Uncertain Geometry in Computer Vision

Peter Veelaert

Hogeschool Gent, Dept. INWE, Belgium
peter.veelaert@hogent.be

Abstract. We give an overview of the main ideas and tools that have been employed in uncertain geometry. We show how several recognition problems in computer vision can be translated into combinatorial optimization problems that involve intersection hypergraphs, and how we can obtain approximate solutions for these problems when we replace the hypergraphs by intersection graphs. The statistical properties of these graphs are important when we design algorithms for the extraction of geometric primitives from images. We illustrate the use of uncertain geometry with examples involving the detection of circles and the computation of transformations between images.

1 Introduction

The computation of the uncertainty of geometric primitives and transformations is an important problem in computer vision. In the classical approach, the uncertainty of geometric parameters is estimated by statistical analysis. This approach has the disadvantage, however, that the computation soon becomes too complicated unless one introduces simplifying, but often unrealistic assumptions with regard to the statistical model, e.g. that uncertainty can be represented by uncertainty ellipsoids.

Since a purely statistical approach has many shortcomings, other methods have been proposed. The way in which geometric uncertainty has been modeled and handled, however, depends very much on the field of application. Several methods are aimed at obtaining numerical robustness in geometric modeling: robust geometric computation [32], interval geometry [16,17], rounded geometry [19]. One of the prototype problems in this field is the robust intersection of line segments [32]. Also in robotics and mechanical design there is a need to deal with impreciseness. The models proposed there include the use of finite precision arithmetic [12], the use of probability density functions [5,6], and the use of tolerance zones for mechanical parts [7].

When the application is the extraction of geometric primitives and relationships, uncertainty involves more than numerical errors. Some ad-hoc approaches have been quite successful. Lowe as well as Nacken introduce significance measures for geometric relations, e.g. a function that measures how far two line segments are from being collinear [11,13]. A recurring problem, however, is that such a function does not necessarily satisfy the basic rules of geometry, such as the transitivity of collinearity [13]. In this paper we advocate a more uniform approach to geometric uncertainty, consisting of one common methodology applied to many geometric problems, which also takes care of transitivity and other properties [29].

E. Andres et al. (Eds.): DGCI 2005, LNCS 3429, pp. 359–370, 2005.

This paper gives an overview of the main ideas and tools that have been employed in uncertain geometry. Positional and geometric uncertainty is modeled by polytopes instead of ellipsoids. Such an approach has several advantages. First, in applications regarding positional or geometric uncertainty, polytopes can give a much better idea about the shape of the uncertainty region. Second, when using polytopes to capture uncertainty, many recognition problems in computer vision can be reformulated as either convex or combinatorial optimization problems for which standard algorithms are known. The paper also mentions more recent work on geometric transformations, where the positions of the points are not precisely known, a topic inspired by registration problems in image processing.

Section 2 describes how geometric uncertainty is modeled in this paper. The advantages of the approach become clear in Section 3 in which we translate the extraction of geometric primitives into a combinatorial optimization problem involving hypergraphs. Section 4 describes how intersection hypergraphs can be replaced by intersection graphs. The statistical properties of intersection graphs are the subject of Section 5.

2 Modeling Geometric Uncertainty

What are the geometric problems that must be solved in computer vision, and for which of these problems uncertainty is an important issue? Uncertainty is almost unavoidable in one of the most common tasks in computer vision, i.e. when we extract geometric primitives from real digital images, look for relations between these primitives, or look for geometric transformations between points or primitives. The positional uncertainty of the extracted features has several causes: distortion by lenses, which, when measured in terms of pixels actually gets larger when the resolution of a digital camera gets better, the positional error made by a feature detector, or the geometric imperfectness of the real world, i.e. straight edges are not perfectly straight, round objects do not form perfect circles, hand-drawn figures are loosely drawn.

Another kind of uncertainty is due to the misclassification of features. A feature detector may report features that are not really there (false positives), may fail to report features (false negatives), or in the best case return a likelihood measure for the presence of a feature. Feature classification errors are mainly caused by noise and illumination conditions, combined with the often ill-posed nature of feature detection, in particular when we want to detect features characterized by high intensity variations. Therefore, while extracting geometric primitives from an image we must cope with different kinds of uncertainty at different stages of the process.

The examples used most often to illustrate the concepts of uncertain geometry are digitized straight lines and planes [24, 27, 25, 26, 29, 28]. Two less-known applications are circles and transformations.

Example 1. Circles. Suppose we don't know the exact location of the center (a, b) and radius r of a circle in an image, but we are given a set S_i of image points (x_j, y_j) that lie inside the circle, and a set S_o of image points (x_k, y_k) that lie outside a circle, as shown in Figure 1. What is known about the position of the center (a, b) of the circle? The parameters a, b and r must satisfy

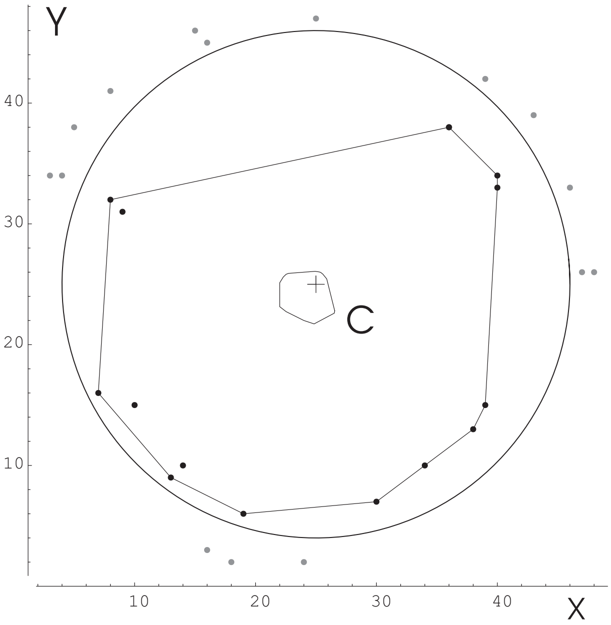

Fig. 1. Uncertainty region for circle centers. The gray dots represent the points that must lie outside each circle; the black dots must lie inside

$$(x_j - a)^2 + (y_j - b)^2 < r^2, \tag{1}$$
$$r^2 < (x_k - a)^2 + (y_k - b)^2 \tag{2}$$

for all $(x_j, y_j) \in S_i$ and $(x_k, y_k) \in S_o$. Such a solution exists if and only if

$$(x_j - a)^2 + (y_j - b)^2 < (x_k - a)^2 + (y_k - b)^2 \tag{3}$$

for all possible point pairs $(x_j, y_j) \in S_i$ and $(x_k, y_k) \in S_o$. Or equivalently, we must have

$$2a(x_k - x_j) + 2b(y_k - y_j) < {x_k}^2 - {x_j}^2 + {y_k}^2 - {y_j}^2, \tag{4}$$

which determines a half-plane bounded by the perpendicular bisector of the points (x_j, y_j) and (x_k, y_k). The intersection of all half-planes determines the uncertainty region C of the center, which is also shown in Figure 1, together with one of the circles that separates S_i and S_o and has its center (+) in C. Likewise we can compute an uncertainty interval for the radius of the circles separating S_i and S_o. The upper bound for this interval is the radius of the largest circle with center in C and not containing any point of S_o. The lower bound is the radius of the smallest circle with center in C and containing all points of S_i.

Example 2. Uncertainty of geometric transformations. Here we want to model geometric transformations for which the position of the image points is not known precisely.

More specifically, to model the uncertainty of the image (x', y') of a point $p = (x, y)$ in $\mathbb{R}^2$, we define the uncertainty region R as a convex polygon bounded by n halfplanes,

$$\begin{aligned} r_1x' + s_1y' &\geq 1 \\ r_2x' + s_2y' &\geq 1 \\ &\dots \\ r_nx' + s_ny' &\geq 1 \end{aligned} \tag{5}$$

in the $x'y'$-plane. We let $\mathcal{T}(p, R)$ denote the set of all transformations T that map p into the uncertainty region R. In this example we restrict ourselves to affine transformations, defined as

$$\begin{aligned} x' &= ax + by + e \\ y' &= cx + dy + f \end{aligned} \tag{6}$$

with (x', y') as image-point, (x, y) as source-point and the 6 parameters of the transformation: $a, \dots, f$. Then the uncertainty of the transformations $\mathcal{T}(p, R) : p \rightarrow R$ can be described as a convex polyhedron in 6 dimensions (one dimension for each transformation parameter) by substituting the equations of (6) in (5), yielding

$$\begin{aligned} r_1(ax + by + e) + s_1(cx + dy + f) &\geq 1 \\ r_2(ax + by + e) + s_2(cx + dy + f) &\geq 1 \\ &\dots \\ r_n(ax + by + e) + s_n(cx + dy + f) &\geq 1. \end{aligned} \tag{7}$$

The notion of an uncertainty transformation can be extended to sets of points: let S be a finite set of points $p_i \in \mathbb{R}^2$, $\mathcal{R}$ a collection of subsets R_j of $\mathbb{R}^2$, and f a mapping that assigns each point in S to its corresponding subset in $\mathcal{R}$, then $\mathcal{T}(S, \mathcal{R}, f)$ denotes the set of all affine transformations that map each point p_i into the set $R_j = f(p_i)$. Clearly, we have $\mathcal{T}(S, \mathcal{R}, f) = \cap_i \mathcal{T}(p_i, f(p_i))$. Thus $\mathcal{T}(S, \mathcal{R}, f)$ is a convex polytope in a 6 dimensional space.

Once a transformation polytope $\mathcal{T}$ has been determined, we can use it to find the image of all other points by uncertain transformations [20]. To be precise, let p be a point not in S and $\mathcal{T}$ a given polytope, then we let $R(p, \mathcal{T})$ denote the uncertainty region resulting from mapping p by the transformations in $\mathcal{T}$; that is, $R(p, \mathcal{T}) = \{q \in \mathbb{R}^2 : q = T(p)$ for some $T \in \mathcal{T}\}$. One can show that $R(p, \mathcal{T})$ is the convex hull of the points $T_k(p)$ where the transformations T_k denote the vertices of the polytope $\mathcal{T}$ [20]. Figure 2 shows how three points p_1, p_2, p_3 and their image regions R_1, R_2, R_3 restrict the set of possible transformations to the transformation uncertainty polytope $\mathcal{T} = \cap_{i=1,\dots,3} \mathcal{T}(p_i, R_i)$, which consists of all transformations that map p_i into R_i, for $i = 1, \dots, 3$. With $\mathcal{T}$ we can compute the uncertainty region $R(q, \mathcal{T})$ for any point q in the plane. Some of these regions are shown in Figure 2.

Examples 1 and 2 involve inequalities that are linear. Unfortunately, this is not always true, even in the case of a straightforward generalization of Example 2.

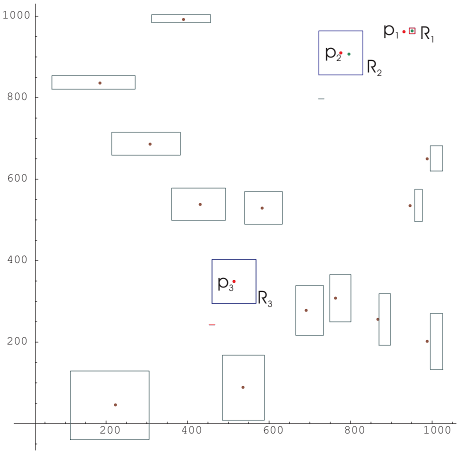

Fig. 2. The uncertainty regions $R(p, \mathcal{T})$ for a number of points where the transformation uncertainty polytope has been determined by (p_i, R_i) for $i = 1, \ldots, 3$. The size of the uncertainty regions varies with the position of the point q

Example 3. Uncertain source and image points. Here we consider transformations where the positions of the source points as well as the positions the image points are uncertain. If we define the uncertainty region of the image point by inequalities of the form $r_i x' + s_i y' \geq 1$, the uncertainty of the source point by inequalities of the form $p_j x + q_j y \geq 1$ and the affine transformation as in (6), then the parameters of the transformation satisfy a system of inequalities of the form

$$r_i(ax + by + e) + s_i(cx + dy + f) \geq 1 \tag{8}$$

$$p_j x + q_j y \geq 1 \tag{9}$$

in which the unknowns $a, \ldots, f, x, y$ appear in a non-linear form (i.e. the product ax).

3 Turning Geometry into Combinatorial Optimization

Although, as illustrated in Example 3, we cannot always describe uncertainty in terms linear inequalities, Examples 1 and 2, and the work done on digital lines and planes, show that it is possible for many geometric problems. When positional uncertainty is formulated in terms of linear inequalities, the detection of a geometric primitive becomes a problem in convex optimization. For example, to find a minimal number of line segments covering as much points as possible we must find the smallest partitioning of a system such that each subsystem has a solution [27]. Is there a general method to solve such problems? One approach is to attack the problem directly e.g. by applying the simplex algorithm to the system, or if necessary to a large number of its subsystems. A more fruitful approach, however, is to employ Helly's Theorem which turns a linear optimization problem into a combinatorial optimization problem. Helly's Theorem can be stated as follows [9, 14, 18]:

Theorem 1 (Helly). *Let $\mathcal{F}$ be a family of convex subsets of $\mathbf{R}^d$ with at least $d+1$ elements. If $\mathcal{F}$ satisfies the following two conditions:*

1. *the intersection of any $d+1$ sets in $\mathcal{F}$ is non-empty,*
2. *$\mathcal{F}$ is finite or all elements of $\mathcal{F}$ are compact,*

then the intersection of all the elements in $\mathcal{F}$ is non-empty.

Helly's Theorem can be immediately applied to the examples of the previous section.

Example 4. A theorem on circles. Let S_i and S_o be defined as in Example 1. Each inequality in (4) defines a convex set in the ab-parameter plane. According to Helly's Theorem the finite system (4) will have a solution if and only if each 3-inequality subsystem of (4) has a solution. The results is an elegant property: the given sets S_i and S_o can be separated by a circle if and only if each 3-point subset of S_i can be separated from each 3-point subset of S_o by a circle.

The property derived in Example 4 is similar to the numerous properties that have been proven for digital straight lines and digital planes, of which the chord property is the most well-known [15]. Although establishing such properties is challenging from a mathematical viewpoint, in computer vision the interesting step is to use Helly's Theorem to reformulate detection problems as combinatorial problems.

Example 5. Circle Detection. We are given two arbitrary sets S_i and S_o of points in the plane that, as in Example 1, must lie either inside or outside a circle. In this example, however, the given sets are not necessarily separable by a circle. To determine the maximal subset of each set, such that the two subsets can be separated by a circle we first write down the system with $|S_i| \times |S_o|$ inequalities as in (4). Next, we construct a 3-uniform intersection hypergraph H with $|S_i| \times |S_o|$ vertices, where each vertex corresponds to one of the inequalities. The hyperedges of H are formed by the triples of vertices that correspond to those 3-inequality subsystems of (4) that have a solution. The largest complete 3-uniform subhypergraph in H corresponds to the subsets $S_1 \subseteq S_i$, $S_2 \subseteq S_o$ such that S_1 and S_2 can be separated by a circle, and $|S_1| + |S_2|$ is maximal.

A further extension of the detection problem is the grouping problem, where multiple instances of a geometric primitive must be grouped according to some criterion, e.g. size.

Example 6. Grouping Circles of Similar Size. We are given a collection of sets S^k of points, where the points of each set S^k have been partitioned into a set S_i^k of points that must lie inside a circle, and a set of points S_o^k that lie outside it. Then, for each set S^k we can determine the uncertainty interval for the radius of the circles that contain the inside points and exclude the outside points. According to Helly's Theorem, if the interval graph of the radii contains a clique of size N, then there are N circles with a common radius corresponding to the sets S^k represented by the vertices of the clique. Thus the interval graph can be used to extract the largest clique of circles that have the same radius, or likewise the graph can be used to partition the circles by a minimum clique covering algorithm into a minimal number of groups so that each group consists of circles that have a similar radius.

The reformulation of a geometric problem as a combinatorial problem has even more advantages, because it becomes easier to add other criteria, for example, the requirement that the points inside a circle should not be too far apart from each other.

Example 7. Additional Constraints for Circles. When we look for a circle that separates the maximum number of points, as in Example 5, we may exclude circles whose inside points are too far apart from each other. To be precise, if we have a circle C then we may require that for each point $p \in S_i$ lying inside C there is at least one other point $q \in S_i$ lying inside C such that the distance between p and q is less than a given threshold. This can be modeled by a graph G whose vertices represent the points in S_i, and in which two vertices are adjacent when the corresponding points lie close enough to each other. The additional constraint states that a circle is only accepted when the vertices that lie inside it form a connected component of the graph G.

In Example 4 Helly's original theorem was used to derive a property for circles in images. Helly's Theorem has been the subject of further extensions and variations. Danzer *et al* and Hadwiger *et al* give extensive overviews of what was known in 1963 [4,8]. More recent advances can be found in [1,2,3,10,23,30], a recent overview in [31]. Some examples in which Helly's extensions are used in uncertain geometry are given in [24]. The application of Helly's Theorem to geometric transformations is described in [20, 21, 22]. Interval graphs have also been used to extract groups of parallel line segments from images [25,26,29].

4 Intersection Graphs as Approximations for Intersection Hypergraphs

Only in $\mathbf{R}^1$ the application of Helly's Theorem leads to intersection graphs. In $\mathbf{R}^d$, $d > 1$, it leads to hypergraphs. Examples illustrating why intersection graphs are not sufficient for $d > 1$ are easy to find. In $\mathbf{R}^2$ the three edges of a triangle are convex sets. Each pair of edges has a non-empty intersection at a corner, the intersection of the three edges, however, is empty. Likewise, in 3-dimensional space a tetrahedron has four 2-dimensional sides. For each triple of sides there is a non-empty intersection at a corner point, but the intersection of the four sides is empty.

In detection problems, however, not only the correctness of the result but also the computational effort needed to obtain a result matters. Even though an intersection graph may give wrong information about common intersections of sets, it may still be useful as long as we can verify afterwards whether an intersection is non-empty. A maximum clique in an intersection graph may not correspond to a non-empty intersection. If the non-emptiness is easy to verify, however, and if the probability of false information is low, then the worst that can happen is that one must try the second largest clique. In fact, by using new types of intersection graphs we can improve the chance that a clique in an intersection graph corresponds to a non-empty intersection of sets, even in $\mathbf{R}^d$, $d > 1$.

The general idea is the following. Let H be a 3-uniform intersection hypergraph, with N vertices v_i each representing a set S_i, and a set of hyperedges $\{v_i, v_j, v_k\}$ such that $S_i \cap S_j \cap S_k \neq \emptyset$. We construct a graph G with $N(N-1)/2$ vertices denoted as $v_i v_j$. The

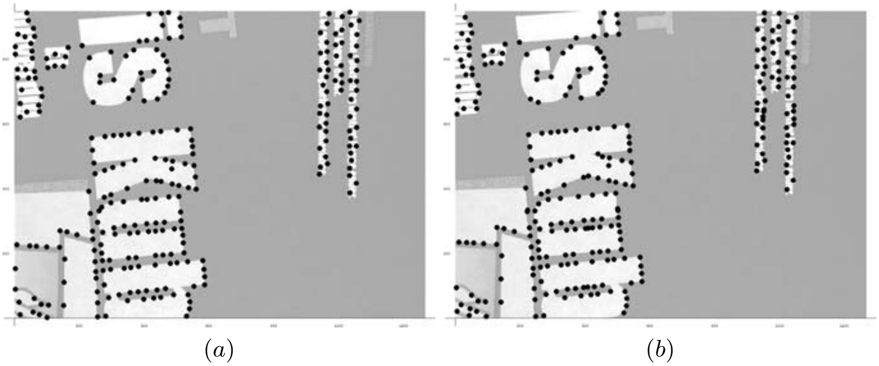

Fig. 3. Image (a) shows the feature points detected in the reference image, while the feature points detected for the second, transformed test image are depicted in image (b)

edges of G consist of all vertex pairs $\{v_iv_j, v_kv_l\}$ satisfying $S_i \cap S_j \cap S_k \cap S_l \neq \emptyset$. The motivation to replace H by G is as follows. Suppose we find a clique in this graph with N vertices $v_iv_j, v_kv_l, \ldots, v_mv_n$. For each edge $\{v_iv_j, v_kv_l\}$ we have $S_i \cap S_j \cap S_k \cap S_l \neq \emptyset$ and therefore $S_i \cap S_j \cap S_k \neq \emptyset, \ldots, S_j \cap S_k \cap S_l \neq \emptyset$ for the four combinations of three sets in a collection of four. Since the clique has $N(N-1)/2$ edges this implies that for the collection $C = \{S_i, S_j, \ldots, S_n\}$ there are at least $4N(N-1)/2$ intersections among the $2N(2N-1)(2N-2)/6$ possible intersections of 3 arbitrary sets in C that are known to have a non-empty intersection. That is, the ratio of triples that have been verified to the total amount of triples is $6(N-1)/((2N-1)(2N-2))$. This means that if one finds a clique in G for which N is not too large, there is a good chance that it also corresponds to a clique in the hypergraph H.

Example 8. Intersection Graphs for Geometric Transformations. We illustrate the use of intersection graphs for transformations. Consider the transformation polytopes of a transformation of the form

$$\begin{aligned} x' &= ax + e \\ y' &= dy + f, \end{aligned} \tag{10}$$

that is a transformation limited to a scaling and translation.

These transformations can be used to solve registration problems. Figure 3 shows an example where in two similar images points have been marked as feature points by a feature detector [22]. We must find a transformation that maps the left image upon the right image as good as possible. Maximal bounds are known for the transformation parameters, i.e. $0.9 \leq a, b \leq 1.1$, and $-20 \leq e, f \leq 20$. We also know that the feature detector is not completely accurate, that is even after adequate scaling and translation, a feature point in the second image may still be displaced a few pixels from the corresponding feature point in the first image.

To find the best transformation we proceed as follows. Let $\mathcal{T}_g$ denote the polytope defined by the inequalities $0.9 \leq a, b \leq 1.1$, and $-20 \leq e, f \leq 20$. We select a small

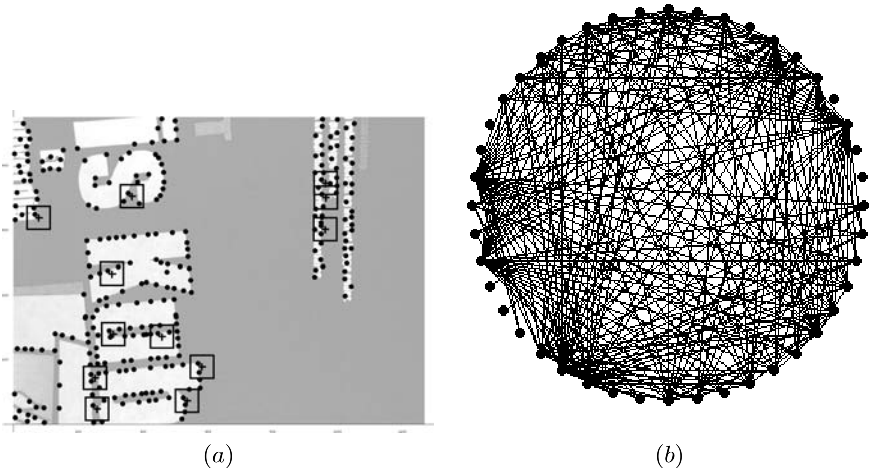

Fig. 4. In image (a), the uncertainty regions are centered on a subset of source feature points $(+)$ projected in the test image with the image feature points $(\bullet)$. All candidate feature image points are located within these regions. The intersection graph for situation (a) is shown in (b)

number of randomly selected feature points p_i in the first image (source points), marked by crosses in Figure 4(a). Around each source point we place a large rectangle R_i defined by $R_i = R(p_i, \mathcal{T}_g)$. Next we construct a graph G whose vertices correspond to the image points q_{ij} that are located inside the regions R_i. Around each point q_{ij} we define a small rectangle Q_{ij} (only a few pixels wide) which takes into account the displacement error made by the feature detector. We compute the polytopes $\mathcal{T}_{ij} = \mathcal{T}_g \cap \mathcal{T}(p_i, Q_{ij})$ as well as the graph G shown in Figure 4(b) which is the intersection graph of the polytopes $\mathcal{T}_{ij}$. Although G is an intersection graph, the existence of an edge in G corresponds to a non-empty intersection of three polytopes (not two), that is, $\mathcal{T}_g \cap \mathcal{T}(p_i, Q_{ij}) \cap \mathcal{T}(p_k, Q_{kl}) \neq \emptyset$. Therefore, as practice shows, a clique in G corresponds almost always to a non-empty transformation polytope, containing transformations that map all the feature points in the clique from the first to the second image.

Intersection graphs were also used as a replacement for 3-uniform hypergraphs to find and group collinear and concurrent line segments [28, 27].

5 Statistical Properties of Intersection Graphs

Real images obey statistical laws. It is almost impossible, however, to estimate the probability density function of, say, the lengths of line segments in a typical indoor scene. Nonetheless, we know for example that parallelism occurs often in real scenes and that the uncertainness of parallel relations in a digital image is mostly due to the impreciseness of the image formation process. Thus, because of transitivity, a graph representing parallel relationships in a digitized image will resemble a collection of disjoint cliques, provided the image data are sufficiently good. More generally, the graphs (and hypergraphs) that appear in uncertain geometry are far from being random graphs. This has important consequences from a computational viewpoint, since many combinatorial optimization problems are NP-complete. With standard algorithms it will take much computational effort to find a maximum clique in a large random graph (> 50 vertices). However, intersection graphs (or hypergraphs) are not random and we can use the statistical properties of the graphs to design an efficient clique finding algorithm.

Example 9. Maximum clique for geometric transformations. For geometric transformations, the intersection graph G as constructed in Example 8 usually contains a single large clique, to which some additional vertices are loosely connected. That is, there is a subset of vertices that have large degree forming a clique, and additional vertices that have low degree. Furthermore, the intersection graph is also known to be a r-partite graph. This results in a considerable speed up for the maximum clique algorithm, since it must only look at subgraphs that contain at most one vertex of each subset in the partition.

This leads to the following algorithm. First, we use Turan's Theorem to compute a lower bound b_l for the maximum clique size. According to Turan's Theorem a graph with n vertices and without a clique of size $p, p > 1$ can have at most $(1 - 1/(p - 1))n^2/2$ edges. Thus we can compute a lower bound for the size of the maximum clique given n and the number of edges $|E|$. Furthermore, we can improve this lower bound by eliminating one by one the vertices of minimal degree from the graph G and recalculating the minimal clique size for each subgraph. Next, we derive an upper bound b_u for the maximum clique size. Let $d_1, d_2, \ldots, d_m$ be the ordered, decreasing degree sequence of the graph G. Then an upper bound is given by the maximum value of the index k such that $d_k \geq (k - 1)$, since for a clique of size k we need at least k vertices of degree $d_i \geq k - 1$.

We then look at all the subgraphs of order b_l of G, and we verify whether they are cliques. Finally, if $b_u > b_l$, we try to extend each of this cliques by adding one vertex at a time. Note that the difference $b_u - b_l$ is a good indication of the computational effort that will be needed to find a maximum clique. If the difference between b_l and b_u is too large (typically $>$ 2), then the graph G is not what we expect and the feature points are not well chosen.

Likewise, in [27] graphs replaced hypergraphs to find collinear line segments. An algorithm based on simplicial elimination orderings was used to find clique coverings of graphs. The method works because the line segment configurations for which cliques do not correspond to non-empty intersections are very rare, and can easily be detected.

6 Conclusion

This paper describes a methodology, which has been applied to several recognition and detection problems in computer vision. Broadly speaking it can be summarized as follows. Linear inequalities are used to model positional and parametric uncertainty. Next, Helly's Theorem is used to reformulate a problem about inequalities as a combinatorial problem involving intersection hypergraphs, to which other combinatorial constraints can be added also in the form of graphs. Finally, to approximate the hypergraph optimization problem, we replace it by a graph optimization problem, and we use the statistics of the graph to select a good detection or grouping algorithm.

What are the limitations of this approach? First, not every form of positional or parametric uncertainty can be modeled by linear inequalities, as Example 3 shows. Nonetheless, the research done on lines, planes and transformations proves that at least some important geometric problems can be solved this way. Second, one possible limitation is the occurrence of hypergraphs as well as NP-complete optimization problems. In practice, however, this has not been an important issue yet. For every case in which this methodology was tried it has always been possible to design efficient heuristic algorithms. What are the benefits of the approach? Applications in computer vision show that it is beneficial to have a more precise model for uncertainty than the ad-hoc approaches that are often taken. Modeling uncertainty by convex sets is certainly an improvement when compared to the use of uncertainty ellipsoids, while keeping computational complexity still acceptable.

References

1. A. B. Amenta, *Helly Theorems and Generalized Linear Programming*, PhD thesis, University of California at Berkeley, 1979.
2. M. Atallah and C. Bajaj, "Efficient algorithms for common transversals", *Inform. Process. Lett.* , vol. 25, pp. 87–91, 1987.
3. D. Avis and M. Doskas, "Algorithms for high dimensional stabbing problems", *Discrete Applied Math.* , vol. 27, pp. 39–48, 1990.
4. L. Danzer, B. Grünbaum, and V. Klee, "Helly's theorem and its relatives", *in Proceedings of the Symposium on Pure Mathematics*, vol. 7, Convexity, pp. 101–180, Providence, RI, 1963. American Mathematical Society.
5. H. F. Durrant-Whyte, "Uncertain geometry in robotics," *IEEE Trans. Robotics Automat.* , pp. 23–31, 1988.
6. H. F. Durrant-Whyte, "Uncertain geometry," in *Geometric Reasoning* (Kapur and Mundy, eds.), pp. 447–481, Cambridge: MIT Press, 1989.
7. A. Fleming, "Geometric relationships between toleranced features," in *Geometric Reasoning* (Kapur and Mundy, eds.), pp. 403–412, Cambridge: MIT Press, 1989.
8. H. Hadwiger and H. Debrunner, *Combinatorial Geometry in the Plane*, Holt, Rinehart and Winston, New York, 1964.
9. E. Helly, "Über Mengen konvexer Körper mit gemeinschaftligen Punkten", *Jahresber. D.M.V.* , vol. 32, pp. 175–176, 1923.
10. D. G. Larman, "Helly type properties of unions of convex sets", *Mathematika*, vol. 15, pp. 53–59, 1968.
11. D. Lowe, "3-d object recognition from single 2-d images," *Artificial Intelligence*, vol. 31, pp. 355–395, 1987.
12. V. J. Milenkovic, "Verifiable implementations of geometric algorithms using finite precision arithmetic," in *Geometric Reasoning* (Kapur and Mundy, eds.), pp. 377–401, Cambridge: MIT Press, 1989.
13. P. Nacken: A metric for line segments. *IEEE Trans. Pattern Anal. Machine Intell.*, **15**,1312–1318, 1993.
14. R. T. Rockafellar, *Convex Analysis*, Princeton Univerisity Press, Princeton, 1970.
15. A. Rosenfeld, "Digital straight line segments", *IEEE Trans. Comput.*, vol. 23, pp. 1264–1269, 1974.

16. M.G.Segal and C.H.Sequin, Consistent calculations for solids modeling. *Proc. 1st Annual ACM Sympos. Comput. Geom.*, pp. 29-38, 1985.
17. M.G.Segal, Using tolerances to guarantee valic polyhedral modeling results. *Comput. Graph. (Proc. SIGGRAPH '90)*, Vol. 24, pp. 105-114, 1990.
18. J. Stoer and C. Witzgall, *Convexity and Optimization in Finite Dimensions I*, Springer, Berlin, 1970.
19. K. Sugihara, On finite-precision representations of geometric objects. *J. Comput. Syst. Sci.*, Vol. 40, pp. 2-18, 1989.
20. Teelen, K., Veelaert, P.: Uncertainty of affine transformations in digital images. Proceedings of ACIVS 2004 (Advanced Concepts for Intelligent Vision Systems), Brussels, (2004) 23–30.
21. Teelen, K., Veelaert, P.: Computing the uncertainty of geometric primitives and transformations. Prorisc, Velthoven, (2004).
22. Teelen, K., Veelaert, P.: Computing the uncertainty of transformations in digital images, accepted for SPIE's Conference on Vision Geometry XIII, San Jose (2005).
23. H. Tverberg, "Proof of Grünbaum's conjecture on common transversals for translates", *Discrete Comput. Geom.*, vol. 4, pp. 191–203, 1989.
24. P. Veelaert, "Geometric constructions in the digital plane," *J. Math. Imaging and Vision*, vol. 11, pp. 99–118, 1999.
25. P. Veelaert, "Line grouping based on uncertainty modeling of parallelism and collinearity," in *Proceedings of SPIE's Conference on Vision Geometry IX*, (San Diego), pp. 36–45, SPIE, 2000.
26. P. Veelaert, Parallel line grouping based on interval graphs, *Proc. of DGCI 2000*, vol. 1953 of Lecture Notes in Computer Science, pp. 530–541. Uppsala, Sweden: Springer, 2000.
27. P. Veelaert, "Collinearity and weak collinearity in the digital plane," *Digital and Image Geometry*, vol. 2243 of Lecture Notes in Computer Science, pp. 434–447, Springer, 2001.
28. P. Veelaert, Concurrency of line segments in uncertain geometry, *Proc. of DGCI 2002*, vol. 2301 of Lecture Notes in Computer Science, pp. 289–300. Bordeaux, France: Springer, 2002.
29. P. Veelaert, "Graph-theoretical properties of parallelism in the digital plane," Discrete Applied Mathematics, **125**, (2003), pp. 135-160.
30. R. Wenger, "A generalization of Hadwiger's transversal theorem to intersecting sets", *Discrete Comput. Geom.*, vol. 5, pp. 383–388, 1990.
31. R. Wenger, "Helly-type theorems and geometric transversals", *Handbook of Discrete and Computational Geometry*, eds. Goodman and Rourke, CRC Press, pp. 63–82, 1997.
32. C.K. Yap, "Robust geometric computation", *Handbook of Discrete and Computational Geometry*, eds. Goodman and Rourke, CRC Press, pp. 653-668, 1997.