i,y_i)$.

The curve has to be broken at points where the sign of the determinant

$$d_i=[P_{i-1}P_i,P_iP_{i+1}]$$

changes. Numerically, we use the formula

$$d_i=(x_i-x_{i-1})(y_{i+1}-y_i)-(y_i-y_{i-1})(x_{i+1}-x_i). \tag{6.1}$$

Since we are interested in the sign of d_i, we must be careful because the finite numerical precision of the computer can make this sign wrong. Let us introduce the relative precision of the computer

$$\varepsilon_0=\max\{x>0,\ (1.0\oplus x)\ominus 1.0=0.0\}. \tag{6.2}$$

In this definition, $\oplus$ (resp., $\ominus$) represent the computer addition (resp., substraction), which is not associative. When computing d_i using (6.1), the computer gives a result $\tilde{d_i}$ such that $|d_i-\tilde{d_i}|\le e_i$, with

$$\begin{aligned}e_i=\varepsilon_0\big(&|x_i-x_{i-1}|(|y_{i+1}|+|y_i|)+(|x_i|+|x_{i-1}|)|y_{i+1}-y_i|\\&+\ |y_i-y_{i-1}|(|x_{i+1}|+|x_i|)+(|y_i|+|y_{i-1}|)|x_{i+1}-x_i|\big).\end{aligned}$$

In practice, we take ε_0 a little bit larger than its theoretical value to overcome other possible errors (in particular, errors in the computation of e_i). For four-bytes C *float* numbers, we use $\varepsilon_0=10^{-7}$, whereas the theoretical value (that can be checked experimentally using (6.2)) is $\varepsilon_0=2^{-24}\simeq 5.96\ 10^{-8}$. For eight-bytes C *double* numbers, the correct value would be $\varepsilon_0=2^{-53}\simeq 1.11\ 10^{-16}$.

The algorithm that breaks the polygonal curve into convex components consists of the iteration of the following decision rule:

1. If $|\tilde{d_i}|\le e_i$, then remove P_i (which means that the new polygon to be considered from this point is $P_0P_1\dots P_{i-1}P_{i+1}\dots P_{n-1}$).
2. If $|\tilde{d}_{i+1}|\le e_{i+1}$, then remove P_{i+1}.
3. If $\tilde{d_i}$ and $\tilde{d}_{i+1}$ have opposite signs, then the middle of P_i,P_{i+1} is an inflexion point where the curve must be broken.
4. If $\tilde{d_i}$ and $\tilde{d}_{i+1}$ have the same sign, then increment i.

This operation is performed until the whole curve has been visited. The result is a chained (looping) list of convex pieces of curves.

6.2. Sampling. At this stage, we add or remove points from each polygonal curve in order to ensure that the euclidean distance between two successive points lies between ε and 2ε (ε being the absolute space precision parameter of the algorithm).

6.3. Discrete affine erosion. This is the main step of the algorithm: compute quickly an approximation of the affine erosion of scale σ of the whole curve.

The first step consists of the calculus of the "area" A_j of each convex component $\mathcal{C}^j=P_0^jP_1^j\dots P_{n-1}^j$, given by

$$A_j=\frac{1}{2}\sum_{i=1}^{n-2}\left[P_0^jP_i^j,P_0^jP_{i+1}^j\right].$$

Then, the effective area used to compute the affine erosion is

$$\sigma_e = \max\left\{\frac{\sigma}{8}, \min_j A_j\right\}.$$

We restrict the erosion area to σ_e (which is less than σ in general) because the simplified algorithm for affine erosion (based on the breaking of the initial curve into convex components) may give a bad estimation of the continuous affine erosion + dilation when the area of one component is less than the erosion parameter. The term $\sigma/8$ is rather arbitrary and guarantees an upper bound to the number of iterations required to achieve the final scale.

Once σ_e is computed, the discrete erosion of each component is defined as the succession of each middle point of each segment $[AB]$ such that

1. A and B lie on the polygonal curve,
2. A or B is a vertex of the polygonal curve,
3. the area enclosed by $[AB]$ and the polygonal curve is equal to σ_e.

These points are easily computed by keeping in memory and updating the points A and B of the curve plus the associated chord area.

Notice that if the convex component is not closed (which is the case if the initial curve is not convex), then its endpoints are kept.

6.4. Iteration of the process. To iterate the process, we use the fact that if E_σ denotes the affine erosion plus dilation operator of area σ, and $h = (h_i)$ is a subdivision of the interval $[0, H]$ with $H = T/\omega$ and $\omega = \frac{1}{2}\left(\frac{3}{2}\right)^{2/3}$, then

$$E_{(h_1-h_0)^{3/2}} \circ E_{(h_2-h_1)^{3/2}} \circ \ldots \circ E_{(h_n-h_{n-1})^{3/2}} \quad \left(c_0\right) \quad \longrightarrow \ c_T$$

as $|h| = \max_i h_{i+1} - h_i \to 0$, where c_T is the affine shortening of c_0 described above by (3.2).

6.5. Comments. The algorithm takes a curve (closed or not) as input and produces an output curve representing the affine shortening of the input curve (it can be empty if the curve has disappeared). The parameters are the following:

- T, the scale to which the input curve must be smoothed.
- ε_r, the relative spatial precision at which the curve must be numerically represented (between 10^{-5} and 10^{-2} when using four-bytes C *float* numbers).
- n, the minimum number of iterations required to compute the affine shortening (it seems that $n \simeq 5$ is a good choice). From n, the erosion area σ used in step 3 is computed with the formula

$$\sigma^{2/3} = \frac{\alpha \cdot T^{4/3}}{n}.$$

Notice that thanks to the $\sigma/8$ lower bound for σ_e, the effective number of iterations cannot exceed $4n$.

- R, the radius of a disk containing the input curve, used to obtain homogeneous results when processing simultaneously several curves. The absolute precision ε used at step 2 is defined by $\varepsilon = R\varepsilon_r$.

The algorithm has linear complexity in time and memory, and its stability is ensured by the fact that each new curve is obtained as the set of the middle points of some particular chords of the initial curve, defined themselves by an integration process (an area computation). Hence, no derivation or curvature computation appears in the algorithm.

Acknowledgments. We thank Dr. Lenny Rudin for valuable conversations and suggestions for the image registration algorithms. We also thank the reviewers of this paper for their useful remarks.

REFERENCES

[1] S. Abbasi and F. Mokhtarian, *Retrieval of similar shapes under affine transformation*, in Proceedings of the International Conference on Visual Information Systems, Amsterdam, The Netherlands, 1999, pp. 566–574.

[2] L. Alvarez, F. Guichard, P. L. Lions, and J. M. Morel, *Axioms and fundamental equations of image processing*, Arch. Rational Mech. Anal., 123 (1993), pp. 200–257.

[3] L. Ambrosio, V. Caselles, S. Masnou, and J. M. Morel, *The connected components of sets of finite perimeter and applications to image processing*, J. Eur. Soc. Math. (JEMS), 3 (2001), pp. 213–266.

[4] H. Asada and M. Brady, *The curvature primal sketch*, IEEE Trans. PAMI, 8 (1986), pp. 2–14.

[5] F. Attneave, *Some informational aspects of visual perception*, Psychological Rev., 61 (1954), pp. 183–193.

[6] R. Bajcsy, R. Lieberson, and M. Reivich, *A computerized system for the elastic matching of deformed radiographic images to idealized atlas images*, J. Comput. Assisted Tomography, 7 (1983), pp. 618–625.

[7] G. Barles and C. Georgelin, *A simple proof of convergence for an approximation scheme for computing motions by mean curvature*, SIAM J. Numer. Anal., 32 (1995), pp. 484–500.

[8] S. Belongie, J. Malik, and J. Puzicha, *Matching shapes*, in Proceedings of the Eighth IEEE International Conference on Computer Vision, Vancouver, BC, Canada, 2001, pp. 454–461.

[9] L. G. Brown, *A survey of image registration techniques*, ACM Comput. Surveys, 24 (1992), pp. 325–376.

[10] V. Caselles, B. Coll, and J. M. Morel, *Topographic maps and local contrast changes in natural images*, Internat. J. Comput. Vision, 33 (1999), pp. 5–27.

[11] V. Caselles, B. Coll, and J. M. Morel, *Geometry and color in natural images*, J. Math. Imaging Vision, 16 (2002), pp. 89–105.

[12] T. Cohignac, C. Lopez, and J. M. Morel, *Integral and local affine invariant parameter and application to shape recognition*, in Proceedings of 12th International Conference on Pattern Recognition, Jerusalem, Israel, 1994, pp. 164–168.

[13] R. Deriche and G. Giraudon, *On corner and vertex detection*, in Proceedings of the IEEE Computer Society Computer Vision and Pattern Recognition Conference, Lahaina, Maui, HI, 1991, pp. 650–655.

[14] A. Desolneux, L. Moisan, and J. M. Morel, *Edge detection by Helmholtz principle*, J. Math. Imaging Vision, 14 (2001), pp. 271–284.

[15] F. Dibos, *Du groupe projectif au groupe des recalages, une nouvelle modélisation*, C. R. Acad. Sci. Paris Sér. I Math., 332 (2001), pp. 799–804.

[16] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.

[17] G. Dudek and J. K. Tsotsos, *Shape representation and recognition from multiscale curvature*, CVIU, 2 (1997), pp. 170–189.

[18] P. Dupuis, U. Grenander, and M. Miller, *Variational problems on flows of diffeomorphisms for image matching*, Quart. Appl. Math., 56 (1998), pp. 587–600.

[19] L. C. Evans, *Convergence of an algorithm for mean curvature motion*, Indiana Univ. Math. J., 42 (1993), pp. 553–557.

[20] O. Faugeras and R. Keriven, *Some recent results on the projective evolution of* 2*d curves*, in Proceedings of the IEEE ICIP, Vol. 3, Washington, D.C., 1995, pp. 13–16.

[21] O. Faugeras, Q. T. Luong, and T. Papadopoulo, *The Geometry of Multiple Images*, MIT Press, Cambridge, MA, 2001.

[22] W. E. L. Grimson, *Object Recognition by Computer: The Role of Geometric Constraints*, MIT Press, Cambridge, MA, 1990.

[23] F. Guichard and J. M. Morel, *Image Iterative Smoothing and P.D.E.'s*, manuscript.

[24] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, Cambridge, UK, 2000.

[25] R. K. Hu, *Visual pattern recognition by moments invariants*, IEEE Trans. Inform. Theory, 8 (1962), pp. 179–187.

[26] D. P. Huttenlocher and S. Ullman, *Recognizing solid objects by alignment with an image*, Internat. J. Comput. Vision, 5 (1990), pp. 195–212.

[27] G. Kanizsa, *Organization in vision: Essays on Gestalt Perception*, Praeger, New York, 1979.

[28] J. J. Koenderink and A. J. van Doorn, *Dynamic shape*, Biol. Cybernet., 53 (1986), pp. 383–396.

[29] A. Krzyzak, S. Y. Leung, and C. Y. Suen, *Reconstruction of two-dimensional patterns from Fourier descriptors*, MVA, 2 (1989), pp. 123–140.

[30] R. Kumar, S. Samarasekera, S. Hsu, and K. Hanna, *Registration of highly-oblique and zoomed in aerial video to reference imagery*, in Proceedings of the IEEE Computer Society Computer Vision and Pattern Recognition Conference, Barcelona, Spain, 2000.

[31] Y. Lamdan, J. Schwartz, and H. Wolfson, *Object recognition by affine invariant matching*, Proceedings of the IEEE Computer Society Computer Vision and Pattern Recognition Conference, Ann Arbor, MI, 1988, pp. 335–344.

[32] C. C. Lin and R. Chellappa, *Classification of partial 2-d shapes using Fourier descriptors*, in Proceedings of the IEEE Computer Society Computer Vision and Pattern Recognition Conference, 1986, pp. 344–350.

[33] J. L. Lisani, *Comparaison automatique d'images par leurs formes*, Ph.D. thesis, Université Paris-Dauphine, Paris, France, 2001.

[34] D. G. Lowe, *Three-dimensional object recognition from single two-dimensional images*, Artificial Intelligence, 31 (1987), pp. 355–395.

[35] G. Matheron, *Random Sets and Integral Geometry*, John Wiley, New York, 1975.

[36] J. Merriman, B. Bence, and S. Osher, *Diffusion generated motion by mean curvature*, in Computational Crystal Growers Workshop, AMS, Providence, RI, 1992, pp. 73–83.

[37] W. Metzger, *Gesetze des Sehens*, Waldemar Kramer, Frankfurt, Germany, 1975.

[38] L. Moisan, *Affine plane curve evolution: A fully consistent scheme*, IEEE Trans. Image Processing, 7 (1998), pp. 411–420.

[39] F. Mokhtarian and A. K. Mackworth, *A theory of multiscale, curvature-based shape representation for planar curves*, PAMI, 14 (1992), pp. 789–805.

[40] P. Monasse, *Contrast invariant image registration*, in Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Vol. 6, Phoenix, AZ, 1999, pp. 3221–3224.

[41] H. Murase and S. K. Nayar, *Learning and recognition of 3d objects from appearance*, in Proceedings of the IEEE Workshop on Qualitative Vision, New York, NY, 1993, pp. 39–50.

[42] E. Persoon and K. S. Fu, *Shape discrimination using Fourier descriptors*, IEEE Trans. Systems Man Cybernet., 7 (1977), pp. 170–179.

[43] T. H. Reiss, *Recognizing Planar Objects Using Invariant Image Features*, Lecture Notes in Comput. Sci. 676, Springer-Verlag, Berlin, 1993.

[44] C. A. Rothwell, *Object Recognition Through Invariant Indexing*, Oxford Science Publications, Oxford, UK, 1995.

[45] E. Rubin, *Visuell wahrgenommene Figuren*, Gyldendals, Copenhagen, 1921.

[46] G. Sapiro and A. Tannenbaum, *Affine invariant scale-space*, Internat. J. Comput. Vision, 11 (1993), pp. 25–44.

[47] J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, New York, 1982.

[48] C. H. Teh and C. R., *On image analysis by the method of moments*, IEEE Trans. PAMI, 10 (1988), pp. 496–513.

[49] M. Wertheimer, *Untersuchungen zur Lehre von der Gestalt ii*, Psychologische Forschung, 4 (1923), pp. 301–350. Translation published as *Laws of organization in perceptual forms,* in A Source Book of Gestalt Psychology, Routledge & Kegan Paul, London, 1938, pp. 71–88.

[50] H. J. Wolfson, *Model-based object recognition by geometric hashing*, in Proceedings of the First European Conference on Computer Vision, Antibes, France, 1990, pp. 526–536.

[51] L. Younes, *Computable elastic distances between shapes*, SIAM J. Appl. Math., 58 (1998), pp. 565–586.