Fast Fourier Transform based Force Histogram Computation for 3D Raster Data

Jaspinder Kaur, Tyler Laforet and Pascal Matsakis
Department of Computer Science, University of Guelph, Guelph, Canada

Keywords: Relative Position Descriptor, Computer Vision, Image Processing, Force Histogram.

Abstract: The force histogram is a quantitative representation of the relative position between two objects. Two practical algorithms have been previously introduced to compute the force histogram between objects: the line-based algorithm (which works well with 2D data, but is computationally unstable in the case of 3D data), and the Fast Fourier Transform (FFT)-based algorithm (which is inefficient in the case of 2D data, but has not been implemented for 3D data). In this paper, an efficient FFT-based algorithm for force histogram computation in the case of 3D raster data is introduced. Its computation time is compared against that of the 3D line-based algorithm; except in a few cases, the computation time for new FFT-based algorithm is less than that of 3D line-based algorithm. The experiments validate that the FFT-based algorithm is computationally efficient regardless of the number of directions, type of forces, and shape of the objects (convex, concave, disjoint or overlapping).

1 INTRODUCTION

Humans often rely on the relative positioning of the objects around them in order to understand and express spatial information. In our daily lives, the relative position of objects is often communicated through linguistic expressions like, “the shopping mall is south of the apartment”, “the school is near my home”, and “the folder is inside the case”. Spatial prepositions like “north”, “inside” and “nearby” represent spatial relationships. Distance relationships (Ryoo and Aggarwal, 2009) describe how far apart the objects are in space, like “far away”, “nearby”, “at”. Directional relationships (also called cardinal or projective (Gapp, 1995) relationships) are mostly identified by words like “west”, “to the right of”. Topological relationships specify concepts of connectivity, adjacency and enclosure (Egenhofer, 1989; Schneider and Behr, 2005; Egenhofer, 1990).

Comprehending the spatial relationships in a scene plays a vital role in several areas such as pattern recognition, human-robot communication, scene description in natural language, image understanding and so forth. Models of spatial relationships (distance, directional and topological) among spatial entities are either quantitative or qualitative. In qualitative models (Frank, 1996), a relationship either holds or does not hold. On the other hand, in quantitative models, a relationship may hold to some degree (Deselaers et al., 2004). Over the years, qualitative models have been used in various areas of computer vision. However, most practical applications of computer vision and image processing (Smeulders et al., 2000; Skubic et al., 2001; Skubic et al., 2002; Miller and Wentz, 2003) require quantitative models called relative position descriptors (RPD).

Relative position descriptors are vectors of values that encode information about an object’s position relative to another object. RPDs bridge the gap between low-level spatial features (e.g., pixels in an image) and high level concepts (i.e spatial relationships). An ideal relative position descriptor encapsulates all possible spatial relationship information between objects, and allows that information to be extracted efficiently. Over the years, much attention has been paid towards identifying effective approaches for modelling objects’ relative positions. Several RPDs have been introduced (Miyajima and Ralescu, 1994; Krishnapuram et al., 1993; Naeem and Matsakis, 2015). The force histogram (Matsakis, 1998; Matsakis and Wendling, 1999) may be one of the most well known relative position descriptors with various theoretical and practical applications. It encapsulates the structural information (e.g., size, shape, etc.) as well as the spatial relationship information between objects. So far, many algorithms for force histogram compu-

Kaur, J., Laforet, T. and Matsakis, P.
Fast Fourier Transform based Force Histogram Computation for 3D Raster Data.
DOI: 10.5220/0008985100690074
In *Proceedings of the 9th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2020)*, pages 69-74
ISBN: 978-989-758-397-1; ISSN: 2184-4313

tation have been introduced. The two main algorithms for force histogram computation are the line-based algorithm and the Fast Fourier Transform (FFT)-based algorithm. The line-based algorithm for force histogram computation exists for 2D raster objects, 2D vector objects, 3D raster objects and 3D vector objects, although the algorithms for the 3D objects are computationally costly. On the other hand, the FFT-based algorithm exists only for 2D raster data, and has not yet been implemented for 3D raster objects.

This paper presents an extension of the FFT-based algorithm for the computation of force histogram in the case of 3D raster objects. In Section 2, we discuss the force histogram and provide a brief overview of various algorithms found in the literature for computing the force histogram. In Section 3, the algorithm which involves handling 3D raster objects is presented. Section 4 presents the experiments and a comparative study in which the line-based force histogram is compared against the FFT-based force histogram. Section 5 concludes the discussion and postulates future work.

2 BACKGROUND

2.1 Force Histogram

A histogram of forces $F_t^{AB}(\theta)$ is a function from $\mathbb{R}$ (the set of directions) to $\mathbb{R}_+$ (the set of real numbers) that represent the position of object A relative to object B (Figure 1b). For a given direction θ, the value $F_t^{AB}(\theta)$ is the sum of the magnitude of all elementary forces exerted by the points of B on those of A; this in turn tends to move A in the direction θ (Figure 1a). Here, the object A is called the argument and the object B is called the referent. The magnitude of each elementary force is given by $1/|r|^t$, where r is the distance between points and t is any real number. The function $F_t^{AB}(\theta)$ has two special cases: $t = 2$, and $t = 0$. When $t = 2$, $F_2^{AB}(\theta)$ is called the gravitational force histogram, and the forces between the points are inversely proportional to the the square of distance between them $(1/r^2)$. When $t = 0$, $F_0^{AB}(\theta)$ is called the constant force histogram, and the forces do not depend on this distance.

2.2 Algorithms for the Force Histogram Computation

2.2.1 Line-based Algorithm

Matsakis indicates in (Matsakis, 1998; Matsakis and Wendling, 1999) that the force histogram between objects in an image can be computed by first partitioning the image into a number of raster lines, and then batch-processing pixels on each line. Consider objects A and B; for any oriented line Λ, $F_t^{AB}(\theta)$ is the scalar resultant of elementary forces exerted by points of $B \cap \Lambda$ on those of $A \cap \Lambda$ in a direction θ. In an ideal case, the shapes of objects A and B are simple (e.g., convex). For any direction θ, the line Λ intersects each object exactly once, thus the number of segments $A \cap \Lambda$ and $B \cap \Lambda$ included in a line is at most 2. The computation of the force histogram is equivalent to computing forces between two segments, thus lowering the computation time. In the worst case, objects A and B have complex shapes (e.g., a fractal-like structure or objects with multiple disconnected components). For any direction θ, the line Λ intersects each objects multiple times, thus the number of segments $A \cap \Lambda$ and $B \cap \Lambda$ may be very high. This will lead to computing forces between multiple segments, thus increasing the computation time.

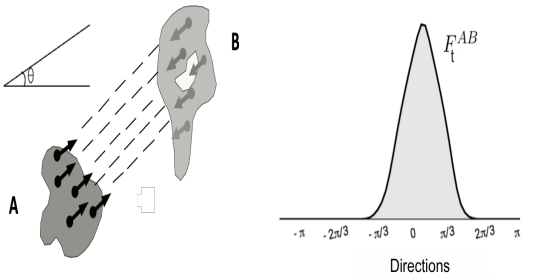


Figure 1: Force histogram: (a) Consider two objects A and B; $F_t^{AB}(\theta)$ is the scalar resultant of elementary forces in the direction θ. (b) The histogram F_t^{AB} between the objects A and B.

In (Ni et al., 2004), the line-based algorithm for force histogram computation is extended for 3D data. The basic principal remains unchanged. For any direction $\vec{\theta}$ (here defined by a unit vector), the function $F_t^{AB}(\vec{\theta})$ is the scalar resultant of elementary forces exerted by the voxels of B on those of A. The $\vec{\theta}$ is taken from a set D^K of evenly distributed reference directions (Bourke, 2007). The K reference directions in the set D^K should follow three primary constraints:

1. They should be evenly distributed in 3D space,
2. They should include six primitive directions i.e., (±1,0,0) (left/right), (0,±1,0) (above/below), and (0,0,±1) (front/behind), and,
3. If a direction θ belongs to the set, then the opposite direction $-\theta$ should also be a part of the direction set.

The above constraints indicate $m = 6 + 8l$, where 6 denotes six primitive directions, 8 denotes the re-

gions demarcated by the XY, YZ, ZX plane, and l implies the number of reference directions in each plane. 3D data, however, is significantly larger than 2D data, so it is necessary to follow appropriate optimization techniques. Therefore, a number of irrelevant directions are identified and dropped in order to ease the computational burden. Forces in the remaining directions are computed in a similar fashion to the 2D case. A simple yet powerful test is conducted to drop irrelevant directions. According to the test in (Ni et al., 2004), a direction $\vec{\theta}$ is considered relevant, if $F_t^{AB}(\vec{\theta}) \neq 0$ i.e., for any oriented line Λ, $A \cap \Lambda \neq 0$ and $B \cap \Lambda \neq 0$. Otherwise, a direction $\vec{\theta}$ is irrelevant, and should not be considered while computing the force histogram $F_t^{AB}(\theta)$.

2.2.2 FFT-based Algorithm

The line-based algorithm that is based on the image partitioning and batch processing of pixels (Section 2.2.1) has two significant drawbacks: (1) the computation time is directly proportional to the number of directions (K) in which the forces are evaluated, and (2) it depends on the shape of the object (concave or convex) and their relative position (overlapping or disjoint). To solve the issues encountered by the line-based algorithm, Matsakis and Jing Bo Ni (Ni and Matsakis, 2010) introduced an alternative definition for force histogram computation. Consider A and B are the membership functions of objects in a digital image. The image is extended through zero padding to occupy the infinite space Z^2, where Z denotes the set of all integers. .The force histogram is derived from the mapping Ψ^{AB} defined over discrete space. The mapping Ψ^{AB} from Z^2 to $\mathbb{R}$ is defined as follows:

$$\Psi^{AB}(d,e) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} A(x,y)B(x+d,y+e)dx\,dy \tag{1}$$

This is a spatial correlation that provides raw information about the relative positions of objects, and can be computed quickly by using the Fast Fourier transform (FFT). The force histogram $\hat{F}_t^{AB}(\theta)$ can be computed using the mapping defined in Equation (1). For two objects A and B, the function $\hat{F}_t^{AB}(\theta)$ is defined by:

$$\hat{F}_t^{AB}(\theta) = \int_0^{\infty} \frac{\Psi^{AB}(d\vec{\theta})}{r^t} dr \tag{2}$$

The FFT-based algorithm has been implemented only for 2D raster objects; it has not been implemented for 3D raster objects.

3 FFT-BASED ALGORITHM FOR 3D RASTER DATA

In this section, we introduce the first FFT-based algorithm for the computation of force histogram in the case of 3D raster objects. The basic principle remains the same. A force histogram is computed using a mapping Ψ^{AB} defined over the 3D space, where Ψ^{AB} is the spatial correlation between two 3D objects. A Fast Fourier Transform (FFT) technique is used to compute the mapping Ψ^{AB} quickly. Section 3.1 focuses on the implementation of the spatial correlation (Ψ^{AB}), then Section 3.2 discusses the computation of the force histogram using Ψ^{AB}.

3.1 Implementation of Ψ^{AB}

Consider A and B as two 3D raster objects in a domain $\mathbf{G}_{h\times k\times l}$. Ψ^{AB} provides raw information about the position of object A relative to object B. The spatial correlation Ψ^{AB} between 3D raster objects A and B is defined as follows:

$$\Psi^{AB}(d,e,f) = \sum_{h=0}^{h-1}\sum_{k=0}^{k-1}\sum_{l=0}^{l-1} A(x,y,z)B(x+d,y+e,z+f) \tag{3}$$

The set $\mathbf{G}_{h\times k\times l} = \{-h+1.....h-1\}\times\{-k+1.....k-1\}\times\{-l+1.....l-1\}$ is the effective domain of Ψ^{AB}. In Equation (3), we suppose that $B(x+d,y+e,z+f) = 0$, if the voxel $(x+d,\ y+e,\ z+f)$ lies outside of $\mathbf{G}_{h\times k\times l}$; such that $(x+d,y+e,z+f) \notin \{0\ldots h-1\} \times \{0\ldots k-1\} \times \{0\ldots l-1\}$. It is not difficult to see that only voxels (d,e,f) that are in the effective domain $\mathbf{G}_{h\times k\times l}$, will satisfy $\Psi^{AB} \neq 0$. Consider the reflection of object A about the origin, such that $A'(-x,-y,-z) = A(x,y,z)$. By replacing $A(x,y,z)$ with $A'(x',y',z')$ in Equation 3, where $x' = -x$, $y' = -y$ and $z' = -z$, we get,

$$\Psi^{AB}(d,e,f) = \sum_{h=0}^{h-1}\sum_{k=0}^{k-1}\sum_{l=0}^{l-1} A(x',y',z')B(d-x,e-y,f-z) \tag{4}$$

Equation (4) refers to the 3-D discrete convolution. Using matrix notation, we get:

$$[\Psi^{AB}] = [A'] \otimes [B'], \tag{5}$$

where $\otimes$ is the convolution operator. According to the Convolution theorem, the Fourier transform has the power to convert the convolution operation into an ordinary multiplication, and vice versa. Using the Convolution theorem, equation (5) can be written as:

$$[\Psi^{AB}] = W_{2h-1}^{-1} \cdot W_{2k-1}^{-1}[(W_{2h-1} \cdot W_{2k-1} \cdot [A'] \cdot W_{2l-1}) \times (W_{2h-1} \cdot W_{2k-1} \cdot [B'] \cdot W_{2l-1})] \cdot W_{2l-1}^{-1} \quad (6)$$

where W_i and W_i^{-1} are the Fourier and inverse Fourier transform matrices of order i and $\times$ is the matrix product. From Equation (6), computation of $[\Psi^{AB}]$ requires three 3-D discrete Fourier transforms. Using the Convolution theorem, Ψ^{AB} can be computed in $O(NlogN)$ time as this is the complexity of the Fast Fourier transform (FFT).

3.2 Implementation of $\hat{F}_t^{AB}$

In this section, $\hat{F}_t^{AB}$ is computed using the spatial correlation Ψ^{AB} (Section 3.1). Only a finite number of directions θ are considered. A reasonable choice for θ (Section 2.2.1) is to choose from a set of evenly distributed reference directions D^K. For 3-dimensional objects, Equation (2) takes the following form:

$$\hat{F}_t^{AB}(\theta) = \int_0^{\infty} \frac{\Psi^{AB}(dcos\theta, dsin\theta, z)}{r^t} dr \quad (7)$$

The FFT-based algorithm described in this paper runs in O(NlogN) time. The computation time for the FFT-based algorithm is independent of the type of forces (t) and number of directions (K) in which the forces are measured. All objects regardless of their shape (complex or simple) and relative position (disjoint or overlapping) are treated in the same manner. The experiments conducted in Section 4 validate this.

4 EXPERIMENTS

In this section, F_t^{AB} denotes a force histogram computed using the line-based algorithm and $\hat{F}_t^{AB}$ denotes a force histogram computed using the new FFT-based algorithm. The experiments are conducted on a variety of 3D raster objects. The test data consists of objects generated from quadratic Koch island fractals at iterations 2 and 3, overlapping objects, cubes and segmented cubes (Figure 2). The size of the objects is kept constant ($N = 512^3$) for all experiments, except for the scenario where the effect of size is investigated on the computational time of $\hat{F}_t^{AB}$ and F_t^{AB}. The algorithms are implemented in C, and the experiments were conducted on a mac OS equipped with an 2GHz Intel Core i5 CPU and 8GB of memory. The computation time is measured using the C library function clock().

Table 1: Tversky Index (μ_T), $N = 512^3$, K=1800.

Figure 2	(a)	(b)	(d)
t= 0	0.91	0.86	0.956
t= 1	0.90	0.82	0.950
t = 2	0.87	0.8	0.943

4.1 Comparing Histogram Values

This experiment demonstrates that F_t^{AB} and $\hat{F}_t^{AB}$ are almost identical, as expected. The histogram associated with the F_t^{AB} is denoted as $h_1(\theta)$ and the histogram associated with $\hat{F}_t^{AB}$ is denoted as $h_2(\theta)$. The dissimilarity between $h_1(\theta)$ and $h_2(\theta)$ is measured using the Tversky index (μ_T):

$$\mu_T(h_1, h_2) = \frac{\sum_{i=0}^{K-1} min\{h_1(\theta_i), h_2(\theta_i)\}}{\sum_{i=0}^{K-1} max\{h_1(\theta_i), h_2(\theta_i)\}} \quad (8)$$

μ_T is 1 when the $h_1(\theta)$ and $h_2(\theta)$ histograms are exactly the same and μ_T is 0 when the $h_1(\theta)$ and $h_2(\theta)$ histograms are orthogonal. The μ_T values tend to be close to 1 for the object pairs considered in our experiments (Table 1). The effect of K and N on μ_T has also been studied (Figure 3). The values of μ_T are almost independent of K (Figure 3a) but grow notably and tend towards 1 when N increases (Figure 3b).

4.2 Comparing Computation Time

The results presented in Figure 4a shows that the computation of $\hat{F}_t^{AB}$ is independent of the choice of t; however, F_t^{AB} computes faster when t=1 or t=2. The prominent reason for this is because the computation of each F_t^{AB} devolves into evaluating a number of numerical expressions which are computationally faster when t=1 and t=2.

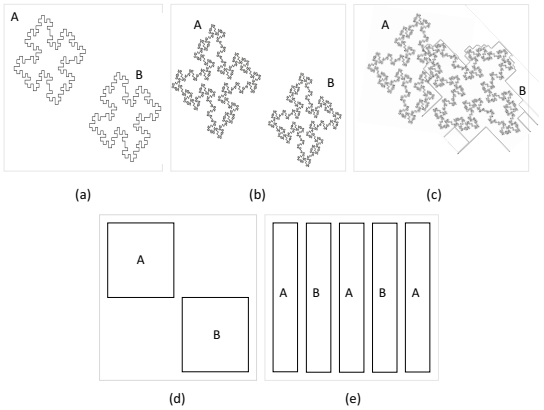
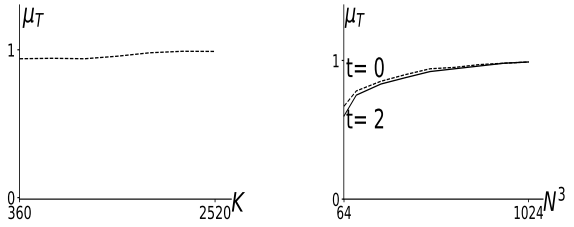
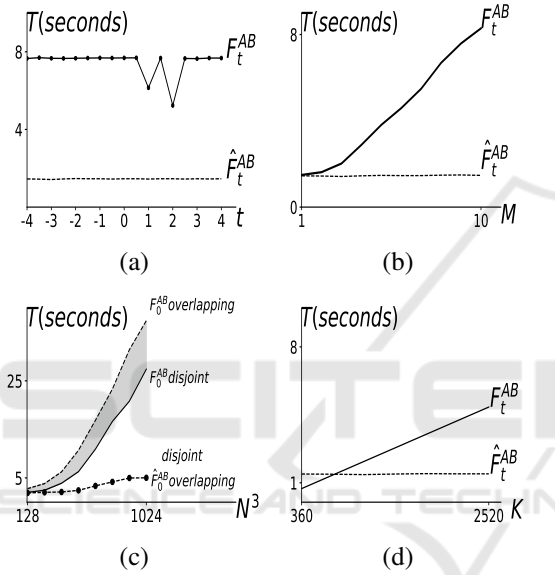


Figure 2: Test Data: The 3D objects used in this experiment are right cylinders whose bases are quadratic Koch island fractals at (a) iteration 2 and (b) iteration 3. Also shown are (c) overlapping objects, (d) cubes and (e) segmented cubes.



(a) (b)

Figure 3: (a) μ_T against the number of directions (K), objects are from Figure 2b and N = 512^3 .(b) μ_T against the number of voxels (N), objects are from Figure 2b and K = 1800.



(a) (b)

(c) (d)

Figure 4: In (a), the objects are from Figure 2a, K= 1800, $N = 512^3$. In (b), the objects are from Figure 2e, M denotes the number of disconnected components in an object, K= 1800, N = 512^3. In (c), the objects are from Figure 2(b,c), K= 1800. In (d), the objects are from Figure 2c, N = 512^3.

In an ideal case, the computational complexity of F_t^{AB} is $O(KN)$ for simple object shapes, whereas for complex shapes (e.g, objects with multiple disconnected parts) the computational complexity of F_t^{AB} is $O(KN\sqrt{N})$. The main reason for this is that for simple convex objects (Figure 2d), any line that runs through both objects intersects each object only once; therefore, computing a force histogram value comes down to computing the forces between two segments. However, for complex shapes (Figure 2e), any line passing through both objects intersects them in multiple segments; therefore, computing a histogram value comes down to computing the forces between a large number of segments which further leads to longer processing times (Section 2.2.1). However, the $\hat{F}_t^{AB}$ always computes in O(NlogN) regardless of the object's shape. This is illustrated by Figure 4b. Furthermore, the relative position (disjoint or overlapping) of the objects also affects the processing time of F_t^{AB}, whereas $\hat{F}_t^{AB}$ stays the same for disjoint as well as overlapping objects (Figure 4c). Finally, when K increases, the processing time of F_t^{AB} also grows linearly. On the other hand, $\hat{F}_t^{AB}$ remains constant (Figure 4d).

5 CONCLUSION

The FFT-based algorithm for the computation of force histogram is a powerful tool to quantitatively represent the relative position between two objects. In earlier work, only 2D objects were considered. In this paper, we have shown that 3D objects can also be handled, and efficiently. The FFT-based algorithm for the force histogram computation described in this paper is computed in O(NlogN) time, where N is the number of voxels in an object. In comparison, the line-based algorithm for the force histogram computation is computed in $O(KN\sqrt{N})$ time, where K is the number of direction in which forces are computed. Furthermore, contrary to the line-based algorithm, the computation time of the FFT-based algorithm is independent of the number of directions, type of forces, shape and relative position of the objects. The FFT-based algorithm is, therefore, an appropriate choice over the line-based algorithm for 3D raster objects, except for the case when the objects' shapes are simple or when the number of directions in which the forces are computed is small. In future work, the ϕ-descriptor algorithm can also be extended for 3D raster objects, and then the processing times and performance of the force histogram and the ϕ-descriptor for 3D raster objects can also be comparatively compared.

REFERENCES

Bourke, P. (2007). Distributing points on a sphere.

Deselaers, T., Keysers, D., and Ney, H. (2004). Features for image retrieval: A quantitative comparison. In *Joint Pattern Recognition Symposium*, pages 228–236. Springer.

Egenhofer, M. (1990). A mathematical framework for the definition of topological relations. In *Proc. the fourth international symposium on spatial data handing*, pages 803–813.

Egenhofer, M. J. (1989). A formal definition of binary topological relationships. In *International conference on foundations of data organization and algorithms*, pages 457–472. Springer.

Frank, A. U. (1996). Qualitative spatial reasoning: Cardinal directions as an example. *International Journal of Geographical Information Science*, 10(3):269–290.

Gapp, K.-P. (1995). Angle, distance, shape, and their relationship to projective relations. In *Proceedings of the 17th annual conference of the cognitive science society*, pages 112–117.

Krishnapuram, R., Keller, J. M., and Ma, Y. (1993). Quantitative analysis of properties and spatial relations of fuzzy image regions. *IEEE Transactions on fuzzy systems*, 1(3):222–233.

Matsakis, P. (1998). *Relations spatiales structurelles et interpretation d'images*. PhD thesis.

Matsakis, P. and Wendling, L. (1999). A new way to represent the relative position between areal objects. *IEEE Transactions on pattern analysis and machine intelligence*, 21(7):634–643.

Miller, H. J. and Wentz, E. A. (2003). Representation and spatial analysis in geographic information systems. *Annals of the Association of American Geographers*, 93(3):574–594.

Miyajima, K. and Ralescu, A. (1994). Spatial organization in 2d segmented images: representation and recognition of primitive spatial relations. *Fuzzy Sets and Systems*, 65(2-3):225–236.

Naeem, M. and Matsakis, P. (2015). Relative position descriptors. In *Proceedings of the International Conference on Pattern Recognition Applications and Methods-Volume 1*, pages 286–295. SCITEPRESS-Science and Technology Publications, Lda.

Ni, J. and Matsakis, P. (2010). An equivalent definition of the histogram of forces: Theoretical and algorithmic implications. *Pattern Recognition*, 43(4):1607–1617.

Ni, J., Matsakis, P., and Wawrzyniak, L. (2004). Quantitative representation of the relative position between 3d objects. In *VIIP 2004 (4th IASTED Int. Conf. on Visualization, Imaging, and Image Processing)*, pages 452–289.

Ryoo, M. S. and Aggarwal, J. K. (2009). Spatio-temporal relationship match: video structure comparison for recognition of complex human activities. In *ICCV*, volume 1, page 2. Citeseer.

Schneider, M. and Behr, T. (2005). Topological relationships between complex lines and complex regions. In *International Conference on Conceptual Modeling*, pages 483–496. Springer.

Skubic, M., Blisard, S., Carle, A., and Matsakis, P. (2002). Hand-drawn maps for robot navigation. In *AAAI Spring Symposium, Sketch Understanding Session*, page 23.

Skubic, M., Matsakis, P., Forrester, B., and Chronis, G. (2001). Extracting navigation states from a hand-drawn map. In *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, volume 1, pages 259–264. IEEE.

Smeulders, A. W., Worring, M., Santini, S., Gupta, A., and Jain, R. (2000). Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (12):1349–1380.