

# The Use of Force Histograms for Affine-Invariant Relative Position Description

Pascal Matsakis, *Member, IEEE*, James M. Keller, *Fellow, IEEE*,  
Ozy Sjahputera, *Student Member, IEEE*, and Jonathon Marjamaa

**Abstract**—Affine invariant descriptors have been widely used for recognition of objects regardless of their position, size, and orientation in space. Examples of color, texture, and shape descriptors abound in the literature. However, many tasks in computer vision require looking not only at single objects or regions in images but also at their spatial relationships. In an earlier work, we showed that the relative position of two objects can be quantitatively described by a histogram of forces. Here, we study how affine transformations affect this descriptor. The position of an object with respect to another changes when the objects are affine transformed. We analyze the link between 1) the applied affinity, 2) the relative position before transformation (described through a force histogram), and 3) the relative position after transformation. We show that any two of these elements allow the third one to be recovered. Moreover, it is possible to determine whether (or how well) two relative positions are actually related through an affine transformation. If they are not, the affinity that best approximates the unknown transformation can be retrieved, and the quality of the approximation assessed.

**Index Terms**—Affine transformations, force histograms, spatial relations, descriptors, invariants, computer vision.

## 1 INTRODUCTION

COLOR, texture, and shape are fundamental concepts in computer vision, and many descriptors have been proposed to handle them. Typical examples are histogram-based color descriptors [1], texture descriptors based on co-occurrence matrices [2], and moment-based shape descriptors [3]. Evaluation and comparison criteria include extraction, storage and representation complexities, effectiveness in similarity retrieval, etc. Considerable attention has been paid to invariance under geometric transformations, i.e., to the design of robust descriptors, not (very) sensitive to the position of the camera with respect to the photographed scene [4]. As is logical, much research has been done considering only the class of similarity transformations (such as translations, rotations, and scalings), less has been done with the larger class of affine transformations (which also includes shears and stretches), and even less with the more general class of projective transformations. The fact is that the latter are often approximated by affinities, under the assumption of weak perspective [5]. For instance, affine invariant color descriptors have been used to handle changes in the irradiance pattern due to different lighting conditions and viewpoints [6], and affine invariant texture

descriptors to identify the same type of texture within an image (such as brick texture on two sides of a house) [7].

Numerous affine invariant shape descriptors can be found in the literature. Techniques include area moments [8], [9], curve moments [10], cross-weighted moments [11], Fourier descriptors [12], Legendre and Zernike descriptors [13], B-splines [14], Curvature Scale Space representation [15], etc. Most of the aforementioned descriptors rely, in a preprocessing step, on the recovery of the object boundary and the extraction of interest points (such as curvature extrema, bitangents, or inflection points). Typically, the boundary is encoded using a curvature, a centroidal distance, or a complex coordinate function. The function needs to be low-pass filtered and sampled. A set of descriptors is derived from it, and a normalization procedure is applied to remove the effects of affine transformations and eliminate dependency on the starting point in the parameterized boundary description. Although there have been a few attempts to handle objects with holes [16] or multiple connected components [17], the traditional underlying assumption is that the object can be defined by a single closed curve.

Relative position is another fundamental concept in computer vision. Many tasks require looking not only at single regions in images but also at their spatial relationships. Knowing the relative position of an object's components often helps in recognizing the object and, by the same token, its components. Similarly, knowing the relative position of objects in a scene often helps in understanding the scene and identifying the objects themselves. Moreover, different spatial layouts may yield different conclusions about the identity or function of the object or the scene. To some extent, relative position is to shape what texture is to color. Even though texture is an intuitive concept, a formal definition has proven elusive. The same applies to relative position. Authors usually describe the position of an object

- P. Matsakis is with the Department of Computing and Information Science, University of Guelph, Guelph, ON N1G 2W1, Canada. E-mail: matsakis@cis.uoguelph.ca.
- J.M. Keller and O. Sjahputera are with the Department of Computer Engineering and Computer Science, University of Missouri-Columbia, Columbia, MO 65211. E-mail: Kellerj@missouri.edu, osb1e@mizzou.edu.
- J. Marjamaa is with the Boeing Company, St Louis, MO. E-mail: jonathon.e.marjamaa@boeing.com.

Manuscript received 16 Jan. 2002; revised 19 Dec. 2002; accepted 25 June 2003.

Recommended for acceptance by S. Dickinson.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 115719.

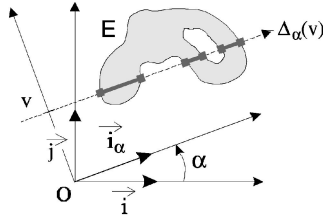


Fig. 1. Oriented straight lines and longitudinal sections. Here,  $E_{\alpha}(v) = E \cap \Delta_{\alpha}(v)$  is the union of three disjoint segments.

with respect to another in terms of a few spatial relationships, such as “to the right of,” “above,” “below,” etc. Moreover, objects are reduced to very elementary entities such as a point (centroid) or a (bounding) rectangle. These procedures are practical, notably for spatial reasoning (see, e.g., [18], [19], [20], [21]), but much information is lost. The histogram of angles [22], [23] was probably the first real relative position descriptor proposed in the literature. Matsakis then introduced the histogram of forces [24], which supersedes and generalizes the angle histogram [25]. The relative position of two objects is described by a periodic function with period  $2\pi$ . The function is sensitive to the shape of the objects, their orientation, their size, and the distance between them. Matsakis [26] reviews and classifies work on the use of force histograms. It touches topics as varied as linguistic scene description [27], classification of cranium orbits [28], human-robot communication [29], and spatial indexing mechanisms for medical image databases [30].

In this paper, we study how affine and “nearly” affine transformations affect relative positions described by force histograms. Our work constitutes a first step towards the design of affine invariant relative position descriptors. Object and affinity terminologies and notations are introduced in Section 2. The notion of the histogram of forces is presented in Section 3. The position of an object with respect to another changes when the two objects are affine transformed: in Section 4, we analyze the link between the applied affinity and the relative positions before and after transformation. Finally, in Section 5, we experimentally study the robustness of the theoretical tools presented in the previous sections to departures from the assumptions on the transformations being handled. Conclusions are given in Section 6.

## 2 WHEN PAIRS MATCH

In this paper, *affine transformations* are continually applied to *objects* in the Euclidean affine plane. We go over these two terms in Section 2.1. Some affine transformations will play a particularly important role among pairs of objects. They are examined in Section 2.2.

### 2.1 Terminology and Notations

As shown in Fig. 1, the plane reference frame is a positively oriented orthonormal frame  $(O, \vec{i}, \vec{j})$ . For any real numbers  $\alpha$  and  $v$ , the vectors  $\vec{i}_{\alpha}$  and  $\vec{j}_{\alpha}$  are the respective images of  $\vec{i}$  and  $\vec{j}$  through the  $\alpha$ -angle rotation, and  $\Delta_{\alpha}(v)$  is the oriented line whose reference frame is defined by  $\vec{i}_{\alpha}$  and the point of coordinates  $(0, v)$ —relative to  $(O, \vec{i}_{\alpha}, \vec{j}_{\alpha})$ . The

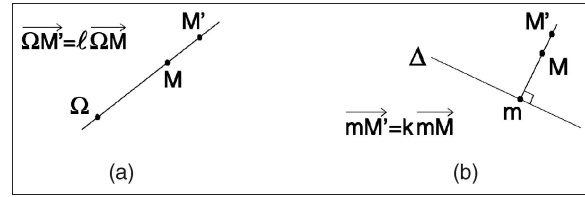


Fig. 2. Two basic affine transformations.  $M$  is transformed into  $M'$ . (a) Scaling:  $\Omega$  is the scaling center and  $\ell$ —a positive value—the scaling factor. (b) Stretch:  $\Delta$  is the axis and  $k$ —a nonzero value—the ratio.

term *object* denotes a nonempty bounded set of points,  $E$ , equal to its interior closure,<sup>1</sup> and such that, for any  $\alpha$  and  $v$ , the intersection  $E \cap \Delta_{\alpha}(v)$  is the union of a finite number of mutually disjoint segments. This intersection  $E \cap \Delta_{\alpha}(v)$ , denoted by  $E_{\alpha}(v)$ , is a *longitudinal section* of  $E$ . Note that an object may have holes in it and may consist of many connected components.

*Affine transformations*, also called *affinities*, are transformations that preserve collinearity (all points lying on a line initially still lie on a line after transformation) and ratios of distances (e.g., the midpoint of a line segment remains the midpoint after transformation). Translations, rotations, scalings, and stretches are basic affine transformations. As shown in Fig. 2, a stretch is characterized by an invariant line and a ratio. It scales in one direction only and does not maintain the objects’ proportions.

From now on,  $\mathcal{O}$  denotes a set of ordered pairs of objects such that for any  $(A, B)$  in  $\mathcal{O}$  and any affine transformation *aff*, the pairs  $(B, A)$  and  $(\text{aff}(A), \text{aff}(B))$  also belong to  $\mathcal{O}$ ; pairs such as  $(A, B)$ ,  $(A_0, B_0)$ ,  $(A', B')$ ,  $(A'_0, B'_0)$ , etc., denote elements of  $\mathcal{O}$ ; the symbols *aff*, *aff*<sub>0</sub>, *aff*<sub>1</sub>, etc., denote affine transformations; *tran*, *tran*<sub>0</sub>, *tran*<sub>1</sub>, etc., are translations; *rot* denotes a  $\rho$ -angle rotation, *rot*<sub>0</sub> a  $\rho_0$ -angle rotation, etc.; the values  $\rho$ ,  $\rho_0$ , etc., belong to the interval  $]-\pi, \pi]$ ; the symbols *sca*, *sca*<sub>0</sub>, etc., denote scalings; the scaling factors,  $\ell$ ,  $\ell_0$ , etc., are positive values; the symbols *stre*, *stre*<sub>0</sub>, *stre*<sub>1</sub>, etc., denote X-axis stretches; the ratio of *stre* is  $k$ , the ratio of *stre*<sub>0</sub> is  $k_0$ , etc.; all these values  $k$ ,  $k_0$ ,  $k'$ , etc., are also positive values.

### 2.2 A Particular Transformation Composition

In Section 4.3, we will focus on affine transformations that can be written in the following form:  $\text{stre}' \circ \text{tran} \circ \text{rot} \circ \text{sca} \circ \text{stre}$ . As illustrated by Fig. 3, the interest of this kind of composition lies in the interpretation that can be given to it. A set of objects before transformation (Fig. 3c) can be seen as a picture of a scene, the set of affine transformed objects (Fig. 3d) as a picture of the same scene taken from another view, and the sets of “partially” transformed objects (Figs. 3a, 3b, 3e, 3f) as different views from above. We will come back to this interpretation in Section 5.1. It is useful at this point to introduce the following definitions:

**Definition 1a.**  $(A'_0, B'_0)$  matches  $(A_0, B_0)$  through  $(\text{stre}', \text{tran}, \text{rot}, \text{sca}, \text{stre})$ —or *through*  $\text{stre}' \circ \text{tran} \circ \text{rot} \circ \text{sca} \circ \text{stre}$ —if and only if:

$$A'_0 = \text{stre}'(\text{tran}(\text{rot}(\text{sca}(\text{stre}(A_0))))))$$

$$\text{and } B'_0 = \text{stre}'(\text{tran}(\text{rot}(\text{sca}(\text{stre}(B_0))))).$$

1. In other words, it is a 2D object that does not include any “grafting,” such as an arc or isolated point.

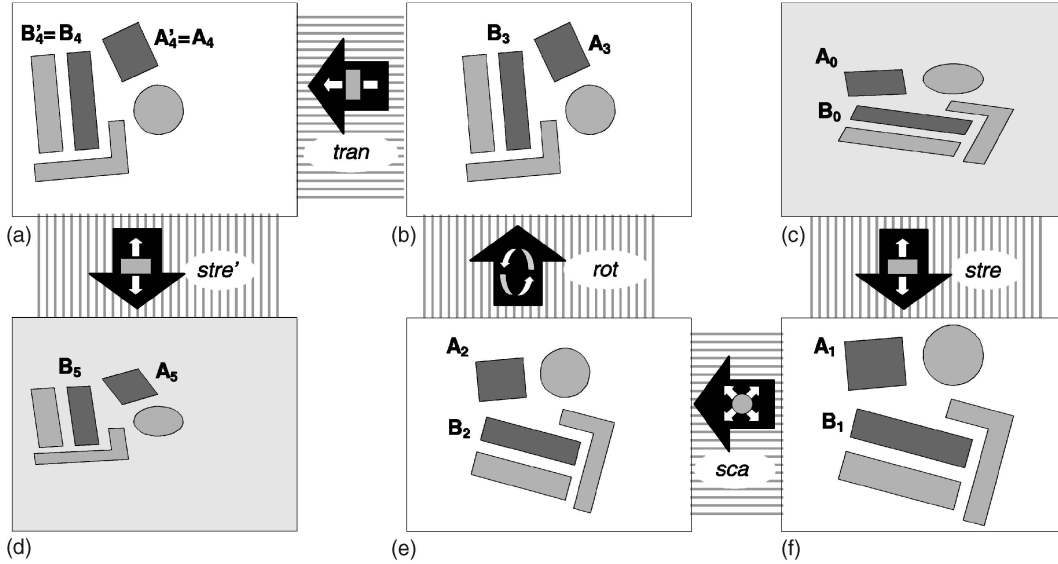


Fig. 3. A particular transformation composition. Here:  $A_5 = \text{stre}'(\text{tran}(\text{rot}(\text{sca}(\text{stre}(A_0))))$  and  $B_5 = \text{stre}'(\text{tran}(\text{rot}(\text{sca}(\text{stre}(B_0))))$ . We say that the pairs  $(A_0, B_0)$  and  $(A_5, B_5)$  match. Note that (c) can be seen as a picture of a scene, (d) as a picture of the same scene from another bird's-eye view, and (a), (b), (e), and (f) as different views from above.

**Definition 1b.**  $(A_0, B_0)$  and  $(A'_0, B'_0)$  match if and only if there exists a 5-tuple  $(\text{stre}', \text{tran}, \text{rot}, \text{sca}, \text{stre})$  such that  $(A'_0, B'_0)$  matches  $(A_0, B_0)$  through  $(\text{stre}', \text{tran}, \text{rot}, \text{sca}, \text{stre})$ .

**Definition 1c.**  $\{A_0, B_0\}$  and  $\{A'_0, B'_0\}$  match if and only if  $(A_0, B_0)$  and  $(A'_0, B'_0)$ , or  $(B'_0, A'_0)$ , match.

In Fig. 3, for instance,  $(A_0, B_0)$  and  $(A_5, B_5)$  match. One may wonder if the 5-tuple of geometric transformations in Definition 1a is unique. Generally speaking, the transformations are not, but the parameters  $\rho$  (rotation angle),  $\ell$  (scaling factor),  $k$ , and  $k'$  (stretch ratios) are. However, there are exceptions to the rule. For instance, the uniqueness of  $\rho$ ,  $\ell$ ,  $k$ , and  $k'$  is obviously not guaranteed if the set  $\mathcal{O}$  of all considered object pairs is *ambiguous*, i.e., if some pairs do not allow three nonaligned anchor points to be determined (Fig. 4).

**Definition 2.** The set  $\mathcal{O}$  is ambiguous if and only if there exists a pair  $(A, B)$  in  $\mathcal{O}$ , and a rotation  $\text{rot}$  different than the identity transformation (0-angle rotation), such that:  $\text{rot}(A) = A$  and  $\text{rot}(B) = B$ .

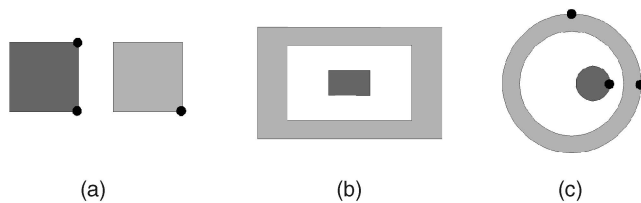


Fig. 4. Ambiguous configurations. Look at these pairs of objects from any bird's-eye view. In case (a), you would be able to retrieve each of the three points even if they were not marked. The same applies to case (c). In case (b), however, the center is the only point that you could retrieve without ambiguity, because a  $\pi$ -angle rotation about it leaves the objects unchanged. The considered pair would make  $\mathcal{O}$  ambiguous.

Proposition 1 is proven in Appendix A. The third case is illustrated by Fig. 5.

**Proposition 1.** Assume  $\mathcal{O}$  is not ambiguous, and assume  $(A'_0, B'_0)$  matches  $(A_0, B_0)$  through  $(\text{stre}'_1, \text{tran}_1, \text{rot}_1, \text{sca}_1, \text{stre}_1)$  and through  $(\text{stre}'_2, \text{tran}_2, \text{rot}_2, \text{sca}_2, \text{stre}_2)$ .

1. If  $\rho_1 \notin \{-\pi/2, 0, \pi/2, \pi\}$ , then  $\rho_1 = \rho_2$ ,  $\ell_1 = \ell_2$ ,  $k_1 = k_2$ , and  $k'_1 = k'_2$ .
2. If  $\rho_1 \in \{0, \pi\}$ , then  $\rho_1 = \rho_2$ ,  $\ell_1 = \ell_2$ , and  $k_1 k'_1 = k_2 k'_2$ .
3. If  $\rho_1 \in \{-\pi/2, \pi/2\}$ , then  $\rho_1 = \rho_2$ ,  $\ell_1 k_1 = \ell_2 k_2$ , and  $k_1/k'_1 = k_2/k'_2$ .

### 3 WHEN OBJECTS ATTRACT

In an earlier work [24], [25], we showed that the position of an object with respect to another one can be described by a histogram of forces. Here, we examine briefly this notion and discuss data models and related complexity issues. We will then be able, in Section 4, to study how affine transformations affect relative positions.

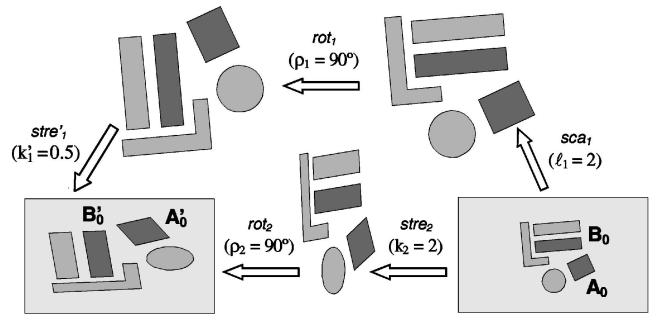


Fig. 5. Nonuniqueness of  $(\rho, \ell, k, k')$ . In general, if  $(A'_0, B'_0)$  matches  $(A_0, B_0)$  through  $(\text{stre}'_1, \text{tran}_1, \text{rot}_1, \text{sca}_1, \text{stre}_1)$ , then the 4-tuple  $(\rho, \ell, k, k')$  is unique. However, as stated by Proposition 1, there are exceptions to the rule. This counter-example exhibits two 4-tuples:  $(\rho_1, \ell_1, k_1, k'_1) = (90^\circ, 2, 1, 0.5)$  and  $(\rho_2, \ell_2, k_2, k'_2) = (90^\circ, 1, 2, 1)$ .

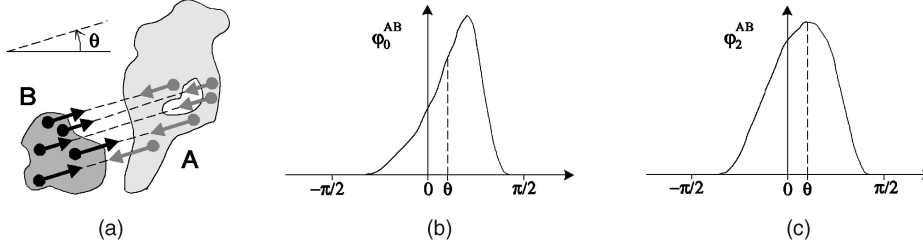


Fig. 6. Force histograms. Physical interpretation and examples. (a)  $\varphi^{AB}(\theta)$  is the scalar resultant of elementary forces (black arrows). Each one tends to move B in direction  $\theta$ . (b) The histogram of constant forces associated with  $(A, B)$  is one possible representation of the position of A relative to B. (c) The histogram of gravitational forces associated with  $(A, B)$  is another possible representation.

### 3.1 The Notion of the Histogram of Forces

Consider two objects A and B, as in Fig. 6a. The position of A with regard to B is represented by a function  $\varphi^{AB}$  from  $\mathbb{R}$  (the set of real numbers) into  $\mathbb{R}_+$  (the set of nonnegative real numbers). For any direction  $\theta$ , the value  $\varphi^{AB}(\theta)$  can be seen as the scalar resultant of elementary forces. These forces are exerted by the points of A on those of B, and each tends to move B in direction  $\theta$ . If the domain of  $\varphi^{AB}$  is all of  $\mathbb{R}$ , then the pair  $(A, B)$  is termed  $\varphi$ -*assessable* and  $\varphi^{AB}$  is called the *histogram of forces associated with  $(A, B)$  via  $\varphi$* , or the  $\varphi$ -*histogram associated with  $(A, B)$* . The object A is the *argument*, and B is the *referent*.

Actually,  $\varphi$  denotes a mapping from  $\mathbb{R}$  into  $\mathbb{R}_+$  and defines the force fields. Two other functions, F and f, can be introduced to conveniently describe the mathematical link between  $\varphi$ , A, B,  $\theta$ , and  $\varphi^{AB}(\theta)$  (Fig. 7). Let T be the set of triples  $(\alpha, E_\alpha(v), G_\alpha(v))$ , where  $\alpha$  and  $v$  are any real numbers and E and G are any objects. Remember that  $E_\alpha(v) = E \cap \Delta_\alpha(v)$  and  $G_\alpha(v) = G \cap \Delta_\alpha(v)$  (Fig. 1). The function F is from T into  $\mathbb{R}_+$ , and we have (Fig. 7d):

$$\varphi^{AB}(\theta) = \int_{-\infty}^{+\infty} F(\theta, A_\theta(v), B_\theta(v)) dv. \quad (1)$$

$F(\theta, A_\theta(v), B_\theta(v))$  corresponds to the resultant of forces exerted by points of  $A_\theta(v)$  on points of  $B_\theta(v)$ . Simply put, F is in charge of the longitudinal sections (Fig. 7c). It delegates

the handling of segments to f, which delegates in turn the handling of points to  $\varphi$ . The function f is from  $\mathbb{R}_+ \times \mathbb{R} \times \mathbb{R}_+$  into  $\mathbb{R}_+$ . It is defined by:

$$f(x, y, z) = \int_{y+z}^{x+y+z} \left( \int_0^z \varphi(u-w) dw \right) du. \quad (2)$$

In (2), the symbols x and z denote the lengths of two aligned segments and y indicates the relative position of these segments (Fig. 7b). The symbols u and w denote the coordinates of two points on an oriented line (Fig. 7a), while  $u-w$  indicates the relative position of these points on the line. Note that the mapping  $\varphi$  is zero on  $\mathbb{R}_-$ : An elementary force that tends to move B in direction  $\theta + \pi$  will be taken into account when computing  $\varphi^{AB}(\theta + \pi)$ , not when computing  $\varphi^{AB}(\theta)$ . As an example, gravitational force fields can be represented by the mapping  $\varphi_2$  such that:  $\forall d \in \mathbb{R}_-, \varphi_2(d) = 0$  and  $\forall d \in \mathbb{R}_+, \varphi_2(d) = 1/d^2$  (where  $\mathbb{R}_+$  denotes the set of positive real numbers). This is according to Newton's law of gravity, which states that every particle attracts every other particle with a force inversely proportional to the square of the distance between them. The objects A and B can then be seen as two flat metal plates of uniform density and constant and negligible thickness—a kind of objects commonly considered in physics [31]. We are not bound, however, to physical laws. The choice of  $\varphi$  only depends on the properties we want the force

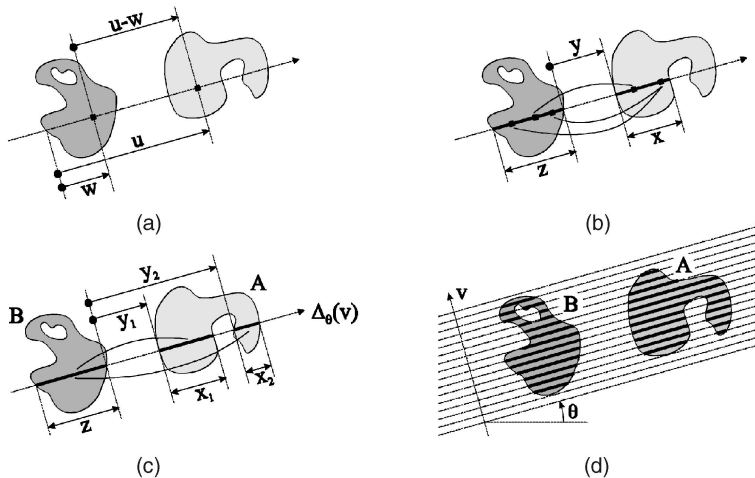


Fig. 7. Force histograms. Mathematical link between  $\varphi$ , A, B,  $\theta$ , and  $\varphi^{AB}(\theta)$ . (a) Handling of points:  $\varphi(u-w)$ . (b) Handling of segments:  $f(x, y, z) = \int_{y+z}^{x+y+z} \left( \int_0^z \varphi(u-w) dw \right) du$ . (c) Handling of longitudinal sections:  $F(\theta, A \cap \Delta_\theta(v), B \cap \Delta_\theta(v)) = f(x_1, y_1, z) + f(x_2, y_2, z)$ . (d) Handling of directions:  $\varphi^{AB}(\theta) = \int_{-\infty}^{+\infty} F(\theta, A \cap \Delta_\theta(v), B \cap \Delta_\theta(v)) dv$ .

histograms to have. For instance,  $\varphi$  can be any of the  $\varphi_r$  functions defined from  $\mathbb{R}$  into  $\mathbb{R}_+$  by:

$$\forall d \in \mathbb{R}_-, \varphi_r(d) = 0 \text{ and } \forall d \in \mathbb{R}_+, \varphi_r(d) = 1/d^r. \quad (3)$$

In the rest of the paper, only  $\varphi_r$ -histograms are considered. As shown in Section 4.1, these histograms have nice geometric properties. For any  $r$ , any pair of disjoint objects is  $\varphi_r$ -assessable. Moreover, if  $r$  is lower than 1, any pair of intersecting objects is  $\varphi_r$ -assessable too. As a matter of fact,  $\varphi_r$ -histograms can also handle unbounded objects (if  $r$  is greater than 1), and fuzzy (i.e., gray-level) objects [24], [25]. In theory, they can handle neither 0D objects, nor 1D objects. In practice, this is usually not a limitation since points and lines can easily be treated as 2D objects. The  $\varphi_0$ -histogram (histogram of constant forces) and  $\varphi_2$ -histogram (histogram of gravitational forces) have very different and interesting characteristics. The former provides a global view of the relative position between the objects. It considers the closest parts and the farthest parts of the objects equally, whereas the  $\varphi_2$ -histogram focuses on the closest parts. Examples are presented in Fig. 6.

### 3.2 Data Models and Complexity Issues

Not only the constraints on the objects are few, but the histogram of forces also allows data to be stored and efficiently processed in raster as well as in vector form [24], [25], [29]. In the case of raster data, the computation of a histogram value,  $\varphi_r^{AB}(\theta)$ , is achieved by partitioning the objects  $A$  and  $B$  into segments (more precisely, into sets of adjacent pixels). The generation of these segments is based on the rasterization of a pencil of parallel lines (see Fig. 7d) by means of Bresenham's algorithm in integer arithmetic (which is commonly circuit coded in visualization systems). The handling of the pair  $(A, B)$  then comes down to the handling of segment pairs. Computation of forces between two aligned segments translates into the instantiation of variables in a predetermined, hard-coded symbolic expression. This expression does not depend on  $A$  nor  $B$ . It is obtained beforehand through symbolic computation of a double integral ((2), Fig. 7b). The computation of  $\varphi_r^{AB}(\theta)$  is of complexity  $O(n\sqrt{n})$ , where  $n$  denotes the number of pixels of the processed image. The complexity drops to  $O(n)$  for convex objects. In practice, of course, only a finite set of directions  $\theta$  is considered. An angle increment of 2 to 3 degrees is usually appropriate [25]. Note that all pairs of image objects can be processed simultaneously. Moreover, force histogram computation is highly parallelizable.

In the case of vector data, each object is represented by a set of polygonal contours—the number of which depends on the number of holes and connected components. Each polygon, in turn, is represented by a list of vertex coordinates. The objects are partitioned into trapezoids by drawing parallel lines through both  $A$  and  $B$  vertices, following direction  $\theta$ . The handling of  $(A, B)$  comes down to the handling of trapezoid pairs. Computation of forces between two trapezoids with aligned bases translates into the instantiation of variables in a predetermined, hard-coded symbolic expression. This time, the expression is obtained through symbolic computation of a triple integral.

The computation of  $\varphi_r^{AB}(\theta)$  is of complexity  $O(n \log(n))$ , where  $n$  denotes the total number of object vertices.

The algorithms above are particularly fast. In [29], for instance, a robot equipped with sonar sensors describes, in natural language, its position with respect to the sensed environment objects. The generation of linguistic descriptions is based on the computation of force histograms. The program runs at real-time speed. Processing of all detected obstacles is done faster than the robot can move.

## 4 FORCE HISTOGRAMS REACT WELL TO AFFINE TRANSFORMATIONS

The position of an object with respect to another one changes when the two objects are affine transformed. In this section, we study the link between the applied affinity and the relative positions before and after transformation. For instance, how will the position of  $A$  with respect to  $B$  be affected by a  $\pi/4$ -angle rotation? Or, knowing the relative positions before and after transformation, is it possible to retrieve that transformation? Relative positions will, of course, be described by  $\varphi_r$ -histograms. Let us formalize the aim of the study. We are interested in the following expression:  $\varphi_r^{A'B'} = \varphi_r^{\text{aff}(A)\text{aff}(B)}$ . First, we would like to be able to retrieve  $\varphi_r^{\text{aff}(A)\text{aff}(B)}$ , knowing  $\varphi_r^{AB}$  and  $\text{aff}$ . This is covered in Section 4.1. Then, we would like to solve the three equations below: (4) for the unknown variables  $A'$  and  $B'$ , (5) for the unknown variables  $A$  and  $B$ , and (6) for the unknown variable  $\text{aff}$ . As will be seen, these equations are related to important issues. Equations (4) and (5) are dealt with in Section 4.2, whereas (6) is examined in Section 4.3.

$$\varphi_r^{A'B'} = \varphi_r^{\text{aff}_0(A_0)\text{aff}_0(B_0)}, \quad (4)$$

$$\varphi_r^{A'_0B'_0} = \varphi_r^{\text{aff}_0(A)\text{aff}_0(B)}, \quad (5)$$

$$\varphi_r^{A'_0B'_0} = \varphi_r^{\text{aff}(A_0)\text{aff}(B_0)}. \quad (6)$$

### 4.1 Fundamental Properties

Assume  $(A, B)$  is  $\varphi_r$ -assessable. We state here that  $(B, A)$  and  $(\text{aff}(A), \text{aff}(B))$  are  $\varphi_r$ -assessable too. Moreover,  $\varphi_r^{BA}$  can easily be deduced from  $\varphi_r^{AB}$ , even if the objects  $A$  and  $B$  themselves are not known. It is also possible to deduce  $\varphi_r^{\text{aff}(A)\text{aff}(B)}$  from  $\text{aff}$  and  $\varphi_r^{AB}$  without, again, knowing either  $A$  or  $B$ . In other words, the relative position of two affine transformed objects can be deduced from the considered affinity and the relative position—known through a  $\varphi_r$ -histogram—of the objects before transformation. There are two basic reasons for that: First, any affine transformation preserves collinearity and the computation of force histogram values is precisely based on the partitioning of objects into longitudinal sections (Fig. 7); then, any affine transformation preserves ratios of distances and the scaling of longitudinal sections conveniently results in the multiplication of resultant forces by a constant factor (Property 8, Appendix B). The following properties illustrate the statements above. Notations are as introduced in Section 2.1. Properties 1 and 4 are depicted by Fig. 8. Proofs of Properties 1 to 4 are in [24, chapter 2, Appendix A].

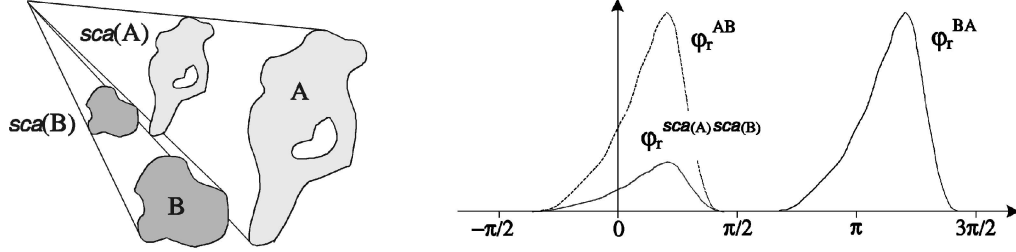


Fig. 8. Properties 1 and 4. Knowing  $\varphi_r^{AB}$ , it is easy to retrieve  $\varphi_r^{BA}$  and  $\varphi_r^{sca(A)sca(B)}$ .

Property 5 is new and its proof is given in Appendix C. All proofs naturally rely on (1), (2), and (3).

Assume  $(A, B)$  is  $\varphi_r$ -assessable:

**Property 1.** The pair  $(B, A)$  is also  $\varphi_r$ -assessable and:  $\forall \theta \in \mathbb{R}$ ,  
 $\varphi_r^{BA}(\theta) = \varphi_r^{AB}(\theta + \pi)$ .

**Property 2.**  $(\text{tran}(A), \text{tran}(B))$  is  $\varphi_r$ -assessable and:  $\forall \theta \in \mathbb{R}$ ,  
 $\varphi_r^{\text{tran}(A)\text{tran}(B)}(\theta) = \varphi_r^{AB}(\theta)$ .

**Property 3.**  $(\text{rot}(A), \text{rot}(B))$  is  $\varphi_r$ -assessable and:  $\forall \theta \in \mathbb{R}$ ,  
 $\varphi_r^{\text{rot}(A)\text{rot}(B)}(\theta) = \varphi_r^{AB}(\varphi - \rho)$ .

**Property 4.**  $(\text{sca}(A), \text{sca}(B))$  is  $\varphi_r$ -assessable and:  $\forall \theta \in \mathbb{R}$ ,  
 $\varphi_r^{\text{sca}(A)\text{sca}(B)}(\theta) = \ell^{3-r} \varphi_r^{AB}(\theta)$ .

**Property 5.** For any  $\alpha$  and positive value  $x$ , let  $\alpha_{[x]}$  denote the value  $\text{atan}(x^{-1}\tan\alpha)$  if  $\cos\alpha$  is positive, the value  $\alpha$  if  $\cos\alpha$  is zero, and the value  $\text{atan}(x^{-1}\tan\alpha) + \pi$  otherwise. The pair  $(\text{stre}(A), \text{stre}(B))$  is  $\varphi_r$ -assessable and:

$\forall \theta \in \mathbb{R}$ ,

$$\varphi_r^{\text{stre}(A)\text{stre}(B)}(\theta) = k^{2-r} [1 + (k^2 - 1)\cos^2\theta]^{(r-1)/2} \varphi_r^{AB}(\theta_{[k]}).$$

Note that Properties 2 to 4 show a way to achieve invariance under translations, rotations and scalings (i.e., similarity transformations). Let  $m$  and  $c$  be the mean and centroid<sup>2</sup> of  $\varphi_r^{AB}$ . The value  $m$  is a force on the Y-axis of the histogram, whereas  $c$  is an angle on the X-axis. Now, let  $\overline{\varphi_r^{AB}}$  be the normalized histogram defined by:  $\forall \theta \in \mathbb{R}$ ,  $\overline{\varphi_r^{AB}}(\theta) = m^{-1} \varphi_r^{AB}(\theta + c)$ . The mean and centroid of  $\overline{\varphi_r^{AB}}$  do not depend on  $A$  nor on  $B$  (they are 1 and 0). Therefore,

$$\overline{\varphi_r^{\text{tran}(A)\text{tran}(B)}} = \overline{\varphi_r^{\text{rot}(A)\text{rot}(B)}} = \overline{\varphi_r^{\text{sca}(A)\text{sca}(B)}} = \overline{\varphi_r^{AB}}.$$

Invariance under stretches seems much harder to achieve. Property 5 can, however, be exploited, as will be seen in Section 4.3.

## 4.2 About Completeness

We are interested here in (4) and (5). Assume all elements in the set  $\mathcal{O}$  of all considered object pairs are  $\varphi_r$ -assessable. Obviously, the presence of  $\text{aff}_0$  in (4) does not bring much:  $\text{aff}_0(A_0)$  and  $\text{aff}_0(B_0)$  could be renamed  $A_1$  and  $B_1$ . It does not bring much in (5) either: We could solve (5) for the unknown variables  $A' = \text{aff}_0(A)$  and  $B' = \text{aff}_0(B)$ ,

and then retrieve  $A$  and  $B$  using the inverse affine transformation  $\text{aff}_0^{-1}$ . Solving (4) and (5) therefore comes down to solving (7).

$$\varphi_r^{A'B'} = \varphi_r^{A_0B_0}. \quad (7)$$

Basically, (7) brings out the fundamental issue of completeness. May a given histogram be associated with more than one pair of objects? If so, how are these pairs related? What is the discriminative power of the histogram of forces? Let  $\mathcal{O}^{A_0B_0}$  be the set of solutions of (7), i.e., the set of pairs  $(A', B')$  such that  $\varphi_r^{A'B'} = \varphi_r^{A_0B_0}$ . Obviously, we can expect  $(A_0, B_0)$  not to be the only element of  $\mathcal{O}^{A_0B_0}$  because a  $\varphi_r$ -histogram represents the relative position of two objects and different objects may be exactly in the same relative position.<sup>3</sup> Consider any translation  $\text{tran}_1$ , any  $\pi$ -angle rotation  $\text{rot}_1$ , and any scaling  $\text{sca}_1$ . According to Properties 1 to 4, the pairs  $(\text{tran}_1(A_0), \text{tran}_1(B_0))$  and  $(\text{tran}_1(\text{rot}_1(B_0)), \text{tran}_1(\text{rot}_1(A_0)))$  also belong to  $\mathcal{O}^{A_0B_0}$ ; moreover, if  $r$  is equal to 3, the pairs  $(\text{tran}_1(\text{sca}_1(A_0)), \text{tran}_1(\text{sca}_1(B_0)))$  and  $(\text{tran}_1(\text{rot}_1(\text{sca}_1(B_0))), \text{tran}_1(\text{rot}_1(\text{sca}_1(A_0))))$  belong to  $\mathcal{O}^{A_0B_0}$  too. One may wonder if  $\mathcal{O}^{A_0B_0}$  contains other elements than the previous ones. As suggested by Fig. 9b, it might well be. Determining these other elements is an intricate problem that has not been solved yet.<sup>4</sup> In practical situations, however, it is most reasonable to assume that if  $(A', B')$  is such that  $\varphi_r^{A'B'} = \varphi_r^{A_0B_0}$ , then  $(A', B')$  is necessarily one of the pairs listed above. In other words, it is reasonable to assume that the set  $\mathcal{O}$  of all considered object pairs is  $r$ -regular.

**Definition 3.** The set  $\mathcal{O}$  is  $r$ -regular if and only if its elements are  $\varphi_r$ -assessable and, for any pairs  $(A, B)$  and  $(A', B')$ , the following is true: If  $\varphi_r^{A'B'} = \varphi_r^{AB}$ , then there exist a translation  $\text{tran}_1$ , a  $\pi$ -angle rotation  $\text{rot}_1$ , and a scaling  $\text{sca}_1$  such that

$$(A', B') = (\text{tran}_1(A), \text{tran}_1(B))$$

$$\text{or } (A', B') = (\text{tran}_1(\text{rot}_1(B)), \text{tran}_1(\text{rot}_1(A)))$$

$$\text{or } r = 3 \text{ and } (A', B') = (\text{tran}_1(\text{sca}_1(A)), \text{tran}_1(\text{sca}_1(B)))$$

$$\text{or } r = 3 \text{ and}$$

$$(A', B') = (\text{tran}_1(\text{rot}_1(\text{sca}_1(B))), \text{tran}_1(\text{rot}_1(\text{sca}_1(A)))).$$

3. In fact,  $\mathcal{O}^{A_0B_0}$  itself can be seen as a relative position. Consider the relation  $\sim$  defined by:  $(A', B') \sim (A, B)$  iff  $\varphi_r^{A'B'} = \varphi_r^{AB}$ . It is an equivalence relation between elements of  $\mathcal{O}$  (the set of all considered object pairs). A relative position can therefore be defined as any equivalence class of  $\sim$ . Under this definition,  $\mathcal{O}^{A_0B_0}$ , i.e., the equivalence class of  $(A_0, B_0)$ , is a relative position.

4. We refer here to the *inverse problem*: Given a force histogram, construct all pairs of objects this histogram is associated with.

2. Obviously, the "centroid" of  $\varphi_r^{AB}$  has to be computed over an appropriate  $2\pi$ -length interval (since  $\varphi_r^{AB}$  is a periodic function with period  $2\pi$ ). In some cases, selecting such an interval might not be easy. Imagine, for instance, that  $A$  surrounds  $B$  and all histogram values are nonzero values.

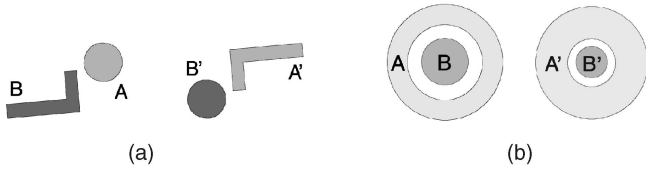


Fig. 9. Force histograms and completeness issue. (a)  $\varphi_r^{A'B'} = \varphi_r^{AB}$  because  $(A', B')$  is the image of  $(B, A)$  through a  $\pi$ -angle rotation. (b) By decreasing the diameter of  $B$  and the inner diameter of  $A$ , it is possible to find a pair  $(A', B')$  such that  $\varphi_r^{A'B'} = \varphi_r^{AB}$ . Note that in (a) as in (b), there is no affine transformation that maps  $(A, B)$  into  $(A', B')$ .

### 4.3 Matching Pairs and Force Histograms

Given two pairs  $(A_0, B_0)$  and  $(A'_0, B'_0)$ , we would like now to solve (6), i.e.,  $\varphi_r^{A'_0, B'_0} = \varphi_r^{\text{aff}(A_0)\text{aff}(B_0)}$ , for the unknown variable  $\text{aff}$ . To simplify the problem in hand, however, we reformulate it as follows: Given two pairs  $(A_0, B_0)$  and  $(A'_0, B'_0)$ , solve (6) for the unknown variable  $(\text{stre}', \text{tran}, \text{rot}, \text{sca}, \text{stre})$ .

$$\varphi_r^{A'_0, B'_0} = \varphi_r^{\text{stre}'(\text{tran}(\text{rot}(\text{sca}(\text{stre}(A_0))))\text{stre}'(\text{tran}(\text{rot}(\text{sca}(\text{stre}(B_0))))))}. \quad (8)$$

Roughly, we show here that: 1) Equation (8) has solutions if and only if the object pairs  $(A_0, B_0)$  and  $(A'_0, B'_0)$  match (see Definition 1, Section 2.2), 2) any solution of (8) allows the affinity the two pairs match through to be recovered, and 3)  $(A_0, B_0)$  and  $(A'_0, B'_0)$  need to be known only through the histograms  $\varphi_r^{A_0 B_0}$  and  $\varphi_r^{A'_0 B'_0}$ . In other words, given two force histograms, it is possible to check whether the pairs of objects they are associated with match and, if so, recover the affine transformation. Points 1) and 2) are handled by Proposition 2 below, and point 3) by Proposition 3. The proofs are given in Appendices D and E.

**Proposition 2.** Assume the set  $\mathcal{O}$  of all considered object pairs is  $r$ -regular (Definition 3). If  $(A'_0, B'_0)$  matches  $(A_0, B_0)$  through  $(\text{stre}', \text{tran}, \text{rot}, \text{sca}, \text{stre})$ , then  $(\text{stre}', \text{tran}, \text{rot}, \text{sca}, \text{stre})$  is a solution of (8). Moreover, if  $(\text{stre}', \text{tran}, \text{rot}, \text{sca}, \text{stre})$  is a solution of (8), then there exist a translation  $\text{tran}_2$ , a rotation  $\text{rot}_2$  and a scaling  $\text{sca}_2$  such that  $(A'_0, B'_0)$  matches either  $(A_0, B_0)$  or  $(B_0, A_0)$  through  $(\text{stre}', \text{tran}_2, \text{rot}_2, \text{sca}_2, \text{stre})$ . In the first case,  $\text{rot}_2$  is a  $\rho$ -angle rotation (like  $\text{rot}$ ). In the second case,  $\text{rot}_2$  is a  $(\pi + \rho)$ -angle rotation. In both cases,  $\text{sca}_2$  can be chosen equal to  $\text{sca}$  if  $r$  is different than 3.

**Proposition 3.**  $(\text{stre}', \text{tran}, \text{rot}, \text{sca}, \text{stre})$  is a solution to (8) if and only if the 4-tuple  $(\rho, \ell, k, k')$  of associated parameters is a solution to (9):

$$\begin{aligned} \forall \theta \in \mathbb{R}, \quad & k'^{r-2} [1 + (k'^{-2} - 1) \cos^2 \theta]^{(r-1)/2} \varphi_r^{A'_0 B'_0}(\theta_{\lfloor k'^{-1} \rfloor}) \\ & = \ell^{3-r} k^{2-r} [1 + (k^2 - 1) \cos^2(\theta - \rho)]^{(r-1)/2} \varphi_r^{A_0 B_0}((\theta - \rho)_{\lfloor k \rfloor}). \end{aligned} \quad (9)$$

Remember that  $\alpha_{\lfloor x \rfloor}$  denotes the value  $\text{atan}(x^{-1} \tan \alpha)$  if  $\cos \alpha$  is positive, the value  $\alpha$  if  $\cos \alpha$  is zero, and the value  $\text{atan}(x^{-1} \tan \alpha) + \pi$  otherwise. Equation (9) is much nicer to work with than (8) because it is a numerical equation, and various methods can be used to solve it. We will come back to this issue in Section 4.4. Note that (9) is found by computing the same histogram in two different ways and

then equating the obtained expressions (Appendix E). These expressions are shown below.  $(A_4, B_4)$  and  $(A'_4, B'_4)$  actually refer to the same pair of objects, as in Fig. 3a.

$$\begin{aligned} \forall \theta \in \mathbb{R}, \\ \varphi_r^{A'_4 B'_4}(\theta) & = k'^{r-2} [1 + (k'^{-2} - 1) \cos^2 \theta]^{(r-1)/2} \varphi_r^{A_0 B_0}(\theta_{\lfloor k'^{-1} \rfloor}). \end{aligned} \quad (10)$$

$$\begin{aligned} \forall \theta \in \mathbb{R}, \\ \varphi_r^{A_4 B_4}(\theta) & = \ell^{3-r} k^{2-r} [1 + (k^2 - 1) \cos^2(\theta - \rho)]^{(r-1)/2} \varphi_r^{A_0 B_0}((\theta - \rho)_{\lfloor k \rfloor}). \end{aligned} \quad (11)$$

### 4.4 The Matching Algorithm

Assume  $\mathcal{O}$ , the set of all considered pairs of objects, is not ambiguous and  $r$ -regular (Definition 2 and Definition 3). Then, according to Propositions 1 and 2, knowing *one* solution of (9) allows us to retrieve *all* solutions of (9). The simplest way to solve this equation is to implement the algorithm below, which we will later refer to as the *matching algorithm*:

```

0. MATCH ← false;
1. Compute  $\varphi_r^{A'_0 B'_0}$ ;
2. Compute  $\varphi_r^{A_0 B_0}$ ;
3. For each  $k'$ 
   3.1. Compute  $\varphi_r^{A'_4 B'_4}$  as in (10);
   3.2. For each  $k$ 
     3.2.1. For each  $\ell$ 
       3.2.1.1. For each  $\rho$ 
         3.2.1.1.1. Compute  $\varphi_r^{A_4 B_4}$  as in (11);
         3.2.1.1.2. If  $\varphi_r^{A'_4 B'_4} = \varphi_r^{A_4 B_4}$ 
           then  $R \leftarrow \rho, L \leftarrow \ell, K \leftarrow k, K' \leftarrow k'$  and
           MATCH ← true;

```

At the end of the run, the Boolean MATCH is *true* if a solution has been found and stored in the floating-point variables  $R, L, K,$  and  $K'$ , and *false* otherwise. Let us go over and briefly comment on the different steps.

#### 4.4.1 Histogram Computation (Steps 1, 2, 3.1, and 3.2.1.1.1)

In practice, of course, only a finite set of evenly distributed directions  $\theta$  is considered. The calculation of each  $\varphi_r^{A'_0 B'_0}(\theta)$  and each  $\varphi_r^{A_0 B_0}(\theta)$  is done once and for all at Steps 1 and 2. It is based on the partitioning of the objects, as seen in Section 3.2. Computation of the  $\varphi_r^{A'_4 B'_4}(\theta)$  and  $\varphi_r^{A_4 B_4}(\theta)$  values, at Steps 3.1 and 3.2.1.1.1, is very different in nature and can also be done using interpolation—which is much more efficient. To preserve precision, however, more directions should be considered when computing  $\varphi_r^{A'_0 B'_0}$  and  $\varphi_r^{A_0 B_0}$  than when computing  $\varphi_r^{A'_4 B'_4}$  and  $\varphi_r^{A_4 B_4}$ . We will not go into details here.

#### 4.4.2 Histogram Comparison (Step 3.2.1.1.2)

In practice, the strict equality condition would never be satisfied. Instead of checking whether or not the two histograms are equal, we should check whether or not they are “almost” equal. However, one might want to know *how*

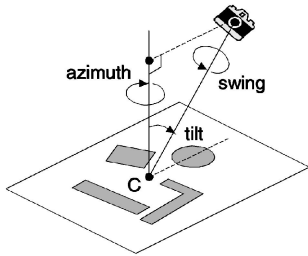


Fig. 10. The orientation of the camera is defined by three angles. The azimuth corresponds to a rotation about the vertical direction. The swing corresponds to a rotation about the principal ray. The tilt is the angle between the vertical and the principal ray.

well the object pairs  $(A_0, B_0)$  and  $(A'_0, B'_0)$  match, rather than whether they match or not. Hence, at Step 3.2.1.1.2, we decide to evaluate the similarity between the two histograms, using a similarity measure  $\mu$ . We therefore replace Steps 0 and 3.2.1.1.2 by:

0.  $\sigma \leftarrow 0$ ;  
 3.2.1.1.2. **If**  $\sigma < \mu(\varphi_r^{A_1 B_1}, \varphi_r^{A_0 B_0})$   
**then**  $R \leftarrow \rho, L \leftarrow \ell, K \leftarrow k, K' \leftarrow k'$   
 and  $\sigma \leftarrow \mu(\varphi_r^{A_1 B_1}, \varphi_r^{A_0 B_0})$ ;

At the end of the run,  $\sigma$  is the *matching degree* between  $\{A_0, B_0\}$  and  $\{A'_0, B'_0\}$ , i.e., between  $(A_0, B_0)$  and  $(A'_0, B'_0)$ , or between  $(B_0, A_0)$  and  $(A'_0, B'_0)$ . The higher  $\sigma$ , the better the match. There is an extensive literature on similarity measures [32], [33], [34], [35]. For our experiments in Section 5, we have examined over 20 measures and retained three: a Tversky index ( $\mu_T$ ), a Pappis' measure ( $\mu_P$ ), and normalized cross-correlation ( $\mu_C$ ). More about these can be found in Appendix F.1.

#### 4.4.3 Loops

The range of each loop variable needs to be discretized into a finite number of values. Different techniques, like the classical gradient descent, can be considered to reach the highest possible matching degree in a more efficient manner. Moreover, for given stretch ratios  $k$  and  $k'$ , the "best"  $\ell$  and  $\rho$  values,  $\ell_{k,k'}$ , and  $\rho_{k,k'}$ , can be computed directly, as shown in Appendix F.2. In other words, we can get rid of the two inmost loops and replace lines 3.2.1 and 3.2.1.1 by  $\ell \leftarrow \ell_{k,k'}$ ; and  $\rho \leftarrow \rho_{k,k'}$ . In practice, however, one might want to keep some flexibility and take  $\ell$  and  $\rho$  from small intervals centered on  $\ell_{k,k'}$  and  $\rho_{k,k'}$ . Note that the value of  $\ell$  is of no matter when using the similarity measure  $\mu_C$  (see Appendix F.2).

## 5 FORCE HISTOGRAMS ALSO REACT WELL TO "NEARLY" AFFINE TRANSFORMATIONS

In this section, we experimentally study the robustness of the theoretical tools presented in this paper to departures from the assumptions on the transformations being handled. In Section 4.3, we focused on affine transformations that can be decomposed into five basic transformations as follows:

$$\text{aff} = \text{stre}' \circ \text{tran} \circ \text{rot} \circ \text{sca} \circ \text{stre}.$$

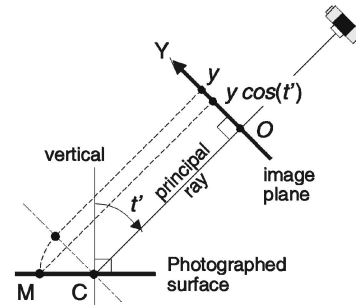


Fig. 11. Tilt and stretch. In the image plane, the photographed surface appears transformed under the X-axis orthogonal stretch with ratio  $\cos(t')$ , where  $t'$  denotes the tilt of the camera platform. The X-axis is perpendicular to the drawing and intersects the principal ray in O.

An interpretation of such a decomposition was given and illustrated in Fig. 3. Here, we rely on this interpretation to create appropriate test data sets and ensure that departures from assumptions are well controlled. The idea is presented in Section 5.1, and the data and tested implementations of the matching algorithm in Section 5.2. We then introduce the evaluation tools (Section 5.3) and discuss the results (Section 5.4).

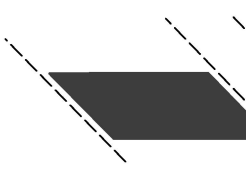
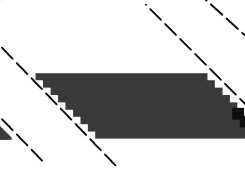
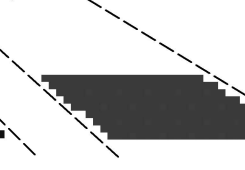
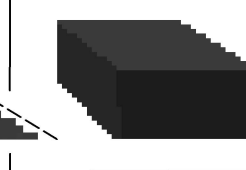

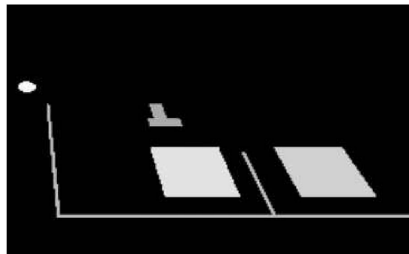


### 5.1 From Obscure Stretch Ratios to Daylight Photography

Consider Fig. 3. As mentioned in its caption, Figs. 3c and 3d can be seen as two different pictures of the same scene. If such was actually the case, the rotation angle  $\rho$  and the stretch ratios  $k$  and  $k'$  would be related to the orientation of the cameras during the shots. Camera orientation is usually defined by three angles: azimuth, tilt, and swing (Fig. 10). Let  $a$  and  $t$  be the azimuth and tilt of the camera platform during the first shot (Fig. 3c), and let  $a'$  and  $t'$  be the azimuth and tilt during the second shot (Fig. 3d). As can be understood from Fig. 11, the ratio of the stretch that transforms  $(A_4, B_4)$  into  $(A_5, B_5)$  is  $\cos(t')$ . In other words:  $k' = \cos(t')$ . Similarly:  $k = 1/\cos(t)$  (since the stretch that transforms  $(A_1, B_1)$  into  $(A_0, B_0)$  is  $\text{stre}^{-1}$ , and not  $\text{stre}$ ). Moreover,  $\rho$  is the azimuth difference:  $\rho = a' - a$ . Note that this interpretation of a decomposition like  $\text{aff} = \text{stre}' \circ \text{tran} \circ \text{rot} \circ \text{sca} \circ \text{stre}$  holds only if: 1)  $k$  is greater than or equal to 1 (logically, in a view from above like Fig. 3f, the objects cannot appear smaller than in another view like Fig. 3c), 2)  $k'$  is lower than or equal to 1 (same reason) and greater than 0 (unless the picture is taken from under the ground!), 3) the swing of each platform is  $+180$  degrees, i.e., a plumb line that intersects the principal ray is mapped to the Y-axis in the image plane (which is the case in most imaging situations).

Now, consider the algorithm described in Section 4.4. Assume the pairs  $\{A_0, B_0\}$  and  $\{A'_0, B'_0\}$  actually match, i.e., the output matching degree  $\sigma$  is 1 (or very close to 1). Assume they come from two different pictures of the same scene. Then, according to what we just said,  $R$ ,  $K$ , and  $K'$  give us information about the orientation of the cameras. According to Proposition 2, the difference between the azimuths of the two camera platforms is either  $R$ —in that case, the actual matching pairs are  $(A_0, B_0)$  and  $(A'_0, B'_0)$ —or  $\pi + R$ —the matching pairs are  $(B_0, A_0)$  and  $(A'_0, B'_0)$ .



TABLE 1  
The Five Data Sets

Set 1	Set 2	Set 3	Set 4	Set 5
vector data orthographic projection 2D scene objects synthetic data	<i>raster data</i> orthographic projection 2D scene objects synthetic data	raster data <i>perspective projection</i> 2D scene objects synthetic data	raster data perspective projection <i>3D scene objects</i> synthetic data	raster data perspective projection 3D scene objects <i>real data</i>
				
Scene 1 (one view)	Scene 2 (one view)	Scene 3 (one view)		
				

According to Proposition 1, if  $R$  is close to 0 or  $\pi$ , the product  $KK'$  is an estimate of  $\cos(t')/\cos(t)$ , where  $t$  denotes the tilt of the camera platform during the first shot and  $t'$  the tilt of the platform during the second shot; if  $R$  is close to  $-\pi/2$  or  $\pi/2$ , the ratio  $K/K'$  is an estimate of  $1/[\cos(t')\cos(t)]$ ; otherwise (general case),  $K$  is an estimate of  $1/\cos(t)$  and  $K'$  an estimate of  $\cos(t')$ .

In real photography, however, the image formation process includes perspective projection and not orthographic projection as implicitly assumed above; the photographed objects are usually 3D objects; digital imaging implies data digitization; even manual or interactive semiautomatic segmentation of image objects is not always accurate. Based on these observations, we propose to test the robustness of the theoretical tools described in the present paper using five data sets. The first set is composed of “perfect,” synthesized data: There is no departure from assumptions. Then, digitization, perspective projection, addition of a third dimension to the “photographed” objects, and, finally, image object segmentation are progressively included in a well-controlled data generation process. The question, of course, is how well the matching algorithm performs on the different data sets.

## 5.2 Data and Methods

The characteristics of the five test data sets are given in Table 1. Each set involves different bird’s-eye views of one of three scenes and each scene presents five objects: one tower, one pipe, one storehouse, and two stack buildings. Scene 3 is “real.” It corresponds to one part of the power plant at China Lake, California. The images 5.1, 5.2, and 5.3 shown in Fig. 12 represent three views of it. They were created from a set of data acquired from a surveillance

plane and provided by the Naval Air Warfare Center (NAWC). The original remote sensed images have been used for various purposes in different publications (see, e.g., [27]). Since the focus of our work is not on segmentation algorithms, we decided to segment them manually. The result, however, is not perfect. In image 5.2, for instance, part of the pipe is missing, and the bottom edge of the rightmost building is irregular. Moreover, some objects are clipped and small occlusions can be noticed. Scene 2 is a synthetic scene inspired by Scene 3. The objects in Scene 1 are the same as in Scene 2, except that the third dimension (height) has been “reset” to 0, i.e., they are 2D objects.

The data sets 1, 2, and 3 are composed each of six pairs of Scene 1 views, while Sets 4 and 5 gather respectively six pairs of Scene 2 views and two of Scene 3 views. Most views are shown in Fig. 12. The second data set is described by Table 2. Sets 1, 3, and 4 are composed of equivalent pairs (e.g., {3.1, 3.3}, {3.2, 3.5}, etc.), while Set 5 is composed of {5.1, 5.2} and {5.2, 5.3} only. Now, consider the two views in Figs. 13a and 13b. Each one defines three nonordered object pairs—hence, three force histograms—and any object pair in Fig. 13a corresponds to some object pair in Fig. 13b. Therefore, Figs. 13a and 13b lead to  $3 \times 3$  histogram comparisons. There is an actual match in three cases (dark continuous arrows) and there is no match in six cases (light dotted arrows; only three are shown). In fact, since the scenes considered in our experiments present five objects and not three as in Fig. 13, each pair of views (like {3.1, 3.3}, {5.1, 5.2}, etc.) leads to  $10 \times 10$  histogram comparisons. There is an actual match in 10 cases, and there is no match in 90 cases.

Twenty independent experiments were performed on each data set, using two types of histogram and 10 matching

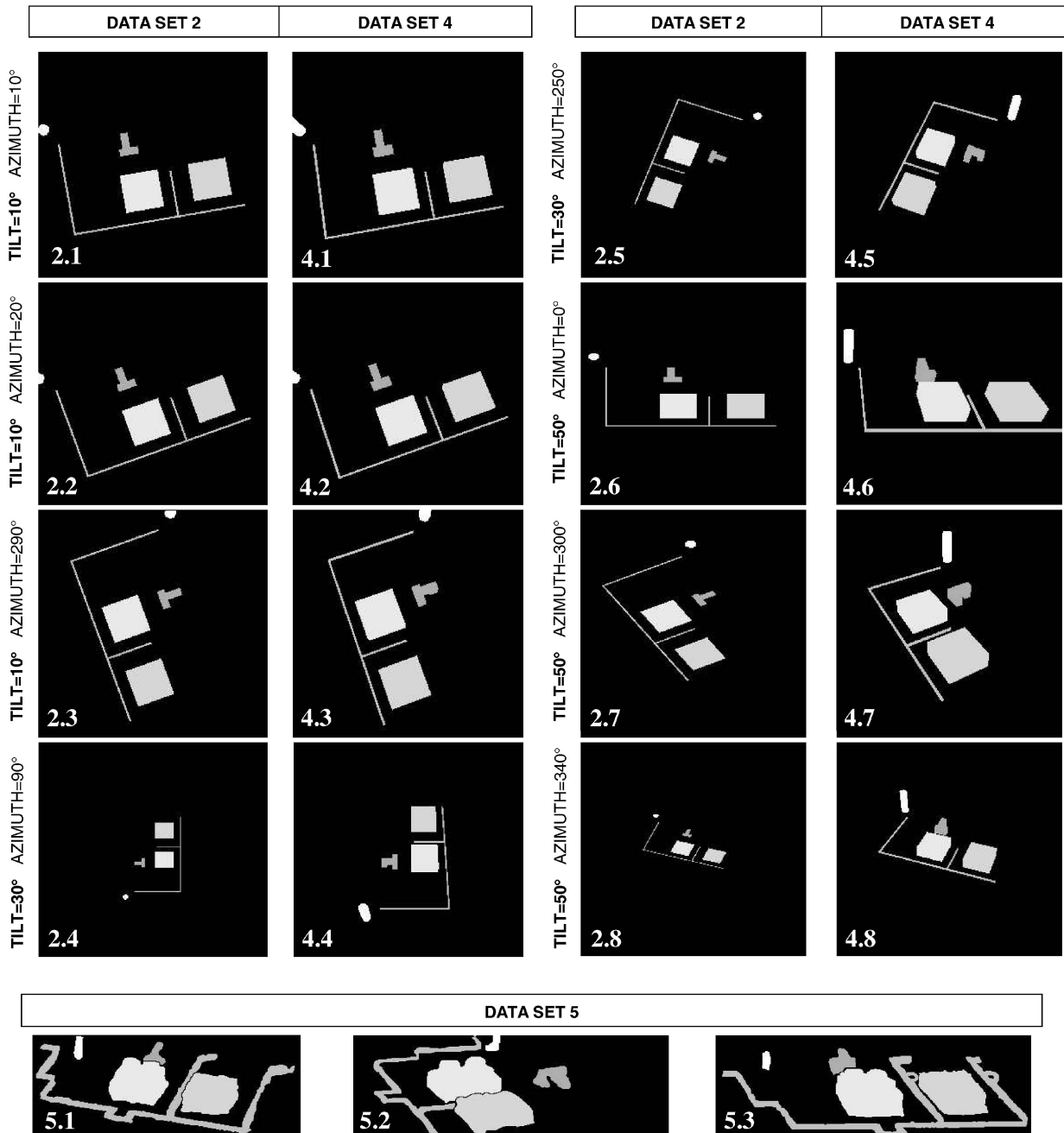


Fig. 12. The views considered in our study. All images are  $256 \times 256$ , except images 5.1 ( $399 \times 142$ ), 5.2 and 5.3 ( $470 \times 148$ ). Note that in some images, like 2.2 and 4.6, the tower and the pipe are accidentally clipped. The views in data sets 1 and 3 are similar to the views in data set 2, except that in set 1 vector data replace raster data (and the tower is never clipped), and in set 3 perspective projection replaces orthographic projection.

methods. The histograms are  $\varphi_0$  and  $\varphi_2$ -histograms. Other histograms could have been considered. However, as observed in [25], [27]: 1)  $\varphi_0$ -histograms coincide with angle histograms, which have been extensively used in the literature, 2) gravitational forces ( $\varphi_2$ ) are a reality of our physical world, 3) the  $\varphi_0$  and  $\varphi_2$ -histograms have very different and interesting characteristics which complement one another and allow for geometric interpretation. The 10 matching methods correspond to different implementations of the matching algorithm presented in Section 4.4. Their main characteristics are summarized by Table 3. *Step 3.2.1.1.2:* Methods 1, 2, 3, and 4 use the similarity measure  $\mu_T$ , Methods 5, 6, 7, and 8 use  $\mu_P$ , and Methods 9 and

10 use  $\mu_C$ . *Loops 3 and 3.2:* For all matching methods, the stretch ratios  $k$  and  $k'$  take their values in  $\{1/\cos t\}_{t \in \{0^\circ, 5^\circ, \dots, 60^\circ\}}$  and  $\{\cos t'\}_{t' \in \{0^\circ, 5^\circ, \dots, 60^\circ\}}$ , i.e., the tilts  $t$  and  $t'$  take their values in the interval  $[0^\circ, 60^\circ]$ , with 5 degree increments. Remember that  $k = 1/\cos t$  and  $k' = \cos t'$  (Section 5.1). *Loop 3.2.1:* For Methods 1 and 2, and 5 and 6, **For each**  $\ell$  is replaced by  $\ell \leftarrow \ell_{k,k'}$ ; as explained in Section 4.4.3. For Methods 3 and 4, and 7 and 8, the loop is kept, and  $\ell$  belongs to a small interval centered on  $\ell_{k,k'}$  (more precisely, it takes its values in

$$\{0.90\ell_{k,k'}, 0.91\ell_{k,k'}, \dots, 0.99\ell_{k,k'}, \ell_{k,k'}, \ell_{k,k'}/0.99, \ell_{k,k'}/0.98, \dots, \ell_{k,k'}/0.90\}.$$

TABLE 2  
The Six View Pairs in Data Set 2 and Their Main Numerical Features

PAIR	Tilt $t$	Tilt $t'$	Azimuth difference $a'-a$	Scaling factor ratio
{2.1, 2.3}	$10^\circ$	$10^\circ$	$280^\circ$	1/1
{2.2, 2.5}	$10^\circ$	$30^\circ$	$230^\circ$	3/4
{2.1, 2.8}	$10^\circ$	$50^\circ$	$330^\circ$	1/2
{2.4, 2.5}	$30^\circ$	$30^\circ$	$160^\circ$	3/2
{2.5, 2.6}	$30^\circ$	$50^\circ$	$110^\circ$	4/3
{2.6, 2.7}	$50^\circ$	$50^\circ$	$300^\circ$	1/1

For Methods 9 and 10, *For each*  $\ell$  is replaced by  $\ell \leftarrow 1$ ; because the value of  $\ell$  is of no matter when using  $\mu_C$  (Section 4.4.3). *Loop 3.2.1.1*: For Methods 1, 3, 5, 7, and 9, *For each*  $\rho$  is replaced by  $\rho \leftarrow \rho_{k,k'}$ ; (except that the angle in degrees is rounded to the closest even integer). For the other methods, the loop is kept and  $\rho$  belongs to an interval centered on  $\rho_{k,k'}$  (in fact, it takes its values in the whole set  $\{0^\circ, 2^\circ, \dots, 358^\circ\}$ ). *Steps 1, 2, 3.1, and 3.2.1.1.1*: The histograms  $\varphi_r^{A_i B_i}$  and  $\varphi_r^{A'_i B'_i}$  are represented by 180 values each (an appropriate choice, according to the experiments in [25]), while  $\varphi_r^{A_0 B_0}$  and  $\varphi_r^{A'_0 B'_0}$  are represented by 360 values to preserve precision when the values  $\varphi_r^{A_i B_i}(\theta)$  and  $\varphi_r^{A'_i B'_i}(\theta)$  are computed using interpolation (Section 4.4.1).

### 5.3 Evaluation Tools

Consider an experiment over a given data set, using a given matching method and a given type of histogram, say, Set 1, Method 1, and  $\varphi_0$ -histograms. Since the data set is composed of six pairs of Scene 1 views, the algorithm described in Section 4.4 has to be run 600 times ( $6 \times 100$  histogram comparisons have to be performed). There is an actual match in 60 cases ( $6 \times 10$ ) and no match in 540 cases ( $6 \times 90$ ). Four values are computed to evaluate the results:

the area under the ROC curve and the error rates  $\Delta R$ ,  $\Delta T$ , and  $\Delta L$ .

#### 5.3.1 ROC Curve

Let  $\{A, B\}$  and  $\{A', B'\}$  be two object pairs. We may decide that the two pairs match if and only if the matching degree  $\sigma$  output by the matching algorithm is greater than or equal to some threshold  $\tau$ . There are four cases:

1.  $\{A, B\}$  and  $\{A', B'\}$  actually match and  $\sigma \geq \tau$ : The right decision is taken; we have a *true positive*.
2.  $\{A, B\}$  and  $\{A', B'\}$  actually match and  $\sigma < \tau$ : The wrong decision is taken; we have a *false negative*.
3.  $\{A, B\}$  and  $\{A', B'\}$  do not match and  $\sigma \geq \tau$ : The wrong decision is taken; we have a *false positive*.
4.  $\{A, B\}$  and  $\{A', B'\}$  do not match and  $\sigma < \tau$ : The right decision is taken; we have a *true negative*.

Let  $TP(\tau)$  be the number of True Positives and  $FP(\tau)$  be the number of False Positives. The *Receiver Operating Characteristic curve*—or *ROC curve*—is a plot of the true positive rate against the false positive rate for the different possible thresholds. In other words, its points are the points of coordinates  $(TP(\tau)/60, FP(\tau)/540)$ , where  $\tau$  belongs to the interval  $[0,1]$ . The closer the curve follows the left-hand border and then the top border of the ROC space, the better. The area under the curve is a measure of accuracy. If the area is 1, some threshold separates the 60 object pairs that match from the 540 pairs that do not match, i.e., the result is perfect. If the area is 0.5, the result is worthless. A rough guide for classifying the accuracy is the traditional academic point system:

- A. “excellent” if the area is greater than 0.9,
- B. “good” if it is greater than 0.8,
- C. “fair” if greater than 0.7,
- D. “poor” if greater than 0.6, and
- E. “fail” otherwise.

#### 5.3.2 $\Delta R$ , $\Delta T$ , and $\Delta L$

Consider again two object pairs  $\{A, B\}$  and  $\{A', B'\}$ , and assume that they actually match. Let  $(a, t, s)$  be the set of parameters that defines the view from which  $A$  and  $B$  come:  $a$  is the azimuth of the camera platform,  $t$  is the tilt, and  $s$  is the image scaling factor. Similarly, let  $(a', t', s')$  be the set of parameters that defines the second view. Now, let  $R$  be the rotation angle,  $L$  the scaling ratio, and  $K$  and  $K'$  the stretch ratios output by the matching algorithm. As seen in Section 5.1,  $R$  is an approximation of  $a' - a$  (modulo  $\pi$ ),  $\text{acos}(K^{-1})$  is an approximation of  $t$ ,  $\text{acos}(K')$  an approximation of  $t'$ , and  $L$  an approximation of  $s'/s$ .

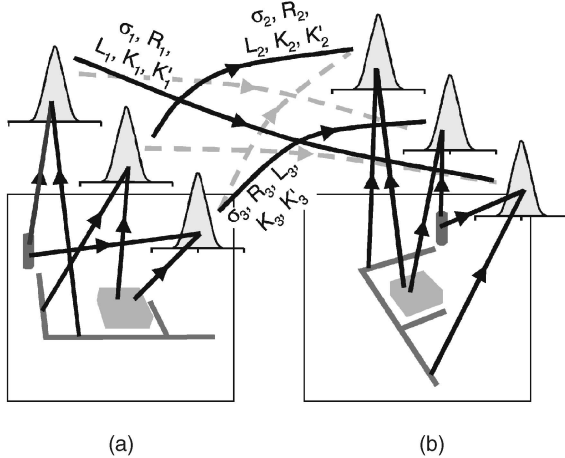


Fig. 13. In this example, each view defines a set of three force histograms. The two views lead to  $3 \times 3 = 9$  histogram comparisons. There is an actual match in three cases (as shown by the continuous lines that join the histograms), and there is no match in six cases (the dotted lines indicate three of them). Each comparison gives a 5-tuple  $(\sigma_i, R_i, L_i, K_i, K'_i)$ . Three 5-tuples are shown here. Note that there exist  $3! = 6$  one-to-one mappings between the two sets of histograms. The three continuous lines constitute one of these mappings, and the three dotted lines another one.

TABLE 3  
The 10 Matching Methods

Method	1	2	3	4	5	6	7	8	9	10
Similarity measure $\mu$	$\mu_T$				$\mu_P$				$\mu_C$	
Scaling factor $\ell$	fixed		variable		fixed		variable		1	
Rotation angle $\rho$	fixed	variable	fixed	variable	fixed	variable	fixed	variable	fixed	variable

Let  $\{\{A, B\}, \{A', B'\}\}$  be alternately each of the 60 pairs of object pairs that actually match. The value  $\Delta R$  is the average absolute difference (modulo  $\pi$ ) between  $R$  and  $a' - a$ . It indicates to which extent the matching algorithm allows the azimuth difference to be retrieved.  $\Delta T$  is the average value of  $(|\text{acos}(K^{-1}) - t| + |\text{acos}(K') - t'|)/2$ . It indicates to which extent the tilts can be retrieved. Finally,  $\Delta L$  is the average value of  $|\log_{10}(L/(s'/s))|$ . It indicates to which extent the scaling factor ratio can be retrieved. Assume for instance that  $s'/s$  is 1 (i.e., the actual scales of the images are the same). A value  $L$  of 2 means that the scaling factor of one image has been mistakenly found twice the scaling factor of the other image. A value  $L$  of 0.5 means the same (up to a permutation of the two images) and the error should therefore be quantified the same. This explains the presence of the absolute value of the logarithm in the above formula. The lower  $\Delta R$ ,  $\Delta T$ , and  $\Delta L$ , the better. Zero represents a perfect result.

#### 5.4 Results

In the experiments over the first data set, each one of the 180 values that represent  $\varphi_r^{A_1B_1}$  (Step 3.2.1.1.1) is calculated exactly, i.e., without using interpolation (Section 4.4.1). The same applies to  $\varphi_r^{A'_1B'_1}$ . Therefore, these experiments over “perfect” data are performed under “ideal” conditions. The results, shown in Table 4 and Fig. 14, beautifully validate the theoretical work. One might point out that some ROC areas are lower than 1.0000. However, this should not be misinterpreted. Consider, for instance, the experiment with  $\varphi_0$ -histograms and matching method 10. The lowest similarity obtained from the 60 matching cases is 0.999991, whereas the highest similarity obtained from the 540 non-matching cases is 0.999356. Consequently, the object pairs that match can be separated from the pairs that do not match, but the separation is really thin and the threshold increment used for constructing the ROC curves (0.001) did

TABLE 4  
Results for Data Set 1

Method	$\varphi_0$ -Histograms				$\varphi_2$ -Histograms			
	ROC	$\Delta T$	$\Delta R$	$\Delta L$	ROC	$\Delta T$	$\Delta R$	$\Delta L$
1	1.0000	0.00	0.00	0.00	1.0000	0.00	0.00	0.00
2	1.0000	0.00	0.00	0.00	1.0000	0.00	0.00	0.00
3	1.0000	0.00	0.00	0.00	1.0000	0.00	0.00	0.00
4	1.0000	0.00	0.00	0.00	1.0000	0.00	0.00	0.00
5	1.0000	0.00	0.00	0.00	1.0000	0.00	0.00	0.00
6	1.0000	0.00	0.00	0.00	1.0000	0.00	0.00	0.00
7	1.0000	0.00	0.00	0.00	1.0000	0.00	0.00	0.00
8	1.0000	0.00	0.00	0.00	1.0000	0.00	0.00	0.00
9	0.9981	0.00	0.00	0.00	0.9981	0.00	0.00	0.00
10	0.9981	0.00	0.00	0.00	0.9981	0.00	0.00	0.00

not allow it to be correctly detected. This illustrates a problematic feature of  $\mu_C$ , the similarity measure on which Methods 9 and 10 rely: It tends to produce extremely high values, very close to each other.

In the experiments over the data sets 2 to 5, the histograms  $\varphi_r^{A_1B_1}$  and  $\varphi_r^{A'_1B'_1}$  are computed by using interpolation (Section 4.4.1). The results on Set 2 (Table 5, Fig. 15) show how data rasterization and histogram interpolation contribute to the decline in matching performances. Perfection is replaced by excellence. There is only a few thousandths drop in accuracy, the  $\Delta T$  values are about the 5 degree increment used to search for the best tilts, and the  $\Delta R$  values are less than the 2 degree rotation angle increment. As explained in Section 5.2, the error rates  $\Delta T$ ,  $\Delta R$ , and  $\Delta L$  are computed considering the 60 matching cases only. It is interesting to note that the same computations considering “only” the 540 *nonmatching* cases give values about 25 degrees (tilts), 40 degrees (rotation angle), and 0.6 or higher (scaling factor ratio). The differences with the error rates shown in Table 5 are significant. With the other data sets, values similar to the three above (25°, 40°, 0.6) are obtained. The differences with the error rates come down (since  $\Delta T$ ,  $\Delta R$ , and  $\Delta L$  naturally increase when the data get less and less perfect), but remain nonnegligible (see Tables 6 and 7).

Replacing orthographic projection with perspective projection results in a decrease of 1 to 2 percent in accuracy (Table 6, Fig. 16). The error rates  $\Delta T$  and  $\Delta R$  more than double and become higher than the 5 and 2 degree angle increments. The adding of a third dimension to the objects translates into more noticeable drops in ROC areas (Table 7, Fig. 17): 4 to 8 percent for experiments using  $\varphi_0$ -histograms, and 8 to 10 percent for experiments using  $\varphi_2$ -histograms. Overall, the accuracy is still excellent. However, the error rates  $\Delta T$  and  $\Delta R$  double again, more or less. Now, the tilts are poorly retrieved and, when using histograms of gravitational forces, the scaling factor ratio is not well recovered either (note that  $0.3 \sim \log(2/1)$ ). Naturally, with

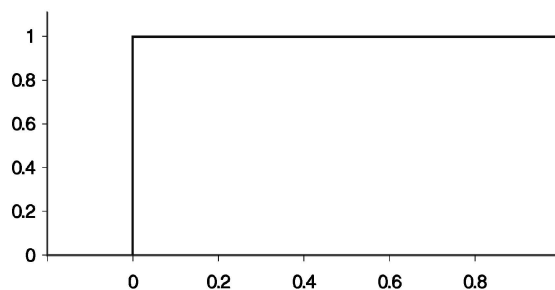


Fig. 14. Perfect ROC curves for data set 1.

TABLE 5  
Results for Data Set 2

Method	$\varphi_0$ -Histograms				$\varphi_2$ -Histograms			
	ROC	$\Delta T$	$\Delta R$	$\Delta L$	ROC	$\Delta T$	$\Delta R$	$\Delta L$
1	0.9987	4.00	1.01	0.03	0.9988	5.54	1.56	0.10
2	0.9986	3.38	0.73	0.03	0.9982	5.33	1.10	0.11
3	0.9987	4.00	1.01	0.04	0.9988	6.17	1.64	0.10
4	0.9986	3.46	0.73	0.03	0.9984	5.04	1.13	0.10
5	0.9988	4.00	1.01	0.03	0.9989	5.54	1.56	0.10
6	0.9985	3.38	0.73	0.03	0.9981	5.33	1.10	0.11
7	0.9986	4.00	1.01	0.04	0.9988	6.17	1.64	0.10
8	0.9985	3.46	0.73	0.03	0.9982	5.04	1.13	0.10
9	0.9959	3.88	1.01	0.03	0.9980	6.00	1.56	0.10
10	0.9958	2.83	0.57	0.03	0.9971	5.13	1.07	0.10

the real data set (Set 5), further decrease in performance is expected. However, despite, at this point, a drastic departure from theoretical assumptions, the accuracy remains “fair” to “good” (Table 8, Fig. 18). Since the images provided by the NAWC did not come with the pose parameters, those error rates were not computed.

For each data set, we may wonder which matching method and type of histogram lead to highest accuracy, which ones lead to best tilt retrieval, etc. The “mathematical” answer to these questions is given by the shadowed boxes in Tables 5, 6, 7, and 8. But, most experiments produce similar results. As a general rule (see Tables 5, 6 and 7), the pose parameters, and especially the scaling factor ratios, are best recovered when using histograms of constant forces. The  $\varphi_2$ -histogram, which focuses on the closest parts of the objects, seems to be more sensitive to deviation from theoretical assumptions. Because they search for the best rotation angles and/or the best scaling factor ratios, we expected even numbered methods to give lower error rates  $\Delta T$  and  $\Delta R$ . In fact, the results are not significantly better (see Tables 5 and 6), and do not always confirm our expectation (see Table 7, two last columns). Finally, we are inclined to systematically use matching Method 1 and  $\varphi_0$ -histograms. By avoiding the two innermost loops of the matching algorithm, Method 1 drastically reduces the computational burden. Moreover, it relies on similarity measure  $\mu_T$  (Tanimoto index) which seems to perform at least as well as the other measures, and is the simplest one.

In conclusion, the theoretical tools presented in this paper appear to be fairly robust. Even with seriously violated assumptions, the matching algorithm is able to

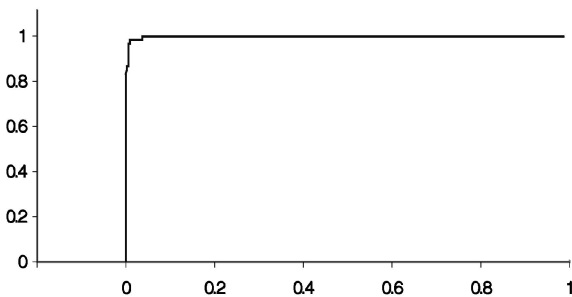


Fig. 15. The best ROC curve for data set 2 (Method 5,  $\varphi_2$ -histograms).

TABLE 6  
Results for Data Set 3

Method	$\varphi_0$ -Histograms				$\varphi_2$ -Histograms			
	ROC	$\Delta T$	$\Delta R$	$\Delta L$	ROC	$\Delta T$	$\Delta R$	$\Delta L$
1	0.9804	10.67	3.31	0.06	0.9933	9.42	3.40	0.11
2	0.9806	10.67	3.23	0.06	0.9943	9.79	3.30	0.12
3	0.9808	10.33	3.25	0.06	0.9940	9.42	3.44	0.12
4	0.9807	10.42	3.13	0.06	0.9952	9.75	3.30	0.12
5	0.9803	10.67	3.31	0.06	0.9934	9.42	3.40	0.11
6	0.9804	10.67	3.23	0.06	0.9944	9.79	3.30	0.12
7	0.9806	10.33	3.25	0.06	0.9939	9.42	3.44	0.12
8	0.9806	10.42	3.13	0.06	0.9950	9.75	3.30	0.12
9	0.9754	10.13	3.22	0.06	0.9917	9.46	3.20	0.12
10	0.9750	10.21	3.27	0.06	0.9931	10.25	3.17	0.12

separate “matching” object pairs from “nonmatching” pairs reasonably well, and also to give useful information about the “nearly” affine transformation two object pairs match through.

## 5.5 Remarks

### 5.5.1 Cases of Multiple Possible Poses

For each pair of views listed in Table 2, calculate the angular distance between the azimuth difference and the closest cardinal direction:  $|280^\circ - 270^\circ| = 10^\circ$ ,  $|230^\circ - 270^\circ| = 40^\circ$ ,  $|330^\circ - 360^\circ| = 30^\circ$ ,  $|160^\circ - 180^\circ| = 20^\circ$ ,  $|110^\circ - 90^\circ| = 20^\circ$ , and  $|300^\circ - 270^\circ| = 30^\circ$ . None of these values is zero. Therefore, according to Proposition 1, a unique set of parameters  $\rho$ ,  $\ell$ ,  $k$ , and  $k'$  (rotation angle, scaling factor, and stretch ratios) can be associated with any given pair. Considering such view pairs was appropriate since we wanted to evaluate the errors in recovering these parameters. However, one should be aware that if the angular distance mentioned above is zero (or “close” to zero), then the recovered tilts and scaling factor ratio do not necessarily correspond to the actual ones. For instance, we processed two views (vector data) with the following features: azimuth difference  $0^\circ$ , first tilt  $10^\circ$ , second tilt  $50^\circ$ . Depending on the matching method and on the object pairs, the retrieved parameters were either  $0^\circ, 10^\circ, 50^\circ$  or  $0^\circ, 40^\circ, 60^\circ$ . This is in perfect accordance with Proposition 1 2) since  $\cos 50^\circ / \cos 10^\circ = \cos 60^\circ / \cos 40^\circ$ .

TABLE 7  
Results for Data Set 4

Method	$\varphi_0$ -Histograms				$\varphi_2$ -Histograms			
	ROC	$\Delta T$	$\Delta R$	$\Delta L$	ROC	$\Delta T$	$\Delta R$	$\Delta L$
1	0.9400	16.58	6.84	0.09	0.9105	18.96	9.82	0.31
2	0.9385	16.79	5.70	0.09	0.9101	17.71	11.43	0.33
3	0.9388	15.75	6.61	0.09	0.9094	18.63	9.81	0.31
4	0.9361	16.42	5.70	0.09	0.9074	17.96	11.10	0.32
5	0.9399	16.58	6.84	0.09	0.9103	18.96	9.82	0.31
6	0.9384	16.79	5.70	0.09	0.9098	17.71	11.43	0.33
7	0.9389	15.75	6.61	0.09	0.9091	18.63	9.81	0.31
8	0.9360	16.42	5.70	0.09	0.9071	17.96	11.10	0.32
9	0.9245	15.83	6.75	0.09	0.9052	17.75	9.59	0.33
10	0.9240	16.17	5.47	0.09	0.8996	17.75	11.03	0.32

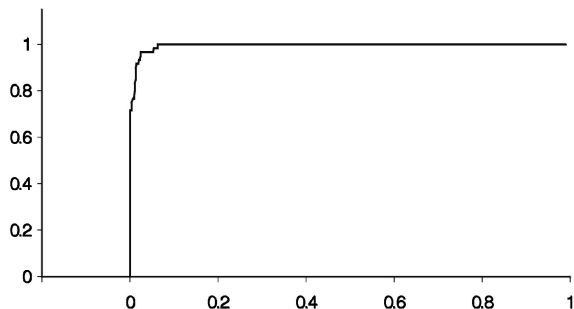


Fig. 16. The best ROC curve for data set 3 (Method 4,  $\varphi_2$ -histograms).

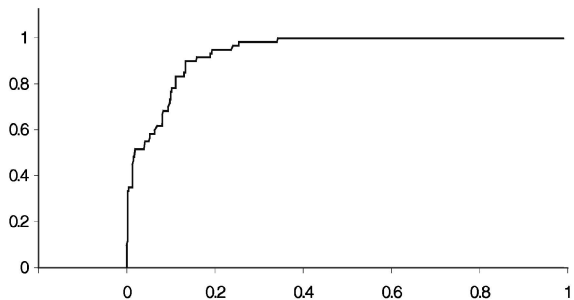


Fig. 17. The best ROC curve for data set 4 (Method 1,  $\varphi_0$ -histograms).

### 5.5.2 Search for the One True Mapping

Consider the two views Figs. 13a and 13b. Each defines a set of three  $\varphi_0$ -histograms and there exist  $3! = 6$  one-to-one mappings between the two sets, i.e., six ways to match the two views. Consider any mapping. Each histogram from view Fig. 13a can be compared with its associated histogram from Fig. 13b using matching Method 1. The output is a 5-tuple  $(\sigma_i, R_i, L_i, K_i, K'_i)$ , where  $i$  belongs to  $\{1, 2, 3\}$ . The one true mapping should be such that the  $\sigma_i$  are close to 1, the  $R_i$  are close to each other (modulo  $\pi$ ), the  $L_i$  are close to each other, etc. A scene matching degree, based on the computation, normalization, and averaging of mean deviations, can therefore be derived from the three tuples  $(\sigma_i, R_i, L_i, K_i, K'_i)$ . It is better, however, to ignore the values  $K_i$  and  $K'_i$  in the scoring equation because the tilts are not always well retrieved (see Table 7). A first experiment involved the views 5.1 and 5.2 of the power plant at China Lake (Fig. 12). Each one leads to the computation of a set of 10  $\varphi_0$ -histograms, instead of three as in Fig. 13. Therefore, there are  $10! = 3,628,800$  ways to match the two views. Of course, the search space could be drastically reduced since the histograms of each set are not totally independent (they are associated with object pairs that share the same five objects). However, we did not care about such “details,” and the 3,628,800 cases were processed exhaustively. The one true mapping produced the highest scene matching degree. We repeated the experiment with the views 5.2 and 5.3. Once again, the one true match produced the highest matching degree. Then, we tested 14 other scoring equations, i.e., using only the  $\sigma_i$  values, or only the  $\sigma_i$  and the  $R_i$ , or all parameters including the retrieved stretch ratios, etc. None of these other matching degrees led to the same perfect result. Note that each scene matching (i.e., complete processing of one pair of views) took about 6 minutes. The computation of one single  $\varphi_0$ -histogram took

TABLE 8  
Results for Data Set 5

Method	$\varphi_0$ -Histograms	$\varphi_2$ -Histograms
	ROC	ROC
1	0.8083	0.8026
2	0.7940	0.7936
3	0.8079	0.8046
4	0.7908	0.7929
5	0.8076	0.8015
6	0.7935	0.7931
7	0.8074	0.8039
8	0.7900	0.7928
9	0.7860	0.7929
10	0.7615	0.7875

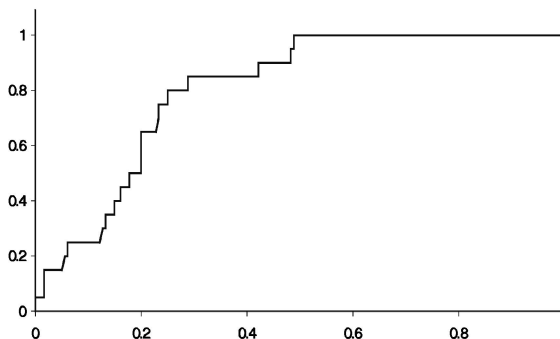


Fig. 18. The best ROC curve for data set 5 (Method 1,  $\varphi_0$ -histograms).

about 1 second on an Athlon 700 MHz PC (20 seconds for the 20 histograms);<sup>5</sup> the comparison of two histograms using matching Method 1 took about 2 seconds (3 minutes 20 seconds for the 100 comparisons); the computation and comparison of the 3,628,800 matching degrees took about 2 minutes on a Pentium III 500 MHz.

## 6 CONCLUSION

As color, texture and shape, relative position is a fundamental concept in computer vision. We argue that, as color, texture, and shape affine invariant descriptors, affine invariant relative position descriptors have a role to play. The histogram of forces was designed to quantitatively represent the position of an object with respect to another. In this paper, we have shown that it can be normalized to achieve invariance under translations, rotations, and scalings, i.e., similarity transformations. Moreover, we have proved that, basically, any two of the following elements, 1) an affine transformation, 2) a relative position (described through a histogram of forces), and 3) the “transformed” relative position, allow the third one to be recovered. An experimental study, for which 700 histograms were computed and more than 50,000 histogram comparisons performed, has validated the theoretical results. It has also shown that the descriptor is fairly robust to departures from the assumptions on the transformations being handled. The affinity that “best” approximates the “nearly” affine transformation two relative positions are related through can be retrieved (up to a translation) and the quality of the

5. All the programs were written in C without excessive attention to optimization. Moreover, our basic implementation does not allow multiple object pairs to be handled simultaneously.

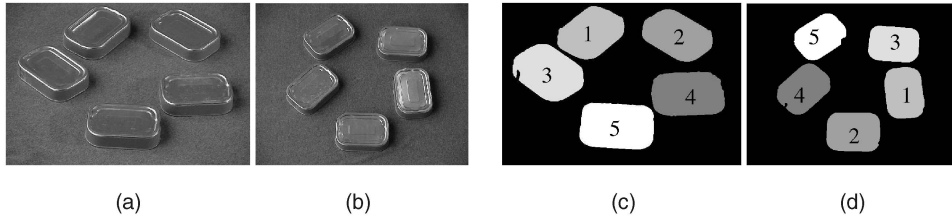


Fig. 19. Relative position descriptors are orthogonal to color, texture, and shape descriptors. (a) The two RGB pictures were taken with a commercial digital camera. Which can is which? (b) Here, segmentation was achieved by choosing the color channel with the best contrast (red channel), running an optimum thresholding algorithm (like Otsu's) on the corresponding gray-level histogram and performing  $7 \times 7$  median filtering on the thresholded image. The correct mapping was found using the same method as in Section 5.5.2.

approximation assessed (through a matching degree). Relative position descriptors like the histogram of forces are orthogonal—and, therefore, constitute a natural complement—to color, texture, and shape descriptors. In our last experiment, for instance, a scene matching problem was translated into the search for a one-to-one mapping among three and a half million others (Section 5.5.2). The search was based solely on force histogram comparisons, i.e., without using any information on color, texture, or shape. The mapping, however, was found. Fig. 19 provides an even better illustration of the above-mentioned orthogonality. In this other scene matching problem—solved as in Section 5.5.2—color, texture, and shape descriptors would clearly not be very helpful.

Technically, the histogram of forces resembles more region-based shape descriptors than boundary-based descriptors—although it allows data to be stored and efficiently processed in vector form as well as in raster form. There are virtually no constraints on the objects. Intersecting concave objects with holes and multiple connected components do not need special care. No preprocessing is required (like recovery of the boundary, extraction of interest points, encoding, filtering, and sampling). One should also point out that additivity in feature space is enforced, i.e., the  $\varphi_r$ -histogram  $\varphi_r^{(A_1 \cup A_2)B}$  is equal to  $\varphi_r^{A_1B} + \varphi_r^{A_2B}$ . This is a most useful property because image objects are often determined through oversegmentation and merging procedures. Conceptually, the histogram of forces is very different than shape descriptors. It is sensitive to the shape of the objects, but only because shape affects relative position, as color affects texture. A force histogram does not allow anything to be said on the individual shapes. By construction, however, it allows propositions such as “object A is to the right of object B” to be assessed [25] and, conversely, linguistic descriptions such as “A is mostly to the right of B but a little above” to be generated [27]. A “subversive” use of the histogram of forces would be the description of the position of an object with respect to *itself* (remember that intersecting objects can be handled when  $r$  is lower than 1). This could be exploited in affine invariant shape recognition and classification problems. For these particular  $\varphi_r$ -histograms (also called  $\varphi_r$ -signatures [28]), a comparative study with classical shape descriptors could be conducted, using test images that have been popular in the literature (like images of military airplanes, see, e.g., [12]).

Much remains to be done. Obviously, in practical applications, various descriptors should cooperate. The results presented in this paper suggest that the histogram of

forces could be of great use in scene matching and would yield powerful edge attributes in attributed relational graphs.<sup>6</sup> We plan to investigate the subject in the near future. We would also like to investigate the histogram of forces—which is able to handle fuzzy objects [25]—as a descriptor of relative position between gray-level objects. Moreover, the inverse problem has not been solved yet (given a relative position, i.e., a force histogram, construct all pairs of objects this histogram is associated with). Finally, although the histogram of forces reacts “well” to affine transformations, in a mathematically predictable way, normalized histograms are similarity invariant only. Relative position descriptors truly invariant under arbitrary affine transformations still have to be found.

## APPENDIX A

### PROOF OF PROPOSITION 1

This appendix will be published in the IEEE PAMI digital library available at <http://computer.org/publications/dlib>.

## APPENDIX B

### SOME PROPERTIES OF FUNCTION $F_r$

This appendix will be published in the IEEE PAMI digital library available at <http://computer.org/publications/dlib>.

## APPENDIX C

### PROOF OF PROPERTY 5

This appendix will be published in the IEEE PAMI digital library available at <http://computer.org/publications/dlib>.

## APPENDIX D

### PROOF OF PROPOSITION 2

Consider the objects  $A = \text{stre}'(\text{tran}(\text{rot}(\text{sca}(\text{stre}(A_0))))))$  and  $B = \text{stre}'(\text{tran}(\text{rot}(\text{sca}(\text{stre}(B_0))))))$ . Equation (8) becomes  $\varphi_r^{A'_0 B'_0} = \varphi_r^{AB}$ , i.e.,  $(A'_0, B'_0) \in \mathcal{O}^{AB}$  (Section 4.2). Since  $\mathcal{O}$  is  $r$ -regular (Definition 3), only four cases have to be examined. Assume, for instance, that  $r$  is 3. Then,  $(A'_0, B'_0)$  might well be

<sup>6</sup> In such a graph [36], [37], [38], [39], a node represents an image region and is given a list of attributes (e.g., color, texture, shape). An edge represents the relations between two regions and is also with attributes (e.g., relative position).

a pair like  $(\text{tran}_1(\text{rot}_1(\text{sca}_1(\text{B}))), \text{tran}_1(\text{rot}_1(\text{sca}_1(\text{A}))))$ , where  $\text{rot}_1$  denotes a  $\pi$ -angle rotation. In that case, we have:

$$A'_0 = \text{tran}_1(\text{rot}_1(\text{sca}_1(\text{stre}'(\text{tran}(\text{rot}(\text{sca}(\text{stre}(\text{B}_0)))))))$$

$$\text{and } B'_0 = \text{tran}_1(\text{rot}_1(\text{sca}_1(\text{stre}'(\text{tran}(\text{rot}(\text{sca}(\text{stre}(\text{A}_0))))))).$$

Function composition is not commutative. However, here, because of the nature of the transformations involved and since  $\text{rot}_1$  is a  $\pi$ -angle rotation and not any rotation, it is possible to find a translation  $\text{tran}_2$  such that:

$$\text{tran}_1 \circ \text{rot}_1 \circ \text{sca}_1 \circ \text{stre}' \circ \text{tran} \circ \text{rot} \circ \text{sca} \circ \text{stre}$$

$$= \text{stre}' \circ \text{tran}_2 \circ \text{rot}_1 \circ \text{rot} \circ \text{sca}_1 \circ \text{sca} \circ \text{stre}.$$

Let  $\text{rot}_2$  and  $\text{sca}_2$  be the compound transformations  $\text{rot}_1 \circ \text{rot}$  and  $\text{sca}_1 \circ \text{sca}$ . The angle of  $\text{rot}_2$  is  $\pi + \rho$ . We now can write:

$$A'_0 = \text{stre}'(\text{tran}_2(\text{rot}_2(\text{sca}_2(\text{stre}(\text{B}_0)))))$$

$$\text{and } B'_0 = \text{stre}'(\text{tran}_2(\text{rot}_2(\text{sca}_2(\text{stre}(\text{A}_0))))).$$

In other words,  $(A'_0, B'_0)$  matches  $(B_0, A_0)$  through  $(\text{stre}', \text{tran}_2, \text{rot}_2, \text{sca}_2, \text{stre})$ . The three other cases can be handled in the same way.

## APPENDIX E

### PROOF OF PROPOSITION 3

Let us use the notations introduced in Fig. 3:  $A_1 = \text{stre}(A_0)$ ,  $B_3 = \text{rot}(B_2)$ , etc. Note that, for convenience's sake, the object  $\text{tran}(A_3)$  is referred to by two symbols,  $A_4$  and  $A'_4$ . The same applies to  $\text{tran}(B_3)$ . Equation (8) then becomes  $\varphi_r^{A'_0 B'_0} = \varphi_r^{\text{stre}'(A'_4) \text{stre}'(B'_4)}$  and, according to Property 5, can be rewritten as follows:

$$\forall \alpha \in \mathbb{R},$$

$$\varphi_r^{A'_0 B'_0}(\alpha) = k^{2-r} [1 + (k'^2 - 1) \cos^2 \alpha]^{(r-1)/2} \varphi_r^{A'_4 B'_4}(\alpha_{|k'|}). \quad (12)$$

Solving  $\theta = \alpha_{|k'|}$  for  $\alpha$  gives  $\alpha = \theta_{|k^{-1}|}$  and allows us to "reverse" (12). After simplification:

$$\forall \theta \in \mathbb{R},$$

$$\varphi_r^{A'_4 B'_4}(\theta) = k^{r-2} [1 + (k'^{-2} - 1) \cos^2 \theta]^{(r-1)/2} \varphi_r^{A'_0 B'_0}(\theta_{|k^{-1}|}). \quad (13)$$

Moreover:  $\varphi_r^{A_1 B_4} = \varphi_r^{\text{tran}(A_3) \text{tran}(B_3)}$ ,  $\varphi_r^{A_3 B_3} = \varphi_r^{\text{rot}(A_2) \text{rot}(B_2)}$ ,  $\varphi_r^{A_2 B_2} = \varphi_r^{\text{sca}(A_1) \text{sca}(B_1)}$ , and  $\varphi_r^{A_1 B_1} = \varphi_r^{\text{stre}(A_0) \text{stre}(B_0)}$ . Using successively Properties 2, 3, 4, and 5, we can rewrite these equations as follows:

$$\forall \theta \in \mathbb{R}, \varphi_r^{A_1 B_4}(\theta) = \varphi_r^{A_3 B_3}(\theta),$$

$$\forall \theta \in \mathbb{R}, \varphi_r^{A_3 B_3}(\theta) = \varphi_r^{A_2 B_2}(\theta - \rho),$$

$$\forall \theta \in \mathbb{R}, \varphi_r^{A_2 B_2}(\theta) = \ell^{3-r} \varphi_r^{A_1 B_1}(\theta), \text{ and}$$

$$\forall \theta \in \mathbb{R}, \varphi_r^{A_1 B_1}(\theta) = k^{2-r} [1 + (k^2 - 1) \cos^2 \theta]^{(r-1)/2} \varphi_r^{A_0 B_0}(\theta_{|k|}). \quad (14)$$

Successive substitutions then yield (15) and (16) below; equating (13) and (15) yields (17), which will prove useful in Appendix F.2; finally, equating (13) and (16) gives (9), the equation we were looking for.

$$\forall \theta \in \mathbb{R}, \varphi_r^{A_1 B_4}(\theta) = \ell^{3-r} \varphi_r^{A_1 B_1}(\theta - \rho). \quad (15)$$

$$\forall \theta \in \mathbb{R}, \varphi_r^{A_1 B_4}(\theta) =$$

$$\ell^{3-r} k^{2-r} [1 + (k^2 - 1) \cos^2(\theta - \rho)]^{(r-1)/2} \varphi_r^{A_0 B_0}((\theta - \rho)_{|k|}). \quad (16)$$

$$\forall \theta \in \mathbb{R}, \varphi_r^{A'_4 B'_4}(\theta) = \ell^{3-r} \varphi_r^{A_1 B_1}(\theta - \rho). \quad (17)$$

Obviously, we can reverse all the steps above and show that (9) implies (8).

Proposition 3 is demonstrated.

## APPENDIX F

### THE MATCHING ALGORITHM: PRACTICAL ISSUES

In this appendix, we take a closer look at the algorithm presented in Section 4.4.

#### F.1 Histogram Comparison (Step 3.2.1.1.2)

For the computation of matching degrees in Section 5, three similarity measures were tested:

$$\mu_T(h_1, h_2) = \frac{\sum_{\theta} \min(h_1(\theta), h_2(\theta))}{\sum_{\theta} \max(h_1(\theta), h_2(\theta))},$$

$$\mu_P(h_1, h_2) = 1 - \frac{\sum_{\theta} |h_1(\theta) - h_2(\theta)|}{\sum_{\theta} |h_1(\theta) + h_2(\theta)|},$$

$$\mu_C(h_1, h_2) = \frac{\sum_{\theta} h_1(\theta) h_2(\theta)}{\sqrt{\sum_{\theta} h_1^2(\theta)} \sqrt{\sum_{\theta} h_2^2(\theta)}}.$$

$\mu_T$  corresponds to a Tversky index (the Tanimoto index),  $\mu_P$  to a Pappis' measure, and  $\mu_C$  to normalized cross-correlation.  $h_1$  and  $h_2$  denote two force histograms.  $\theta$  belongs to the finite set of directions in which forces are considered. Although they represent three different types of measure,  $\mu_T$ ,  $\mu_P$ , and  $\mu_C$  all satisfy:

$$0 \leq \mu(h_1, h_2) \leq 1, \quad (18)$$

$$h_1 = h_2 \Rightarrow \mu(h_1, h_2) = 1, \quad (19)$$

$$\mu(h_1, h_2) = \mu(h_2, h_1), \quad (20)$$

$$\forall q \in \mathbb{R}_+^*, \mu(qh_1, qh_2) = \mu(h_1, h_2). \quad (21)$$

Equation (20) expresses the fact that the two histograms play equivalent parts. We want to assess the degree to which they are similar to each other. No pair of objects is privileged. The last equation, (21), expresses that the similarity between two histograms is invariant with respect to overall scale changes: Property 4 and (21) give  $\mu(\varphi_r^{\text{sca}(A'_4) \text{sca}(B'_4)}, \varphi_r^{\text{sca}(A_4) \text{sca}(B_4)}) = \mu(\varphi_r^{A'_4 B'_4}, \varphi_r^{A_4 B_4})$ . Note that  $\mu_C$  also satisfies:

$$\forall q_1 \in \mathbb{R}_+^*, \forall q_2 \in \mathbb{R}_+^*, \mu(q_1 h_1, q_2 h_2) = \mu(h_1, h_2). \quad (22)$$

Therefore, in the algorithm described in Section 4.4, the value of  $\ell$  is of no matter when using  $\mu_C$ .



We can get rid of the loop **For each**  $\ell$  and replace line 3.2.1 by, say,  $\ell \leftarrow 1$ ;

## F.2 Loops

In fact, it is better to compute  $\varphi_r^{A_1B_4}$  in two stages, and to use (14) and (15) (Appendix E) rather than (11). The reason is that, if  $k$  and  $k'$  are known, then  $\ell$  and  $\rho$  can be deduced from  $\varphi_r^{A_1B_4}$  in (10) and  $\varphi_r^{A_1B_1}$  in (14). In other words, we can get rid of the two inmost loops and replace lines 3.2.1 and 3.2.1.1 (**For each**  $\ell$  and **For each**  $\rho$ ) by statements like  $\ell \leftarrow \ell_{k,k'}$ ; and  $\rho \leftarrow \rho_{k,k'}$ ; . Let  $m'$  and  $c'$  be the mean and centroid<sup>7</sup> of  $\varphi_r^{A_1B_4}$ , and let  $m$  and  $c$  be the mean and centroid of  $\varphi_r^{A_1B_1}$ . The values  $m$  and  $m'$  represent forces on the Y-axis of the histograms, whereas  $c$  and  $c'$  represent angles on the X-axis. By equating  $\varphi_r^{A_1B_4}$  and  $\varphi_r^{A_1B_1}$  as defined in (15), we get  $\varphi_r^{A_1B_4}(\theta) = \ell^{3-r} \varphi_r^{A_1B_1}(\theta - \rho)$ , for any  $\theta$ . As a result,  $m' = \ell^{3-r}m$  and  $c' = c + \rho$ , i.e.,  $\ell = \ell_{k,k'}$  and  $\rho = \rho_{k,k'}$ , where  $\ell_{k,k'}$  denotes  $[m'/m]^{1/(3-r)}$  and  $\rho_{k,k'}$  denotes  $c' - c$ .

## ACKNOWLEDGMENTS

The authors want to express their gratitude for support from the US Office of Naval Research grant N00014-96-0439. They also wish to thank the referees for their constructive comments.

## REFERENCES

- [1] M.J. Swain and D.H. Ballard, "Indexing via Color Histograms," *Proc. Defense Advanced Research Projects Agency Conf.*, pp. 623-630, 1990.
- [2] R.M. Haralick, K. Shanmugam, and I. Dinstein, "Textural Features for Image Classification," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 3, no. 6, pp. 610-621, 1973.
- [3] M.-K. Hu, "Visual Pattern Recognition by Moment Invariants," *IRE Trans. Information Theory*, vol. 8, pp. 179-187, 1962.
- [4] *Proc. Invariants for Recognition, ESPRIT Workshop, Second European Conf. Computer Vision*, H. Burkhardt and A. Zisserman, eds., May 1992.
- [5] Y. Cheng, "Analysis of Affine Invariants as Approximate Perspective Invariants," *Computer Vision & Image Understanding*, vol. 63, no. 2, pp. 197-207, 1996.
- [6] F. Mindru, T. Moons, and L. Van Gool, "Recognizing Color Patterns Irrespective of Viewpoint and Illumination," *IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 368-373, June 1999.
- [7] F. Schaffalitzky and A. Zisserman, "Viewpoint Invariant Texture Matching and Wide Baseline Stereo," *Proc. IEEE Int'l Conf. Computer Vision*, vol. 2, pp. 636-643, 2001.
- [8] J. Flusser and T. Suk, "Pattern Recognition by Affine Moment Invariants," *Pattern Recognition*, vol. 26, no. 1, pp. 167-174, 1993.
- [9] T.H. Reiss, "The Revised Fundamental Theorem of Moment Invariants," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, no. 8, pp. 830-834, Aug. 1991.
- [10] D. Zhao and J. Chen, "Affine Curve Moment Invariants for Shape Recognition," *Pattern Recognition*, vol. 30, no. 6, pp. 895-901, 1997.
- [11] Z. Yang and F.S. Cohen, "Cross-Weighted Moments and Affine Invariants for Image Registration and Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 804-814, Aug. 1999.
- [12] K. Arbter, W.E. Snyder, H. Burkhardt, and G. Hirzinger, "Application of Affine-Invariant Fourier Descriptors to Recognition of 3D objects," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, pp. 640-647, 1990.
- [13] I. Rothe, H. Susse, and K. Voss, "The Method of Normalization to Determine Invariants," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 4, pp. 366-376, Apr. 1996.
- [14] F.S. Cohen, Z. Huang, and Z. Yang, "Invariant Matching and Identification of Curves Using B-Splines Curve Representation," *IEEE Trans. Image Processing*, vol. 4, no. 1, pp. 1-10, 1995.
- [15] F. Mokhtarian and S. Abbasi, "Shape Similarity Retrieval under Affine Transforms," *Pattern Recognition*, vol. 35, no. 1, pp. 31-41, 2002.
- [16] B. Li and S. De Ma, "On the Relation between Region and Contour Representation," *Proc. IAPR Int'l Conf. Pattern Recognition*, vol. 1, pp. 352-355, 1994.
- [17] S.-S. Wang, P.-C. Chen, and W.-G. Lin, "Invariant Pattern Recognition by Moment Fourier Descriptor," *Pattern Recognition*, vol. 27, no. 12, pp. 1735-1742, 1994.
- [18] T.S. Levitt and D.T. Lawton, "Qualitative Navigation for Mobile Robots," *Artificial Intelligence*, vol. 44, no. 3, pp. 305-360, 1990.
- [19] S. Dutta, "Approximate Spatial Reasoning: Integrating Qualitative and Quantitative Constraints," *Int'l J. Approximate Reasoning*, vol. 5, pp. 307-331, 1991.
- [20] S.Y. Lee and F.J. Hsu, "Spatial Reasoning and Similarity Retrieval of Images Using 2D C-String Knowledge Representation," *Pattern Recognition*, vol. 25, no. 3, pp. 305-318, 1992.
- [21] J. Sharma and D.M. Flewelling, "Inferences from Combined Knowledge about Topology and Directions," *Proc. Int'l Symp. Spatial Databases*, 1995.
- [22] R. Krishnapuram, J.M. Keller, and Y. Ma, "Quantitative Analysis of Properties and Spatial Relations of Fuzzy Image Regions," *IEEE Trans. Fuzzy Systems*, vol. 1, no. 3, pp. 222-233, 1993.
- [23] K. Miyajima and A. Ralescu, "Spatial Organization in 2D Segmented Images: Representation and Recognition of Primitive Spatial Relations," *Fuzzy Sets and Systems*, vol. 65, nos. 2/3, pp. 225-236, 1994.
- [24] P. Matsakis, "Relations Spatiales Structurelles et Interprétation d'Images," PhD thesis, Institut de Recherche en Informatique de Toulouse, France, 1998.
- [25] P. Matsakis and L. Wendling, "A New Way to Represent the Relative Position between Areal Objects," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 7, pp. 634-643, July 1999.
- [26] P. Matsakis, "Understanding the Spatial Organization of Image Regions by Means of Force Histograms: A Guided Tour," *Applying Soft Computing in Defining Spatial Relations*, P. Matsakis and L. Sztandera, eds., vol. 106, pp. 99-122, 2002.
- [27] P. Matsakis, J. Keller, L. Wendling, J. Marjamaa, and O. Sjahputera, "Linguistic Description of Relative Positions in Images," *IEEE Trans. Systems, Man, and Cybernetics (Part B)*, vol. 31, no. 4, pp. 573-588, 2001.
- [28] P. Matsakis and L. Wendling, "Orbit and Sinus Classification Based on Force Histogram Computation," *Proc. Int'l Conf. Pattern Recognition*, vol. 2, pp. 451-454, Sept. 2000.
- [29] M. Skubic, P. Matsakis, G. Chronis, and J. Keller, "Generating Multi-Level Linguistic Spatial Descriptions from Range Sensor Readings Using the Histogram of Forces," *Autonomous Robots*, vol. 14, no. 1, pp. 51-69, 2003.
- [30] C. Shyu and P. Matsakis, "Spatial Lesion Indexing for Medical Image Databases Using Force Histograms," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 603-608, Dec. 2001.
- [31] J.D. Cutnell and K. Johnson, *Physics*, fifth ed. Wiley and Sons, 2001.
- [32] S. Santini and R. Jain, "Similarity Measures," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 9, pp. 871-883, Sept. 1999.
- [33] X. Wang, B. De Baets, and E. Kerre, "A Comparative Study of Similarity Measures," *Fuzzy Sets and Systems*, vol. 73, no. 2, pp. 259-268, 1995.
- [34] C. Pappis and N. Karacapilidis, "A Comparative Assessment of Measures of Similarity of Fuzzy Values," *Fuzzy Sets and Systems*, vol. 56, no. 2, pp. 171-174, 1993.
- [35] P. Aschwanden and W. Guggenbühl, "Experimental Results from a Comparative Study on Correlation-Type Registration Algorithms," *Robust Computer Vision*, W. Forstner and S. Ruwiedel, eds., pp. 268-289, 1992.
- [36] M.A. Eshera and K.S. Fu, "An Image Understanding System Using Attributed Symbolic Representation and Inexact Graph-Matching," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, pp. 604-619, 1986.

7. Assuming the centroids can be defined (see the footnote in Section 4.1). Otherwise, the loop **For each**  $\rho$  cannot be avoided.

- [37] K.P. Chan and Y.S. Cheung, "Fuzzy-Attribute Graph with Application to Chinese Character Recognition," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 22, no. 1, pp. 153-160, 1992.
- [38] S. Medasani and R. Krishnapuram, "A Fuzzy Approach to Content-Based Image Retrieval," *Proc. IEEE Int'l Conf. Fuzzy Systems*, vol. 3, pp. 1251-1260, 1999.
- [39] G.M. Petrakis, "Design and Evaluation of Spatial Similarity Approaches for Image Retrieval," *Image & Vision Computing*, vol. 20, no. 1, pp. 59-76, 2002.



**Pascal Matsakis** received the PhD degree in computer science (1998), the BSc degree in mathematics, and the BSc degree in computer science from Paul Sabatier University, Toulouse, France. He has been an associate professor of computing and information science at the University of Guelph, Ontario, Canada, since August 2002. He was an assistant professor of computer engineering and computer science and member of the Computational

Intelligence Research Laboratory, at the University of Missouri-Columbia, from September 1999 to July 2002. Before moving to the United States, he worked in the Image Processing and Understanding Research Group at the Toulouse Institute of Research in Computer Science (IRIT). His interests include computer vision, computer graphics, pattern recognition, satellite image analysis, human-machine interaction, fuzzy set theory, and fuzzy logic. From a general point of view, Dr. Matsakis' research concerns the exploitation of expert knowledge and ancillary data for image interpretation and scene understanding. His particular field is in modeling and utilizing the spatial relations between multidimensional objects. He is a member of the IEEE and an associate editor of the *IEEE Transactions on Fuzzy Systems*.



**James M. Keller** received the PhD degree in mathematics in 1978. He has had faculty appointments in the Bioengineering/Advanced Automation Program, the Electrical and Computer Engineering Department, and the Computer Engineering and Computer Science Department at the University of Missouri-Columbia, where he currently is a professor. He is also the R.L. Tatum Research Professor in the College of Engineering. His research interests include

computer vision, pattern recognition, fuzzy set theory and fuzzy logic, fractal geometry, and neural networks. He has been funded by several industrial and government institutions, including the Electronics and Space Corporation, Union Electric, Geo-Centers, NASA/JSC, the US Air Force Office of Scientific Research, the US Army Research Office, the US Office of Naval Research, and the US Army Night Vision and Electronic Sensors Directorate. Dr. Keller has coauthored more than 175 technical publications. He is a fellow of the IEEE, for whom he has presented live and video tutorials on fuzzy logic in computer vision, is a national lecturer for the ACM, is an IEEE Neural Networks Council Distinguished Lecturer, and is a past president of the North American Fuzzy Information Processing Society (NAFIPS). He is the editor-in-chief of the *IEEE Transactions on Fuzzy Systems*, is an associate editor of the *International Journal of Approximate Reasoning*, and is on the editorial board of *Pattern Analysis and Applications*, *Fuzzy Sets and Systems*, the *International Journal of Fuzzy Systems*, and the *Journal of Intelligent and Fuzzy Systems*. He is currently serving a three-year term as an elected member of the IEEE SMC Society Administrative Committee. He was the conference chair of the 1991 NAFIPS Workshop, program cochair of the 1996 NAFIPS meeting, program cochair of the 1997 IEEE International Conference on Neural Networks, and the program chair of the 1998 IEEE International Conference on Fuzzy Systems. He was the general chair for the 2003 IEEE International Conference on Fuzzy Systems.



**Ozy Sjahputera** received the BSc degree in electrical engineering and computer engineering in 1994, and the MSc degree in electrical engineering in 1996, from University of Missouri-Columbia, where he is currently a PhD candidate in computer engineering and computer science. His research interests are in high-level computer vision, fuzzy logic applications, and spatial relations. He was with the Laboratory of Aero, Gas Dynamics and Vibration of the

Agency of Assessment and Application of Technology of the Republic Indonesia from 1996-1997. He then joined the Industrial Automation Research Group of the same agency from 1997-1998. He is a student member of the IEEE.



**Jonathon Marjamaa** received the BA degree in computer science in 1995 at St. Louis University, Missouri. In 2001, he earned the MSc degree at the University of Missouri-Columbia. He currently works for the Rotorcraft Training Systems Group at the Boeing Company in St. Louis, Missouri.

► For more information on this or any other computing topic, please visit our Digital Library at <http://computer.org/publications/dlib>.

## APPENDIX A — PROOF OF PROPOSITION 1

Since  $(A'_0, B'_0)$  matches  $(A_0, B_0)$  through  $(stre'_1, tran_1, rot_1, sca_1, stre_1)$  and through  $(stre'_2, tran_2, rot_2, sca_2, stre_2)$ , we have (Definition 1a):

$$A'_0 = stre'_1(tran_1(rot_1(sca_1(stre_1(A_0)))))) \text{ and } B'_0 = stre'_1(tran_1(rot_1(sca_1(stre_1(B_0)))))) \text{ and}$$

$$A'_0 = stre'_2(tran_2(rot_2(sca_2(stre_2(A_0)))))) \text{ and } B'_0 = stre'_2(tran_2(rot_2(sca_2(stre_2(B_0)))))).$$

Since  $\mathcal{O}$  is not ambiguous (Definition 2), the above equalities imply:

$$stre'_1 \circ tran_1 \circ rot_1 \circ sca_1 \circ stre_1 = stre'_2 \circ tran_2 \circ rot_2 \circ sca_2 \circ stre_2.$$

These two affine transformations can be represented by the following matrices, H and K:

$$H = \begin{pmatrix} 1 & 0 & 0 \\ 0 & k_1' & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & u_1'' \\ 0 & 1 & v_1'' \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\rho_1) & -\sin(\rho_1) & u_1' \\ \sin(\rho_1) & \cos(\rho_1) & v_1' \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \ell_1 & 0 & u_1 \\ 0 & \ell_1 & v_1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & k_1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \ell_1 \cos(\rho_1) & -\ell_1 k_1' \sin(\rho_1) & u_1 \cos(\rho_1) - v_1 \sin(\rho_1) + u_1' + u_1'' \\ \ell_1 k_1' \sin(\rho_1) & \ell_1 k_1' k_1 \cos(\rho_1) & k_1' u_1 \sin(\rho_1) + k_1' v_1 \cos(\rho_1) + k_1' v_1' + k_1' v_1'' \\ 0 & 0 & 1 \end{pmatrix}$$

$$K = \begin{pmatrix} \ell_2 \cos(\rho_2) & -\ell_2 k_2 \sin(\rho_2) & u_2 \cos(\rho_2) - v_2 \sin(\rho_2) + u_2' + u_2'' \\ \ell_2 k_2' \sin(\rho_2) & \ell_2 k_2' k_2 \cos(\rho_2) & k_2' u_2 \sin(\rho_2) + k_2' v_2 \cos(\rho_2) + k_2' v_2' + k_2' v_2'' \\ 0 & 0 & 1 \end{pmatrix}$$

(i), (ii) and (iii) are obtained by equating H to K. For instance, if  $\rho_1 \notin \{-\pi/2, 0, \pi/2, \pi\}$ ,  $H_{22}/H_{11} = K_{22}/K_{11}$  and  $H_{12}/H_{21} = K_{12}/K_{21}$  imply  $k_1 = k_2$  and  $k_1' = k_2'$ . Note that, in all cases,  $(u_1, v_1, u_1', v_1', u_1'', v_1'')$  and  $(u_2, v_2, u_2', v_2', u_2'', v_2'')$  are not necessarily equal. There are an infinite number of ways to choose *tran* and the rotation and scaling centers.

## APPENDIX B — SOME PROPERTIES OF FUNCTION $F_r$

$F_r$  denotes the F function (Fig. 7cd) associated with  $\phi_r$  (Equation (3)). Consider any element  $(\theta, I, J)$  of T, where T is defined as in Section III.A. Assume  $F_r$  is defined at  $(\theta, I, J)$ . The following properties hold.

The proofs, which can be found in [24], naturally rely on Equations (2) and (3).

*Property 6:*  $F_r$  is defined at  $(\theta, \text{tran}(I), \text{tran}(J))$  and:  $F_r(\theta, \text{tran}(I), \text{tran}(J)) = F_r(\theta, I, J)$ .

*Property 7:*  $F_r$  is defined at  $(\theta + \rho, \text{rot}(I), \text{rot}(J))$  and:  $F_r(\theta + \rho, \text{rot}(I), \text{rot}(J)) = F_r(\theta, I, J)$ .

*Property 8:*  $F_r$  is defined at  $(\theta, \text{sca}(I), \text{sca}(J))$  and:  $F_r(\theta, \text{sca}(I), \text{sca}(J)) = \ell^{2-r} F_r(\theta, I, J)$ .

### APPENDIX C — PROOF OF PROPERTY 5

*stre* is an affine transformation: it preserves collinearity and ratios of distances. In the following (refer to Fig. 20),  $A'$  and  $B'$  denote the transformed objects  $\text{stre}(A)$  and  $\text{stre}(B)$ .  $v$  is a given real,  $\alpha$  a given angle, and  $\beta$  is  $\alpha + \pi/2$ .  $U$  is the oriented line  $\Delta_\alpha(0)$ ,  $V$  is  $\Delta_\beta(0)$ ,  $U'$  is  $\text{stre}(U)$ , and  $V'$  is  $\text{stre}(V)$ .  $\alpha'$  denotes the angle such that  $U' = \Delta_{\alpha'}(0)$ , and  $\beta'$  the angle such that  $V' = \Delta_{\beta'}(0)$ .  $W'$  is the line  $\Delta_{\alpha' + \pi/2}(0)$ , and  $w'$  the real such that  $\text{stre}(\Delta_\alpha(v)) = \Delta_{\alpha'}(w')$ . This line  $\Delta_{\alpha'}(w')$  intersects  $V'$  in a point whose coordinate with respect to the reference frame  $(O, \vec{i}_{\beta'})$  is  $v'$ .

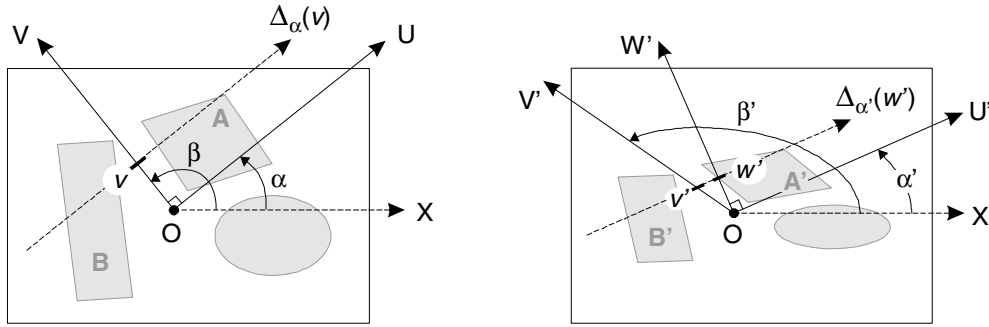


Fig. 20. Proof of Property 5. Notations.

*Lemma 1:* If  $\cos\alpha$  is zero,  $\alpha'$  is  $\alpha$ . Otherwise:  $\cos\alpha' \cos\alpha > 0$  and  $\sin\alpha' \sin\alpha \geq 0$  and  $\tan\alpha' = k \tan\alpha$ .

*Proof:* Let  $M(x, y)$  be the point of  $\Delta_\alpha(0)$  whose coordinate with respect to the reference frame  $(O, \vec{i}_\alpha)$  is 1, and let  $M'(x', y')$  be  $\text{stre}(M)$ . We have:  $\cos\alpha = x$ ,  $\sin\alpha = y$ ,  $\cos\alpha' = x' = x$  and  $\sin\alpha' = y' = ky$ . The rest is straightforward. □

*Lemma 2:*  $\cos\alpha' = k_1^{-1} \cos\alpha$  and  $\sin\alpha' = k k_1^{-1} \sin\alpha$ , where  $k_1$  denotes  $[k^2 + (1 - k^2) \cos^2\alpha]^{1/2}$ .

*Proof:* These identities come easily using Lemma 1 and the following formulas:

$$\forall x \in \mathbb{R}, \cos(\operatorname{atan}x) = [1+x^2]^{-1/2} \text{ and } \sin(\operatorname{atan}x) = x[1+x^2]^{-1/2}. \quad \square$$

*Lemma 3:* Assume:  $M \in \Delta_\alpha(v)$ ,  $N \in \Delta_\alpha(v)$ ,  $M' = \operatorname{stre}(M)$  and  $N' = \operatorname{stre}(N)$ . Then:  $\overrightarrow{M'N'} \cdot \vec{i}_{\alpha'} = k_1 \overrightarrow{MN} \cdot \vec{i}_\alpha$ .

*Proof:* We assume that  $(x_1, y_1)$  and  $(x_2, y_2)$  are the coordinates of  $M$  and  $N$ . Note that  $M'(x_1, ky_1)$  and  $N'(x_2, ky_2)$  belong to  $\Delta_{\alpha'}(w')$ . If  $x_2$  is not equal to  $x_1$ , we have:  $\overrightarrow{MN} \cdot \vec{i}_\alpha = (x_2 - x_1)\cos\alpha + (y_2 - y_1)\sin\alpha = (x_2 - x_1)[\cos\alpha + \tan\alpha \sin\alpha] = (x_2 - x_1)/\cos\alpha$ . In the same way:  $\overrightarrow{M'N'} \cdot \vec{i}_{\alpha'} = (x_2 - x_1)/\cos\alpha'$ . Using Lemma 2, we obtain:  $\overrightarrow{M'N'} \cdot \vec{i}_{\alpha'} = k_1 \overrightarrow{MN} \cdot \vec{i}_\alpha$ . It is easy to check that the identity holds when  $x_2$  is equal to  $x_1$ .  $\square$

*Lemma 4:*  $\cos\beta' = -k_2^{-1}\sin\alpha$  and  $\sin\beta' = k_2^{-1}\cos\alpha$ , where  $k_2$  denotes  $[1+(k^2-1)\cos^2\alpha]^{1/2}$ .

*Proof:* Lemma 2 obviously holds when replacing the pair  $(\alpha, \alpha')$  by  $(\beta, \beta')$ . We also use the fact that  $\beta$  is  $\alpha + \pi/2$  (hence,  $\cos\beta = -\sin\alpha$ , and  $\cos^2\beta = 1 - \sin^2\beta = 1 - \cos^2\alpha$ ).  $\square$

*Lemma 5:*  $v' = k_2 v$ .

*Proof:* Let  $M$  and  $N$  be two points of  $\Delta_\beta(0)$  (i.e., two point of  $V$ ), and let  $M'$  and  $N'$  be  $\operatorname{stre}(M)$  and  $\operatorname{stre}(N)$ . We can show, exactly as we showed Lemma 3, that:  $\overrightarrow{M'N'} \cdot \vec{i}_{\beta'} = k_2 \overrightarrow{MN} \cdot \vec{i}_\beta$ . We obtain  $v' = k_2 v$  by choosing  $M$  equal to the origin  $O$  and by choosing  $N$  equal to the point of  $V$  whose coordinate is  $v$  (with respect to the reference frame  $(O, \vec{i}_\beta)$ ).  $\square$

*Lemma 6:*  $w' = k[k_1 k_2]^{-1} v' = k k_1^{-1} v$ .

*Proof:* We have:  $w' = v' \cos(\beta' - (\alpha' + \pi/2)) = v' \sin(\beta' - \alpha') = v' [\sin\beta' \cos\alpha' - \sin\alpha' \cos\beta']$ .

We then use Lemmas 2, 4 and 5.  $\square$

*Lemma 7:* There exist a translation  $\operatorname{tran}$ , a scaling  $\operatorname{sca}$ , and a rotation  $\operatorname{rot}$  such that:  $A'_{\alpha'}(w') = (\operatorname{tran} \circ \operatorname{sca} \circ \operatorname{rot})(A_\alpha(v))$  and  $B'_{\alpha'}(w') = (\operatorname{tran} \circ \operatorname{sca} \circ \operatorname{rot})(B_\alpha(v))$ . The scaling ratio is  $k_1$  and the rotation angle is  $\alpha' - \alpha$ .

*Proof:*  $\operatorname{stre}$  is an affine transformation that transforms line  $\Delta_\alpha(v)$  into line  $\Delta_{\alpha'}(w')$ . Therefore, there exist a translation  $\operatorname{tran}$ , a scaling  $\operatorname{sca}$ , and a rotation  $\operatorname{rot}$  such that:  $\forall M \in \Delta_\alpha(v)$ ,  $\operatorname{stre}(M) = (\operatorname{tran} \circ \operatorname{sca} \circ \operatorname{rot})(M)$ . The scaling ratio is  $k_1$  because of Lemma 3, and the rotation angle is  $\alpha' - \alpha$  because  $\alpha' - \alpha$  is the angle between the oriented line  $\Delta_\alpha(v)$  and its image  $\Delta_{\alpha'}(w')$ . Hence:  $\operatorname{stre}(A_\alpha(v)) = (\operatorname{tran} \circ \operatorname{sca} \circ \operatorname{rot})(A_\alpha(v))$ . But we also have:

$$\operatorname{stre}(A_\alpha(v)) = \operatorname{stre}(\Delta_\alpha(v) \cap A) = \operatorname{stre}(\Delta_\alpha(v)) \cap \operatorname{stre}(A) = \Delta_{\alpha'}(w') \cap A' = A'_{\alpha'}(w').$$

Finally:  $A'_{\alpha'}(w') = (\operatorname{tran} \circ \operatorname{sca} \circ \operatorname{rot})(A_\alpha(v))$ . In the same way:  $B'_{\alpha'}(w') = (\operatorname{tran} \circ \operatorname{sca} \circ \operatorname{rot})(B_\alpha(v))$ .  $\square$

We can now proceed to the body of the proof. Using successively Lemma 7 and Properties 7, 8 and 6 (Appendix B), we can state that  $F_r$  is defined at  $(\alpha', A'_{\alpha'}(w'), B'_{\alpha'}(w'))$ . Moreover:

$$\begin{aligned} F_r(\alpha', A'_{\alpha'}(w'), B'_{\alpha'}(w')) &= F_r(\alpha', (tran \circ sca \circ rot)(A_{\alpha}(v)), (tran \circ sca \circ rot)(B_{\alpha}(v))) \\ &= F_r(\alpha', (sca \circ rot)(A_{\alpha}(v)), (sca \circ rot)(B_{\alpha}(v))) = k_1^{2-r} \times F_r(\alpha + (\alpha' - \alpha), rot(A_{\alpha}(v)), rot(B_{\alpha}(v))) \\ &= k_1^{2-r} \times F_r(\alpha, A_{\alpha}(v), B_{\alpha}(v)) \end{aligned}$$

This result, Lemma 6 and Equation (1) allow us to write:

$$\begin{aligned} \varphi_r^{A'B'}(\alpha') &= \int_{-\infty}^{+\infty} F_r(\alpha', A'_{\alpha'}(w'), B'_{\alpha'}(w')) dw' = \int_{-\infty}^{+\infty} k_1^{2-r} \times F_r(\alpha, A_{\alpha}(v), B_{\alpha}(v)) (k k_1^{-1} dv) \\ &= k k_1^{1-r} \int_{-\infty}^{+\infty} F_r(\alpha, A_{\alpha}(v), B_{\alpha}(v)) dv = k [k^2 + (1-k^2) \cos^2 \alpha]^{(1-r)/2} \varphi_r^{AB}(\alpha). \end{aligned}$$

Lemmas 1 and 2 obviously hold when replacing the 3-tuplet  $(\alpha, \alpha', k)$  by  $(\alpha', \alpha, k^{-1})$ . In other words, we can substitute  $[k^{-2} + (1-k^{-2}) \cos^2 \alpha']^{-1/2} \cos \alpha'$  for  $\cos \alpha$  in the last expression of  $\varphi_r^{A'B'}(\alpha')$ . We obtain:

$$\varphi_r^{A'B'}(\alpha') = k^{2-r} [1 + (k^2 - 1) \cos^2 \alpha']^{(r-1)/2} \varphi_r^{AB}(\alpha).$$

The trigonometric function *atan* is a one-to-one map from  $\mathbb{R}$  onto  $]-\pi/2, \pi/2[$ . According to Lemma 1, the angle  $\alpha$  is *atan*( $k^{-1} \tan \alpha'$ ) if  $\cos \alpha'$  is positive, is  $\alpha'$  if  $\cos \alpha'$  is zero, and is *atan*( $k^{-1} \tan \alpha'$ ) +  $\pi$  otherwise. Now, replace  $A'$  by *stre*( $A$ ),  $B'$  by *stre*( $B$ ),  $\alpha'$  by  $\theta$ , and  $\alpha$  by  $\theta_{[k]}$ : Property 5 is demonstrated.