Force Histograms Computed in $O(NlogN)$

Jingbo Ni
University of Guelph, Canada
jni@uoguelph.ca

Pascal Matsakis
University of Guelph, Canada
matsakis@cis.uoguelph.ca

Abstract

The relative position between two objects in a 2D raster image is often represented quantitatively by a force histogram. In the general case, force histograms are computed in $O(KN\sqrt{N})$ time: N is the number of pixels in the image and K is the number of directions in which forces are considered. When the objects are defined as fuzzy sets, this complexity also depends quadratically on the number M of possible membership degrees. In the present paper, an algorithm that runs in $O(NlogN)$ is introduced. Computation times are basically independent of K and M. All objects (convex, concave, crisp, fuzzy) are handled in an equally fast manner. Experiments validate the theoretical analysis.

1. Introduction

Motivations for automated spatial representations come from practical applications like geographic information systems [1], robot navigation [2], and content-based image retrieval [3].

Spatial representations are either qualitative or quantitative [4]. Qualitative representations focus on uniqueness and essentialness of a feature, whereas quantitative representations use numerical values to express the degrees of significances of a feature. In addition, quantitative representations require considering objects' visual features, like shape and size, whereas qualitative ones treat spatial knowledge as a concept independent of visual knowledge [5].

Numerous approaches have been proposed to quantitatively represent the relative position between two spatial objects. We focus here on two influential approaches within the field. The angle histogram [6][7] is appealing for its simplicity, but suffers from weaknesses including long processing times, anisotropy, and an inability to handle vector data. The F-histogram, introduced by Matsakis in [8], is a generic and more complex representation. Most related work has been done on particular F-histograms called force histograms. The force histogram generalizes and overcomes the weaknesses of the angle histogram [9]. It has been used in scene description, in human-robot communication, for the classification of cranium sinuses, the design of spatial indexing mechanisms for medical image databases, etc. See [10] for a review of work on and applications of the force histogram. See also [11], Section 2.2, for general information on the different types of F-histograms.

The spatial objects considered here are objects (or regions) in a 2D raster image. Angle histograms are then computed in $O(N^2)$ time, where N is the number of pixels in the image [6][7]. For convex objects, force histograms are computed in $O(KN)$, where K is the number of directions in which forces are taken account of [9][10]. Some histograms (constant force histograms, which are fundamentally equivalent to angle histograms) can be computed in $O(KN)$ or $O(NlogN)$, regardless of whether the objects are convex or not [12]. In the general case, however, force histograms are computed in $O(KN\sqrt{N})$ time [9][10]. When fuzzy objects are considered, this complexity also depends quadratically on the number M of possible membership degrees. In the present paper, a new algorithm is introduced. Computation times are basically independent of K and M. The algorithm is in $O(NlogN)$ for all force histograms, whether the objects are convex or concave, crisp or fuzzy. The histograms are derived from a mapping defined over the 2D discrete vector space. The mapping provides raw information about the objects' relative position and can be computed efficiently using the Fast Fourier Transform (FFT).

The angle and force histograms are briefly reviewed in Section 2. The new algorithm is presented in Section 3. An experimental study, in Section 4, validates the theoretical analysis. Conclusions are given in Section 5.

2. Review

2.1. Angle histogram

Let A and B be two objects defined as finite sets of pixels: $A=\{a_1,a_2,\ldots,a_m\}$ and $B=\{b_1,b_2,\ldots,b_n\}$. For any $a_i \in A$ and $b_j \in B$, let $\angle a_i b_j \in [0,2\pi)$ be the direction of the oriented line that runs from the centre of a_i to the centre of b_j. The *histogram of angles* H^{AB} is a function from $[0,2\pi)$ to $\Re$ (the set of real numbers). It is a possible representation of the position of B relative to A. The value $H^{AB}(\theta)$ is defined as the number of pixel pairs (a_i,b_j) such that $\angle a_i b_j = \theta$. Note that only $m \times n$ directions θ may satisfy $H^{AB}(\theta) \neq 0$. For fuzzy objects A and B with membership functions μ_A and μ_B, the value $H^{AB}(\theta)$ is usually set to $\sum_{\angle a_i b_j = \theta} \mu_A(a_i)\mu_B(b_j)$.

2.2. Force histograms

Here, A and B are true 2D objects. Each one is an infinite set of points, with a measurable, nonzero area. Consider any $p \in A$ and $q \in B$. Let $|pq|$ be the distance between p and q, and let $\angle pq$ be the direction of the oriented line that runs from p to q. The points p and q are seen as particles that attract each other: q exerts on p an elementary force whose direction is $\angle pq$ and whose magnitude is $1/|pq|^t$, where $t \in \Re$ is a constant. The *histogram of forces* F_t^{AB} is a function from $[0,2\pi)$ to $\Re$. It is another possible representation of the position of B relative to A. The value $F_t^{AB}(\theta)$ is defined as the sum of the magnitudes of all the elementary forces in direction θ. For fuzzy objects A and B with membership functions μ_A and μ_B, the magnitude of the elementary force exerted by q on p is usually set to $\mu_A(p)\mu_B(q)/|pq|^t$. When $t=2$, this expression matches Newton's law of gravity, and F_2^{AB} is a *gravitational force histogram*. As another example, F_0^{AB} is a *constant force histogram*. In practice, of course, a force histogram F_t^{AB} is represented by a limited number of $F_t^{AB}(\theta)$ values. The computation of each $F_t^{AB}(\theta)$ translates into the assessment of algebraic expressions, which are predetermined through integral calculus. In the case of raster data, each assessment corresponds to the process of a pair of object segments (more precisely, a batch of pairs of object pixels). The generation of these segments is based on the rasterization of lines using Bresenham's algorithm. The manipulation of fuzzy objects is reduced to that of their level cuts using the double sum scheme [13]. Details in [8][9].

3. Algorithm

A new algorithm for force histogram computation is introduced. The algorithm is dedicated to the handling of 2D raster data. As shown in Section 3.2, a force histogram can be derived from a mapping defined over the 2D discrete vector space. The mapping, presented in Section 3.1, is a mathematical correlation, and can therefore be computed using the Fast Fourier Transform (FFT).

3.1. Spatial correlation

Consider two pixels p and q, with coordinates (x_p,y_p) and (x_q,y_q). The position of q with respect to p can be represented by the vector $pq=(x_q-x_p,y_q-y_p)$, of length $|pq|$ and direction $\angle pq$. Now, consider two objects A and B (possibly fuzzy) in a digital image of size $N=m \times n$. Let μ_A and μ_B be the membership functions of A and B. Through zero padding, the image can be expanded to occupy the infinite space $\mathcal{Z}^2$, where $\mathcal{Z}$ denotes the set of all integers. The mapping Ψ^{AB} from $\mathcal{Z}^2$ to $\Re$ defined as follows:

$$\Psi^{AB}(u,v) = \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} \mu_A(x,y)\, \mu_B(x+u, y+v) \quad (1)$$

is a mathematical correlation that provides raw information about A and B's relative position. We call it the *spatial correlation between the objects* A and B. When A and B are crisp, $\Psi^{AB}(u,v)$ counts the number of pixel pairs (p,q), with $p \in A$ and $q \in B$, such that $pq=(u,v)$. See Figure 1. The set $\Delta=\{-m+1,\ldots,m-1\} \times \{-n+1,\ldots,n-1\}$ is the *effective domain* of Ψ^{AB}. Since the image is expanded through zero padding, only elements (u,v) of Δ may satisfy $\Psi^{AB}(u,v) \neq 0$. There are (about) $4N$ elements in Δ. Since each $\Psi^{AB}(u,v)$ is computed in $O(N)$ time, Ψ^{AB} is computed in $O(4N^2)$ time. Ψ^{AB}, however, can be computed in a much faster manner. Let A' denote the reflection of A about the origin: $\mu_A(-x,-y)=\mu_A(x,y)$. By replacing $\mu_A(x,y)$ in (1) with $\mu_A(x',y')$, where $x'=-x$ and $y'=-y$, we get:

$$\Psi^{AB}(u,v) = \sum_{x'=-m+1}^{0} \sum_{y'=-n+1}^{0} \mu_{A'}(x',y')\ \mu_B(u-x', v-y') \quad (2)$$

Equation (2) corresponds to the definition of discrete convolution. As we all know, the Fourier transform has the ability to convert a convolution into an ordinary product, and vice versa. Therefore, by taking advantage of the Fast Fourier Transform (FFT), the spatial correlation Ψ^{AB} can be computed in $O(N \log N)$ time.

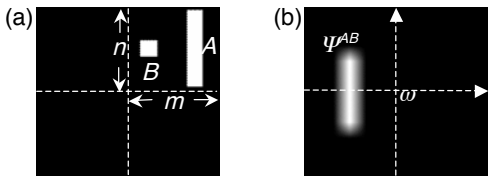


Figure 1. Spatial correlation Ψ^{AB}

3.2. Histogram generation

Let t be any real number, and let A and B be two objects in an image of size $N=m\times n$. The spatial correlation between A and B is Ψ^{AB}, with effective domain Δ. The origin, i.e., pixel at $(0,0)$, is ω. Consider the function $\hat{F}_t^{AB}$ from $[0,2\pi)$ to $\Re$ defined by:

$$\hat{F}_t^{AB}(\theta) = \sum_{p\in\Delta} \Psi^{AB}(p)\, f_t(p,\theta) \qquad (3)$$

where $f_t(p,\theta)=1/|\omega p|^t$ if $\angle\omega p=\theta$ and $f_t(p,\theta)=0$ otherwise. When $t=0$, we have $f_0(p,\theta)=1$ if $\angle\omega p=\theta$ and $f_0(p,\theta)=0$ otherwise. It is not hard to see that $\hat{F}_0^{AB}=H^{AB}$. We know that the histogram of constant forces F_0^{AB} and the histogram of angles H^{AB} are two different functions. For example, only a finite number of directions θ satisfy $H^{AB}(\theta)\neq 0$, whereas an infinite number of directions satisfy $F_0^{AB}(\theta)\neq 0$. We also know, however, that F_0^{AB} and H^{AB} are fundamentally equivalent [9]. One can say, therefore, that $\hat{F}_t^{AB}$ is to F_t^{AB} what H^{AB} (i.e., $\hat{F}_0^{AB}$) is to F_0^{AB}. To let $\hat{F}_t^{AB}$ better correspond to the histogram of forces F_t^{AB}, we need to modify the way $f_t(p,\theta)$ is defined. In the effective domain Δ, using some line-drawing algorithm such as Bresenham's, let us draw a straight line Δ_θ that starts from the origin ω and runs in direction θ. We redefine $f_t(p,\theta)$ as follows: $f_t(p,\theta)=1/|\omega p|^t$ if $p\in\Delta_\theta-\{\omega\}$ and $f_t(p,\theta)=0$ otherwise. Equation (3) can then be rewritten as:

$$\hat{F}_t^{AB}(\theta) = \sum_{p\in\Delta_\theta-\{\omega\}} \Psi^{AB}(p) \,/\, |\omega p|^t \qquad (4)$$

Equation (4) brings $\hat{F}_t^{AB}$ closer (although not exactly equal) to F_t^{AB}: an infinite number of directions θ now satisfy $\hat{F}_t^{AB}(\theta)\neq 0$. The $\hat{F}_t^{AB}$ histograms are computed in $O(N\log N+K\sqrt{N})$ time, where K is the number of directions θ in which forces are considered. $O(N\log N)$ is required for generating Ψ^{AB} (see Section 3.1), while $O(K\sqrt{N})$ is required by Equation (4). As illustrated in Figure 2(a), given an image of size $N=n\times n$, there are n different directions within $[0,\pi/4]$ for lines going from the bottom-left pixel to other boundary pixels. In $[0,2\pi)$, there are $8n-8$ (i.e., $8\sqrt{N}-8$) such directions. This shows that K might be negligible compared to $\sqrt{N}$ ($K<<\sqrt{N}$), or might be of the same order ($K\approx\sqrt{N}$), but there is no interest in choosing greater values ($K>>\sqrt{N}$). The presented algorithm, therefore, runs in $O(N\log N+\sqrt{N}\sqrt{N}) = O(N\log N)$. Computation times are basically independent of t (type of force histograms), K (number of directions in which forces are computed), and M (number of possible membership degrees). All objects (convex, concave, of simple or complex shapes, defined as crisp or fuzzy sets) are handled in an equally fast manner. The experiments in Section 4 validate this theoretical analysis.

4. Experiments

In this section, F_t^{AB} denotes a force histogram computed using the standard algorithm (see Section 2.2) and $\hat{F}_t^{AB}$ denotes a histogram computed using the new algorithm (see Section 3). For a given K, the $\theta_i=2\pi i/K$ with $i\in 0..K-1$ are the directions in which forces are considered. The test data include three pairs of crisp objects, as depicted in Figure 2(b,c,d). Moreover, each pair (A,B) was used to generate several pairs of fuzzy objects. For a given M, the pixels in A were assigned different membership degrees, randomly selected from $\{i/M\}_{i\in 1..M}$. The object B was fuzzified in the same way. All algorithms were implemented in C and run on a machine equipped with Intel Pentium D CPU 3.00 GHz and 1GB memory.

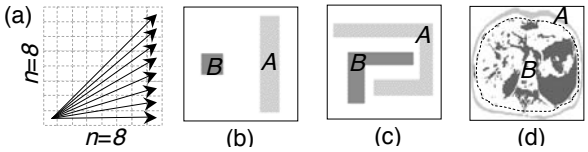


Figure 2. (a) About the number of directions in an image; (b,c,d) Test object pairs

4.1. Comparing histogram values

This series of experiments illustrates that F_t^{AB} and $\hat{F}_t^{AB}$ are practically identical. The dissimilarity between F_t^{AB} and $\hat{F}_t^{AB}$ is measured by the average difference ratio (ADR):

$$ADR = \left(\sum_{i=0}^{K-1} \frac{\left|\hat{F}_t^{AB}(\theta_i) - F_t^{AB}(\theta_i)\right|}{max\left(\hat{F}_t^{AB}(\theta_i), F_t^{AB}(\theta_i)\right)} \right) \times \frac{1}{K} \times 100\% \qquad (5)$$

The ADR values tend to be very low (see Table 1). They do not depend much on K and M (results not shown here), and they drop and tend towards 0 when the size N of the image increases (see Figure 3).

Table 1. *ADR*
$N=128^2$, $K=1800$

Figure 2		(b)	(c)	(d)
Crisp	$t=0$	0.4%	0.5%	0.2%
($M=1$)	$t=2$	0.4%	0.8%	1.1%
Fuzzy	$t=0$	0.5%	0.7%	0.3%
($M=5$)	$t=2$	0.5%	0.9%	1.2%

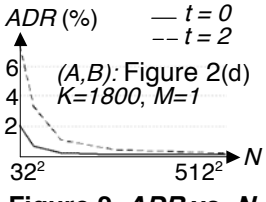
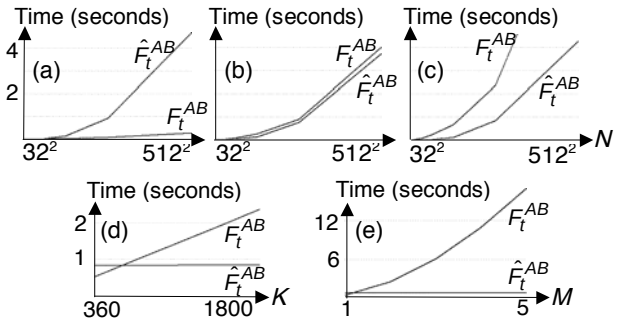


Figure 2. *ADR* vs. *N*

4.2. Comparing processing times

The results shown in this section are identical for all types of force histograms (processing times are basically independent of t). If we assume only convex objects are allowed, F_t^{AB} is computed in $O(KN)$ time.

However, the efficiency decreases for objects with more complex shapes. In the worst case, F_t^{AB} is computed in $O(KN\sqrt{N})$ time, whereas $\hat{F}_t^{AB}$ is always computed in $O(NlogN)$. This is illustrated by Figure 4. For the object pair depicted in Figure 2(b), when N increases, the time required to compute F_t^{AB} grows slowly in comparison to the computation time of $\hat{F}_t^{AB}$ (Figure 4(a)). For the pair in Figure 2(c), the performance of F_t^{AB} decreases rapidly, while that of $\hat{F}_t^{AB}$ remains constant, which, therefore, makes them comparable (Figure 4(b)). Finally, for the object pair in Figure 2(d), $\hat{F}_t^{AB}$ outperforms F_t^{AB} (Figure 4(c)). Figure 3 also shows that K and M have almost no influence on the computational efficiency of $\hat{F}_t^{AB}$. On the other hand, the complexity of F_t^{AB} depends linearly on K (Figure 4(d)) and quadratically on M (Figure 4(e)).



In (a,b,c), $K=1800$, $M=1$. In (d), $M=1$, $N=256^2$. In (e), $K=360$, $N=256^2$. The objects A and B are from Figure 2(b,c,d,d,d), respectively.

Figure 3. Processing times

5. Conclusions

The standard algorithm for force histogram computation in the case of 2D raster data runs in $O(KM^2N\sqrt{N})$ time, where K is the number of directions in which forces are considered, M is the number of possible membership degrees, and N is the number of pixels in the image. In the present paper, we have described an algorithm that runs in $O(NlogN)$. For crisp objects ($M=1$) with simple shapes, or if only a relatively small number K of histogram values are needed, the standard algorithm performs well. In all other cases, however, the new algorithm is much more efficient. The computed histograms are derived from a mapping that provides raw information about the objects' relative position. From a theoretical point of view, they are not true force histograms. However, they are practically identical. In future work, we plan to further refine the algorithm and achieve absolute identity.

Acknowledgements

The authors want to express their gratitude for support from the Natural Science and Engineering Research Council of Canada (NSERC), grant 262117.

References

[1] H.J. Miller, and E.A. Wentz. Representation and Spatial Analysis in Geographic Information Systems. *In: Annals of the Association of American Geographers*, 93(3):574–94, 2003.

[2] J.D.R. Millan. Robot Navigation. *In: Handbook of Brain Theory and Neural Networks*, Second Edition, Cambridge, MA: MIT Press (Arbib, M.A., Ed.), 2002.

[3] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based Image Retrieval at the End of the Early Years. *In: IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(12):1349–80, 2000.

[4] D. Hernandez. Qualitative Representation of Spatial Knowledge. *Lecture Notes in Artificial Intelligence 804*, Springer-Verlag, 1994.

[5] D. Papadias. Relation-based Representations for Spatial Knowledge. *PhD. Thesis*, National Technical University of Athens, 1994.

[6] R. Krishnapuram, J. Keller, and Y. Ma. Quantitative Analysis of Properties and Spatial Relations of Fuzzy Image Regions. *In: IEEE Transactions on Fuzzy Systems*, 1(3):222–33, 1993.

[7] K. Miyajima, and A. Ralescu. Spatial Organization in 2D Segmented Images: Representation and Recognition of Primitive Spatial Relations. *In: Fuzzy Sets and Systems*, 65(2-3):225–36, 1994.

[8] P. Matsakis. Relations spatiales structurelles et interprétation d'images, *PhD. Thesis*, Institut de Recherche en Informatique de Toulouse, France, 1998.

[9] P. Matsakis, and L. Wendling. A New Way to Represent the Relative Position of Areal Objects. *In: IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(7):634–43, 1999.

[10] P. Matsakis. Understanding the Spatial Organization of Image Regions by Means of Force Histograms: A Guided Tour. *In: Applying Soft Computing in Defining Spatial Relations*, Springer-Verlag Publications (Matsakis, P., and Sztandera, L., Eds.):99–122, 2002.

[11] P. Matsakis, and D. Nikitenko. Combined Extraction of Directional and Topological Relationship Information from 2D Concave Objects. *In: Fuzzy Modeling with Spatial Information for Geographic Problems*, Springer-Verlag Publications (Cobb, M., Petry, F., and Robinson, V., Eds.):15–40, 2005.

[12] Y. Wang, F. Makedon, and R.L. (Scot) Drysdale. Fast Algorithms to Compute the Force Histogram. *Accepted by Pattern Recognition*, 2004.

[13] D. Dubois, and M.C. Jaulent. A General Approach to Parameter Evaluation in Fuzzy Digital Pictures. *In: Pattern Recognition Letters*, 6(4):251–9, 1987.