New visibility partitions with applications in affine pattern matching

Michiel Hagedoorn* Mark Overmars Remco C. Veltkamp
Department of Computer Science, Utrecht University
Padualaan 14, 3584 CH Utrecht, The Netherlands
{ mh, markov, Remco.Veltkamp }@cs.uu.nl

Abstract

Visibility partitions play an important role in computer vision and pattern matching. A visibility partition is formed by regions in which the combinatorial structure of the visibility is constant. This paper studies new types of visibility partitions with applications in affine pattern matching. First, we discuss the conventional visibility partition for finite line segment collections. The machinery from this analysis is then used to describe the two new partitions induced by alternative forms of visibility, namely *trans-visibility* and *reflection-visibility*. We present algorithms that compute each of the partitions in $O((n+k)\log(n)+v)$ randomised time, where k is the number of visibility edges (at most $O(n^2)$), and v is the number of vertices in the partition (at most $O(n^2+k^2)$). In addition, we show that each of the partitions has worst-case combinatorial complexity $\Omega(n^4)$. Our model of computation assumes that the absolute value of a quotient of polynomials of degree at most d can be integrated over a triangle in $\Theta(d)$ time. We use reflection-visibility partitions to compute the *reflection metric* in $O(r(n_A+n_B))$ randomised time, for two line segment unions, with n_A and n_B line segments, separately, where r is the complexity of the overlay of two reflection-visibility partitions (at most $O({n_A}^4+{n_B}^4)$).

1 Introduction

The visibility from a particular viewpoint in a pattern gives a local description of the pattern. The visibilities from all possible viewpoints give a complete representation of the pattern. This insight led to the use of visibility for pattern matching. Visibility has been used for object-recognition as early as 1982, see Chakravarty and Freeman [6]. Visibility is defined in terms of affine geometry, the concept does not depend on Euclidean distances. Therefore, visibility can be used as a tool for affine invariant shape recognition and affine pattern matching.

*supported by Philips Research

We use a strong type of visibility to compute an affine invariant pattern metric, called the reflection metric.

A well-studied structure related with visibility is the visibility graph. For a collection of n planar line segments, the visibility graph is the graph having the endpoints of the line segments as vertices, and edges between vertices for which the corresponding endpoints can be connected by an open line segment disjoint with all segments in the collection. A trivial algorithm, consisting of three nested loops, computes the visibility graph in $O(n^3)$ time. Lee [16] was the first to improve this, by giving an $O(n^2 \log(n))$ time algorithm. Optimal $O(n^2)$ time algorithms were found by Welzl [21], and Asano et al. [3]. If the number of visibility edges is k, an output-sensitive algorithm by Pocchiola and Vegter [20] computes the visibility graph in $O(n \log(n) + k)$ time and $O(n)$ space.

A plausible approach to using visibility for pattern matching would be visibility graph recognition, see Ghosh [9], and Everett [8]. However, visibility graph recognition is a computationally expensive problem, see Lin and Skiena [17]. Moreover, visibility graphs depend heavily on the topology of the pattern, and are therefore by themselves not suitable for robust pattern matching.

Our approach uses a special form of visibility, called reflection-visibility, for defining a similarity measure on line patterns. This measure, the reflection metric, is affine invariant by definition. That is, the distance between affine transformed patterns $t(A)$ and $t(B)$ equals the distance between the original A and B. The reflection metric is robust. It responds proportionally when lines are deformed a little, slightly translated copies of existing lines are added, small cracks are made in lines, and small new lines are added.

To compute the reflection metric we need to know the structure of the visibility partition corresponding to a special type of visibility. Visibility partitions consists of equivalence classes with constant combinatorial visibility. Plantinga and Dyer [19] call this structure the *viewpoint space partition*. The dual of the visibility partition is called the *aspect graph*, see Kriegman and Ponce [15], Bowyer and Dyer [5], and Gigus et al. [11]. The number of possible views, the size of the visibility partition, was investigated, under varying assumptions, by Agarwal and Sharir [1], and de Berg et al. [7]. For polygons, results about visibility partitions were found by Guibas et al. [13], Aronov et al. [2], and Bose et al. [4].

In this paper, we focus on the structure of visibility partitions as two-dimensional arrangements. We also investigate two alternative visibility partitions for increasingly strong types of visibility, called trans-visibility and reflection-visibility. As a start, we consider the standard visibility partition. We use an alternative way to define and describe it that is useful for the other types of visibility. After that, the analysis becomes more interesting as we proceed to the trans-visibility and reflection-visibility partitions. We provide $\Omega(n^4)$ worst-case constructions for the complexity of each of the partitions. Let k be the number of visibility edges (at most quadratic in n) and v be the number of vertices in the partition (at most quadratic in $n + k$). We present randomised algorithms that compute each of the three partitions in $O((n + k) \log(n) + v)$ time. We use this to compute the reflection distance. We assume a model of computation in

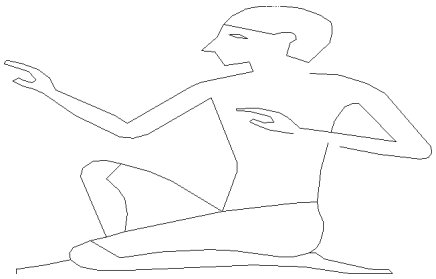

Figure 1: A pattern $A \in \mathcal{R}^2$.

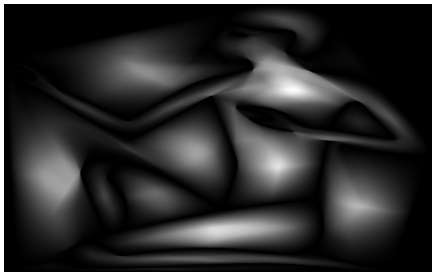

Figure 2: The function ρ_A.

which the absolute value of any rational function, a quotient of polynomials of degree at most d, can be integrated over any triangle in $\Theta(d)$ time. Let A and B be unions of n_A and n_B segments, respectively. If the overlay of the two corresponding reflection-visibility partitions has complexity r, the reflection distance between A and B can be computed in $O(r(n_A + n_B))$ time.

2 The reflection metric

The reflection metric, introduced in [14], defines a distance between finite unions of algebraic curve segments in the plane. If A and B are such unions, the reflection distance is denoted as $d_{\mathtt{R}}(A, B)$. The reflection metric turns the patterns A and B into functions $\rho_A, \rho_B : \mathbb{R}^2 \to \mathbb{R}$, after which the integrated absolute difference of these functions is normalised. The functions are defined using a strong form of visibility, called reflection-visibility. We defer the exact definition of the reflection metric until Section 6.

Figure 1 shows a two-dimensional pattern A consisting of a finite number of line segments. Figure 2 shows the corresponding function $\rho_A : \mathbb{R}^2 \to \mathbb{R}$ represented as a grey-scale image, in which black corresponds with value 0. The examle pattern is hieroglyphic "A1" obtained from the hieroglyphica sign list, see [12].

Let T be the group of affine transformations on $\mathbb{R}^2$. The reflection metric is invariant for T, meaning that $d_{\mathtt{R}}(t(A), t(B)) = d_{\mathtt{R}}(A, B)$ for any affine transformation $t \in T$. As a result, $d_{\mathtt{R}}$ can be used to construct a metric on *affine shapes*, patterns modulo affine transformation:

$$D_R(T(A), T(B)) = \inf_{t \in T} d_{\mathtt{R}}(t(A), B).$$

The reflection metric is robust for various types of effects caused by discretisation and unreliable feature extraction. Slight *deformations* of patterns only increase or decrease distances slightly. Introducing *blur*, by adding new lines near existing lines in a pattern, only causes a proportional change in the reflection distance. Making *cracks* in the interior of lines, splitting them up

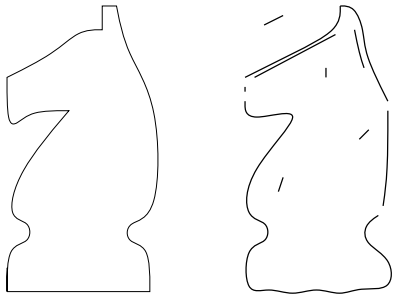

Figure 3: Deformation, blur, cracks and noise.

into multiple smaller ones, only changes the distance proportional in the length of these cracks. Adding *noise* in the form of new lines, far away from other lines, only increases the distance proportionally to the length of the added line. Figure 3 illustrates the effects of deformation, blur, cracks, and noise. The left pattern is the "original", the right pattern is "affected" by the four types of distortion.

In Section 6, we show that the reflection metric can be computed by constructing and traversing an arrangement, which is the overlay of two reflection-visibility partitions. In the following sections, we characterise the visibility, trans-visibility, and reflection partitions. For each of them, we provide a worst-case construction for the combinatorial complexity, and give output-sensitive randomised algorithms that are optimal in the number of segments n.

3 Visibility partitions

Let $\mathcal{S} = \{ S_1, \ldots, S_n \}$ be a collection of closed line segments and let $P = \{ p_1, \ldots, p_m \}$ be the corresponding set of endpoints. In all that follows, we assume that the endpoints are in general position. For convenience, set $A = \bigcup \mathcal{S}$. We say a point $y \in \mathbb{R}^2$ is visible from $x \in \mathbb{R}^2$, if the open line segment $\overline{xy}$ is disjoint with A. For any viewpoint $x \in \mathbb{R}^2$, define the *visible part* of A as the subset $\mathrm{Vp}_A(x) \subseteq A$ given by:

$$\mathrm{Vp}_A(x) = \{ a \in A \mid A \cap \overline{xa} = \varnothing \}.$$

This set is sometimes called the *visibility polygon* of x in A, see [3]. The *visibility star* $\mathrm{Vst}_A(x)$ is the union of all open line segments connecting the viewpoint x with the visible part of A:

$$\mathrm{Vst}_A(x) = \bigcup_{a \in \mathrm{Vp}_A(x)} \overline{xa}.$$

Visibility stars are similar to *view zones*, see [18] pp. 383–391. A view zone is a visibility star extended with all infinite rays from x disjoint with A. Figure 4

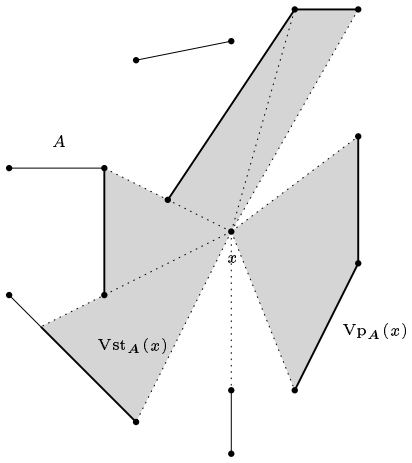

Figure 4: A visibility star.

shows a visibility star for an example consisting of eight line segments with thirteen distinct endpoints. The visible part $\mathrm{Vp}_A(x)$ is drawn thick. The visibility star $\mathrm{Vst}_A(x)$ is the light grey region, including the dotted lines.

Consider the endpoints and segments bounding the visibility star $\mathrm{Vst}_A(x)$ ordered by slope with respect to x. This describes the structure of the visibility star. The visibility star is a finite union of triangles. Each triangle is an intersection of three half-planes. Two of the half-planes are bounded by lines through x and a point in P. The third half-plane is bounded by the line through a segment of $\mathcal{S}$. If a segment $S_i \in \mathcal{S}$ has a visible endpoint p_j, and x is collinear with S_i, then the triangle "degenerates" to the open line segment $\overline{xp_j}$. We are interested in the regions in the plane in which the structure of the visibility star is constant.

We simplify the presentation by introducing an additional "line segment". Let D be an open rectangle containing the union of segments A, and let S_0 be its boundary. We will simply call S_0 a segment. This gives an extended collection of segments $\mathcal{S}' = \{ S_0 \} \cup \mathcal{S}$. Let $A' = S_0 \cup A$. Each ray starting from any point in D, intersects A'.

We need a compact description of the structure of the visibility star $\mathrm{Vst}_A(x)$, for any viewpoint $x \in D$. For this purpose, we define a collection of identifiers, referring either to segments or endpoints. An identifier is an (integer) index subscripted with a p or an s, indicating an endpoint or a segment, respectively. We order the identifiers linearly as follows:

$$0_s < 1_s < \cdots < n_s < 1_p < \cdots < m_p.$$

Each point $a \in A'$ is assigned an identifier $\mathrm{id}(a)$ as follows. If $a = p_i$, then set $\mathrm{id}(a) = i_p$. If $a \in S_i - P$, then set $\mathrm{id}(a) = i_s$.

We represent the structure of the visibility star by a tuple identifiers. Choose

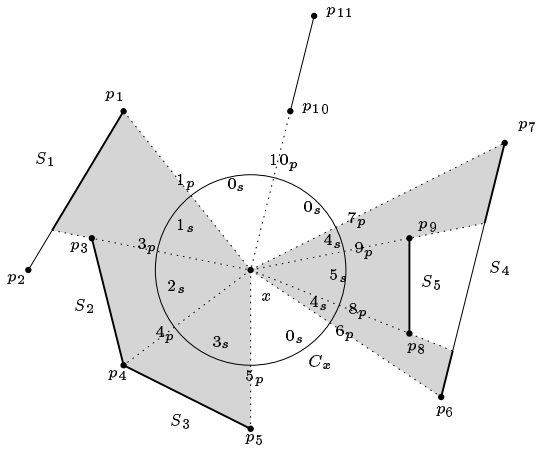

Figure 5: A view map.

any closed disc centred at x disjoint with A. The boundary of such a disc is called the *view circle*, denoted by C_x. We give each point $c \in C_x$ a label $l(c)$, an identifier, as follows. For each point $a \in A'$ visible from x, compute the intersection $c \in C_x \cap \overline{xa}$ and set $l(c) = l(a)$.

The view circle C_x is a disjoint union of inverse images $l^{-1}(d)$, for each identifier d. Each non-empty inverse image $l^{-1}(d)$, the subset of points in C_x with label d, can be decomposed into its connected components. These components are either single points or open arcs. The *view map* of x, denoted with $\mathrm{Vmp}_{\mathcal{S}}(x)$, is a labeled circuit graph whose vertices are the components with their constant labels. We call vertices labeled with endpoint identifiers (i_p) *p-vertices*. We call vertices labeled with segment identifiers (i_s) *s-vertices*. Edges of $\mathrm{Vmp}_{\mathcal{S}}(x)$ are defined by pairs of vertices, constant-label components, with intersecting closures. Figure 5 shows the labeled view circle, inducing the view map, for a collection of four closed line segments having nine distinct endpoints. Labels of p-vertices are indicated on the dotted lines on the outside of C_x. Labels of s-vertices are indicated inside the view circle between successive dotted lines.

The view map $\mathrm{Vmp}_{\mathcal{S}}(x)$ can be represented using a tuple of labels encountered when traversing all edges, starting with some initial vertex and some incident edge. Of all possible tuples, the lexicographically smallest one represents the view map. This representation does not depend on the direction (clockwise or counter-clockwise) in which the labels occur on the view circle. We identify the view map with this unique tuple. In the situation of Fig. 5 this gives:

$$\mathrm{Vmp}_{\mathcal{S}}(x) \quad = \quad (0_s, 1_p, 1_s, 3_p, 2_s, 4_p, 3_s, 5_p, 0_s, 6_p, 4_s, 8_p, 5_s, 9_p, 4_s, 7_p, 0_s, 10_p).$$

The view map $\mathrm{Vmp}_{\mathcal{S}}(x)$ is a function of points $x \in D$. Define points $x, y \in D$ to be equivalent if their view maps (labeled graphs) are isomorphic, that is, $\mathrm{Vmp}_{\mathcal{S}}(x) = \mathrm{Vmp}_{\mathcal{S}}(y)$. This equivalence relation results in a partition of D into

equivalence classes. This *visibility partition* is denoted by $\mathcal{Q}_v(\mathcal{S})$. If x and y lie in the same class $Q \in \mathcal{Q}_v(\mathcal{S})$ of the partition, the visibility stars $\mathrm{Vst}_A(x)$ and $\mathrm{Vst}_A(y)$ have the same structure.

The visibility partition is affine invariant: The partition for the affine transformed set $\mathcal{S}$ equals the affine transformed partition (including the labels). The other two partitions in this paper also have this property. A single class in the visibility partition can have more than one connected component. In the example of a single segment $\mathcal{S} = \{ S_1 \}$ with endpoints $P = \{ p_1, p_2 \}$, the open half-planes left and right of S_1 are the connected components of the equivalence class in $\mathcal{Q}_v(\mathcal{S})$ having view map $(0_s, 1_p, 1_s, 2_p)$.

The visibility partition has the structure of an arrangement induced by a finite union of closed line segments. Each cell in this arrangement is a connected component of an equivalence class in the partition. As the viewpoint x moves continuously within D, changes occur in $\mathrm{Vmp}_{\mathcal{S}}(x)$. Each time such a change occurs, the set of vertices visible from x changes. The sets of viewpoints x on which changes in the viewmap occur form one-dimensional boundaries in the arrangement describing the visibility partition.

We construct a collection of "event segments" for the view map. Let $E_{\mathcal{S}}$ be the collection of (directed) edges in the visibility graph. That is, $E_{\mathcal{S}}$ consist of all pairs of endpoint-indices (i, j) such that p_j is visible from p_i. We extend $E_{\mathcal{S}}$ to a collection $E'_{\mathcal{S}}$ by also including the endpoint-index pairs of each segment in $\mathcal{S}$. Given an endpoint p_i, sort all endpoints p_{j_k}, with $(i, j_k) \in E'_{\mathcal{S}}$, on clockwise angle. This results in a list of endpoint identifiers $j_1, \ldots, j_c$. Let s_k be the segment-identifier of the segment visible from p_i inbetween the angles of p_{j_k} and $p_{j_{k+1}}$ relative to p_i (where $k+1$ is modulo c). We construct event segments bounding the set of points in D from which p_i is visible.

First, we define the collection of event segments $\mathcal{P}_i$. For each $k = 1, \ldots, c$, we include in $\mathcal{P}_i$ the closure of the visible part of segment S_{s_k} (visible from p_i). This includes parts of the special segment with index 0_s.

Second, we construct a segment collection $\mathcal{B}_i$ connecting pairs of segments in $\mathcal{P}_i$. For each $k = 1, \ldots, c$, we construct a closed segment between the two intersections of $\mathrm{ray}(p_i, p_{j_k})$ with segments in $\mathcal{S}'$. If these two intersection coincide, we include no segment in $\mathcal{B}_i$, for that particular k.

The third and last types of segments $\mathcal{X}_i$ are extensions of segments in $\mathcal{S}$. Consider each segment S in $\mathcal{S}$ having endpoint p_i, and having another endpoint p_j. Include in $\mathcal{X}_i$, the closed segment having endpoints p_i, and the intersection of $\mathrm{ray}(p_i, p_j)$ with $\bigcup \mathcal{P}_i$ that is closest to p_i.

The three types of segments result in the arrangement describing the visibility partition. Let $\mathcal{P}$, $\mathcal{B}$, and $\mathcal{X}$, denote the unions of $\mathcal{P}_i$, $\mathcal{B}_i$, and $\mathcal{X}_i$, over all $i = 1, \ldots, m$, respectively.

Theorem 1 *The boundaries in the visibility partition are formed by the event segments:*

$$\bigcup_{Q \in \mathcal{Q}_v(\mathcal{S})} \partial Q = \bigcup \mathcal{P} \cup \bigcup \mathcal{B} \cup \bigcup \mathcal{X}.$$

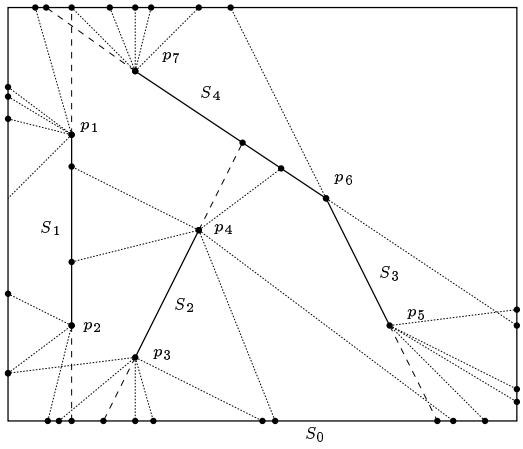

Figure 6: A visibility partition.

Figure 6 shows the visibility partition for four line segments having seven distinct endpoints. The union of $\mathcal{P}$ coincides with A. The segments of $\mathcal{B}$ are drawn dashed. The segments of $\mathcal{X}$ are drawn coarse dashed. The points where event segments meet are indicated as dots. The rectangle containing the segments is the "segment" S_0.

We will now investigate the complexity of the visibility partition. Let k be the number of visibility edges ($k = |E_{\mathcal{S}}|$). It is not difficult to see that total number of event segments in $\mathcal{P}$, $\mathcal{B}$, and $\mathcal{X}$ is $O(n+k)$. Assume this number is s. Since the event segments may intersect, there is an $O(s^2)$ upper bound on the complexity of the visibility partition $\mathcal{Q}_v(\mathcal{S})$ (represented as an arrangement). Since k is $O(n^2)$, it follows that the complexity of $\mathcal{Q}_v(\mathcal{S})$ in terms of n only is $O(n^4)$. Next, we show that this bound is tight.

We sketch an $\Omega(n^4)$ lower-bound construction for the complexity of the visibility partition. Let $n = 2l$. For each $i = 1, \ldots l$, define points $q_i = (i-1, -l)$ and $q_{l+i} = (il, 0)$. Of all lines through these point pairs, at least l^2 have distinct slopes. These l^2 lines form l^4 distinct intersection points. It is possible to perturb the set of q_i $(i = 1, \ldots, n)$ slightly such that the resulting set of p_i $(i = 1, \ldots, n)$ is in general position. To each p_i, we attach a small line segment S_i. By making these attached line segments sufficiently small, the arrangement of event segments gets $\Omega(n^4)$ vertices. Figure 7 shows the lower bound construction.

The visibility partition $\mathcal{Q}_v(\mathcal{S})$ corresponding to $\mathcal{S}$ can be computed as follows. First, we compute the visibility graph for $\mathcal{S}$, in $O(n\log(n)+k)$ time using algorithms by Ghosh and Mount [10] or Pocchiola and Vegter [20]. Using the visibility graph, the view map can be computed for each endpoint. A simple algorithm discovers the segment visible between successive visibility edges incident to each endpoint by performing $\log(n)$ iterations over the visibility edges.

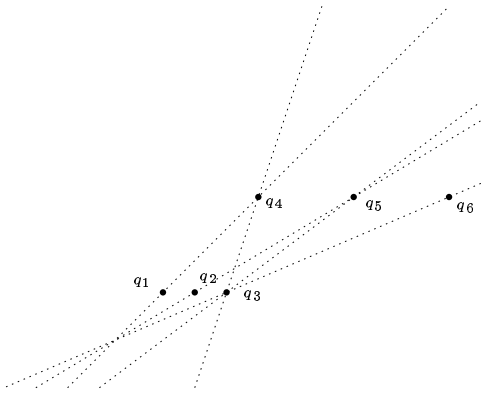

Figure 7: A lower bound construction.

Include the segments of $\mathcal{S}$ as visibility edges. For each (directed) visibility edge $e = (p, q)$, such that no segment adjacent to q is visible to the left of q relative to p, store a pointer $\mathrm{r}(e)$ to the visibility edge incident to q that turns right relative to e. If a segment adjacent to q is visible to the left of q (relative to p), we store a pointer to this segment in $\mathrm{r}(e)$. In each iteration we consider all edges e in the visiblity graph. If $\mathrm{r}(e)$ is not a segment blocking the view to the left we replace $\mathrm{r}(e)$ by $\mathrm{r}(\mathrm{r}(e))$. Analogous, we maintain pointers $\mathrm{l}(e)$, where the roles of left and right exchange. After $O(\log(k)) = O(\log(n))$ iterations, we have found the segments that block the view directly to the left and the right of each directed visibility edge. This takes a total of $O((n + k)\log(n))$ time. Using this information, we can generate the total collection of event segments in $O(n + k)$ time. Let v be the number of intersections in the collection of event segments thus generated. Using randomised incremental construction we construct a trapezoidal decomposition of this collection, see Mulmuley [18] pp. 84-94, in $O((n + k)\log(n) + v)$ time. The arrangement defined by the event segments can be obtained by merging together trapezoids into polygonal cells. Thus, the visibility partition, represented as an arrangement, can be computed using randomised techniques in $O((n + k)\log(n) + v)$.

Theorem 2 *The visibility partition of n segments has worst-case complexity $\Theta(n^4)$. Using randomisation, it can be computed in $O((n + k)\log(n) + v)$ time, where k is the number of visibility edges, and $v = O(n^2 + k^2)$ is the number of vertices in the arrangement.*

4 Trans-visibility partitions

In this section, we consider a stronger notion of visibility, resulting in different stars and partitions. We say that a point $y \in \mathbb{R}^2$ is *trans-visible* from a point $x \in \mathbb{R}^2$ if y is visible from x and both $\mathrm{ray}(x, y)$ and $\mathrm{ray}(y, x)$ intersect A. The

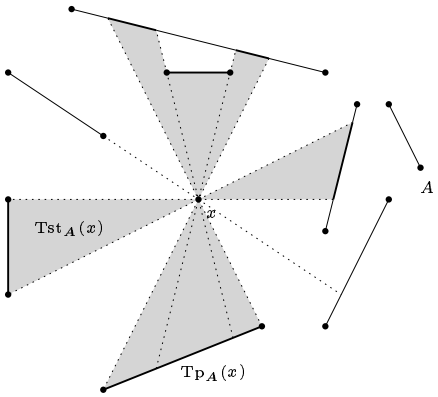

Figure 8: A trans-visibility star.

trans-visible part of A is given by:

$$\mathrm{Tp}_A(x) = \{\, a \in \mathrm{Vp}_A(x) \mid A \cap \mathrm{ray}(a, x) \neq \varnothing \,\}.$$

The *trans-visibility star* $\mathrm{Tst}_A(x)$ is the union of all open line segments between x and the trans-visible part of A.

$$\mathrm{Tst}_A(x) = \bigcup_{a \in \mathrm{Tp}_A(x)} \overline{xa}.$$

Figure 8 shows the trans-visibility star for seven disjoint segments. The dotted line segments, connecting x with endpoints, do not belong to the trans-visibility star. The trans-visible part of A is drawn thick.

Consider a view circle C_x with radius $r > 0$ centred at x. We label the view circle to find the structure of the trans-visibility star at x. We define the labeling l of C_x as follows. We introduce polar coordinates, where the point x acts as the origin, and some fixed direction has the angle 0. Choose an angle $\epsilon > 0$ smaller than the angle between any two endpoints (modulo 2π). For each angle $\alpha \in [0, 2\pi)$, we assign the point $c = (\alpha, r)$ on C_x a label. Let $\mathrm{ray}(x, \alpha)$ be the open ray starting at x in the direction α. There are three cases:

1. If $\mathrm{ray}(x, \alpha)$ or $\mathrm{ray}(x, \pi + \alpha)$ intersects no point of A, then $l(c) = 0_s$.
2. Otherwise, if $\mathrm{ray}(x, \pi + \alpha)$ contains a visible $p \in P$, and $\mathrm{ray}(x, \pi + \alpha - \epsilon)$ or $\mathrm{ray}(x, \pi + \alpha + \epsilon)$ does not intersect A, then $l(c) = \mathrm{id}(p)$.
3. In all other cases, $l(c) = \mathrm{id}(a)$, where a is the visible point of A intersected by $\mathrm{ray}(x, \alpha)$.

It is possible that the previous rules overlap for some $c \in C_x$. We resolve those cases by only choosing the identifier with minimum index assigned to c.

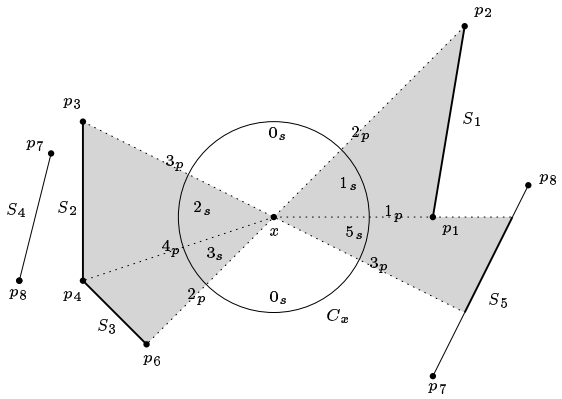

Figure 9: A trans-view map.

The resulting labeling l of the view circle C_x indicates the structure of the trans-visibility star at the point x. The corresponding labeled circuit graph is called the *trans-view map*, denoted by $\mathrm{Tmp}_{\mathcal{S}}(x)$. The trans-visibility star is a finite union of triangles, where each triangle, the intersection of three half-planes, is given by an adjacent *psp* vertex triple in the trans-view map (where the segment-label is nonzero). The lines through x and the two p-vertices define two half-planes, and the s-vertex defines the third half-plane. Sometimes, just the endpoint of a segment intersects $\mathrm{Tp}_A(x)$. In that case, the triangle "degenerates" to the open line segment connecting x with this endpoint. Figure 9 shows a trans-view map for six segments with eleven distinct endpoints. The labels in the trans-view map are indicated along the view circle. Points that belong to the visible part of A are drawn thick.

Order tuples of labels lexicographically using the underlying order on the identifiers. This way, we obtain a unique identifier-tuple describing the trans-view map. For the situation of Fig. 9, this results in the following trans-view map:

$$\mathrm{Tmp}_{\mathcal{S}}(x) \quad = \quad (0_s, 2_p, 1_s, 1_p, 5_s, 3_p 0_s, 2_p, 3_s, 4_p, 2_s, 3_p).$$

Identifying points $x, y \in D$ if $\mathrm{Tmp}_{\mathcal{S}}(x)$ equals $\mathrm{Tmp}_{\mathcal{S}}(y)$, gives a *trans-visibility partition* $\mathcal{Q}_t(\mathcal{S})$. The trans-visibility partition has the structure of an arrangement induced by a finite number of line segments. We identify these line segments below.

We construct a collection of event segments for the trans-view map. Given an endpoint p_i, sort all endpoints p_{j_k}, with $(i, j_k) \in E^l_{\mathcal{S}}$, on clockwise angle, resulting in a list of endpoint identifiers $j_1, \ldots, j_c$. For each $k = 1, \ldots, c$, s_k is the segment-identifier of the segment visible between p_{j_k} and $p_{j_{k+1}}$. We construct event segments, bounding the set of points in D from which p_i is trans-visible. The construction is very similar to that of the visibility partition in the previous section.

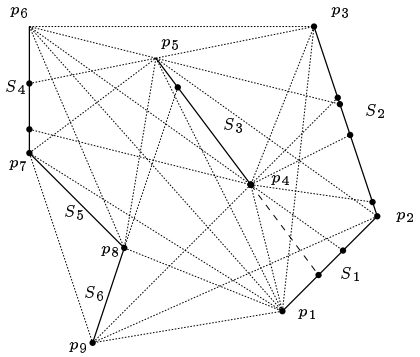

Figure 10: A trans-visibility partition.

First, we define the collection of event segments $\mathcal{P}_i$. For each $k = 1, \ldots, c$, we include in $\mathcal{P}_i$, the closure of the part of segment S_{s_k} that is visible from p_i. Except for $s_k = 0$, in which case we do not include a segment.

Second, we construct a collection of segments $\mathcal{B}_i$, connecting the segments of $\mathcal{P}_i$ at the endpoints. For each $k = 1, \ldots, c$, consider the intersections of $\text{ray}(p_i, p_{j_k})$ with segments in $\mathcal{B}_i$. If two segments of $\mathcal{B}_i$ intersect the ray in distinct points, we include the segment connecting these points. If there are two coinciding intersections, we do not include a segment. If there is only one intersection, we include the closed segment connecting p_i with this intersection.

Third and finally, we construct a segment collection $\mathcal{X}_i$. Consider each segment $S \in \mathcal{S}$ having p_i as an endpoint, where p_j is the other endpoint. If the intersection of $\text{ray}(p_i, p_j)$ with $\mathcal{B}_i$ closest to p_i, is distinct from p_j, then include a closed segment connecting p_i with this intersection.

Again, let $\mathcal{P}$, $\mathcal{B}$, and $\mathcal{X}$ be the unions of $\mathcal{P}_i$, $\mathcal{B}_i$, and $\mathcal{X}_i$, respectively.

Theorem 3 *The boundaries in the trans-visibility partition are formed by the event segments:*

$$\bigcup_{Q \in \mathcal{Q}_t(\mathcal{S})} \partial Q = \mathcal{P} \cup \mathcal{B} \cup \mathcal{X}.$$

Figure 10 shows a trans-visibility partition. The elements of $\mathcal{P}$, $\mathcal{B}$, and $\mathcal{X}$ are shown dotted, dashed, and coarse dashed, respectively.

Using an analysis similar to that for the visibility partition, it follows that the complexity of the trans-visibility is $O(n^4)$. The worst case lower bound of this complexity is also $\Omega(n^4)$. This bound can be achieved with a construction similar to that presented in the previous section for the visibility partition. That construction ensured $\Omega(n^4)$ intersections between event segments. The same result is obtained here by copying the construction, and adding four extra segments that make up a rectangle containing all these intersections.

Computation of the trans-visibility partition is analogous to the computation of the visibility partition. Again, there are $O(n+k)$ event segments. Therefore,

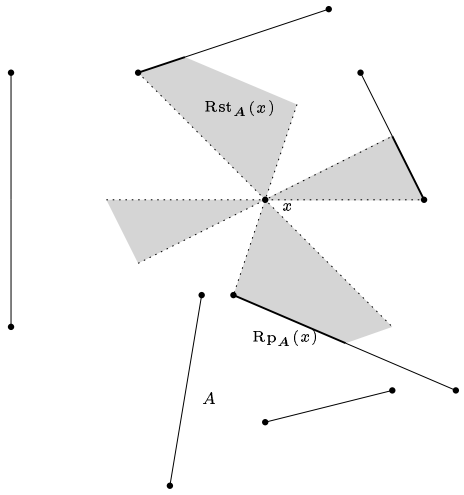

Figure 11: A reflection star.

the trans-visibility partition can be constructed using the visibility partition algorithm from the previous section, resulting in the same time.

Theorem 4 *The trans-visibility partition of n segments has worst-case complexity $\Theta(n^4)$. Using randomisation, it can be computed in $O((n+k)\log(n)+v)$ time, where k is the number of visibility edges, and $v = O(n^2+k^2)$ is the number of vertices in the arrangement.*

5 Reflection partitions

We say a point $y \in \mathbb{R}^2$ is *reflection visible* from a point x, if y is trans-visible from x and the open segment between y and the reflection of y in x is disjoint with A. Define the *reflection-visible part* of A as:

$$\mathrm{Rp}_A(x) = \{\, x+v \in \mathrm{Tp}_A(x) \mid \overline{(x+v)(x-v)} = \varnothing \,\}.$$

Define the *reflection-visibility star* as:

$$\mathrm{Rst}_A(x) = \bigcup \{\, \overline{(x+v)(x-v)} \mid x+v \in \mathrm{Rp}_A(x) \,\}.$$

The reflection-visibility star equals the intersection of a trans-visibility star $\mathrm{Tst}_A(x)$ with its reflection around x. Figure 11 shows a reflection-visibility star for six disjoint segments. The grey area, including the dotted segments, forms the reflection-visiblity star.

Consider a view circle C_x centred at x. We label the view circle to describe the structure of the reflection-visibility star at x. We define the labeling l of C_x as follows.

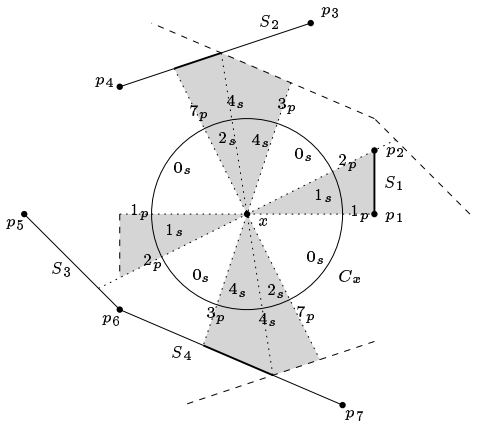

Figure 12: A reflection-view map.

We define a labeling l of a view circle C_x, that represents the structure of the reflection visibility star. We use polar coordinates, such that each point $c \in C_x$, is represented as $c = (\alpha, r)$, where r is the radius of C_x. Let $L(x, \alpha)$ be the line through x and (α, r). Let $\epsilon > 0$ be smaller than the angle between any two endpoints. There are three cases:

1. If $L(x, \alpha)$ intersects a visible point of S_0, then $l(c) = 0_s$.
2. If $L(x, \alpha)$ intersects a visible endpoint $p \in P$ and $L(x, \alpha - \epsilon)$ or $L(x, \alpha + \epsilon)$ intersects a visible point of S_0, then $l(c) = \mathrm{id}(p)$.
3. In all other cases, set $l(c) = \mathrm{id}(a)$, where a is the visible point in $A \cap L(x, \alpha)$ closest to x.

Overlaps in the rules are resolved by choosing the minimum-index identifier.

The labeling l defines a labeled circuit graph called the *reflection view map*, denoted by $\mathrm{Rmp}_S(x)$. Figure 12 shows the view circle along with the labels of the reflection-view map. The reflection-visible part is shown thick. The dashed lines are reflections of segments in the view point.

The reflection-view map $\mathrm{Rmp}_S(x)$ at a point x contains the structure of the reflection-visibility star $\mathrm{Rst}_A(x)$ at a given point. Starting at some vertex and some initial edge, we obtain a tuple of labels representing the reflection-view map. Since this tuple repeats itself, we only take the first half. We choose the lexicographically smallest half-tuple as a unique representation of $\mathrm{Rmp}_S(x)$. In the situation of Fig. 12 this gives:

$$\mathrm{Rmp}_S(x) \quad = \quad (0_s, 1_p, 1_s, 2_p, 0_s, 3_p, 4_s, 2_s).$$

We obtain a reflection-visibility partition by identifying points $x, y \in D$ if their reflection-view maps $\mathrm{Rmp}_S(x)$ and $\mathrm{Rmp}_S(y)$ are equal. When moving x,

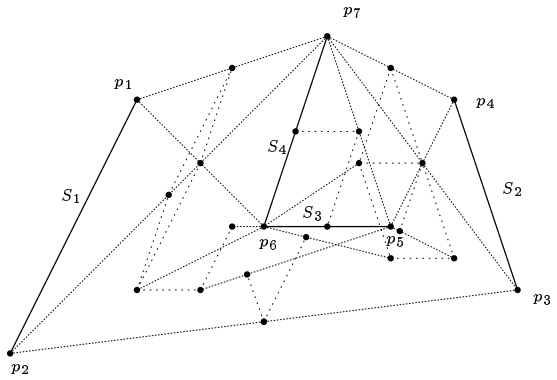

Figure 13: A reflection-visibility partition.

if the reflection-view map changes, then the set of reflection-visible endpoints (relative to x) changes. We see this as follows. If we move x, either an endpoint identifier, a segment identifier, or an intersection identifier, appears or disappears. In case of an endpoint identifier, the result is immediate. In case of a segment identifier, a change occurs only if one endpoint starts to occlude another one. If an intersection identifier disappears it is replaced by an endpoint identifier. Since a change in the set of endpoints also implies a change in the reflection view map, we can conclude the following. Each class reflection-visibility partition is a maximal connected subset of D in which a fixed set of endpoints is reflection-visible.

We construct a collection of event segments for the reflection-view map. Consider an endpoint p_i. Consider $\mathcal{P}_i$, $\mathcal{B}_i$ and $\mathcal{X}_i$ exactly as for trans-visibility. Consider a scaling transformation f that leaves p_i fixed and which has a scaling factor of $1/2$. That is, all coordinates relative to p_i are multiplied by $1/2$. To obtain the desired reflection-view map, replace all segments in $\mathcal{P}_i$, $\mathcal{B}_i$ and $\mathcal{X}_i$ by their images under f. Like before we construct unions of the segment collections over $i = 1, \ldots, m$.

Theorem 5 *The boundaries in the reflection-visibility partition are formed by the event segments:*

$$\bigcup_{Q \in \Omega_r(\mathcal{S})} \partial Q = \bigcup \mathcal{P} \cup \bigcup \mathcal{B} \cup \bigcup \mathcal{X}.$$

Figure 13 shows a reflection-visibility partition.

The complexity of the reflection-visibility partition is at most $O(n^4)$. An $\Omega(n^4)$ worst case lower bound is achieved by copying the construction for trans-visibility and making sure that the new segments lie far enough from the original segments.

To compute the reflection-visibility, we can use the same basic techniques as we used to compute the visibility and trans-visibility partitions.

Theorem 6 *The reflection-visibility partition of n segments has worst-case complexity $\Theta(n^4)$. Using randomisation, it can be computed in $O((n+k)\log(n)+v)$ time, where k is the number of visibility edges, and $v = O(n^2+k^2)$ is the number of vertices in the arrangement.*

6 Computing the reflection metric

As an application of the previously defined structures, we use them in computing the reflection metric. Let $\rho_A(x)$ be the area of the reflection star $\mathrm{Rst}_A(x)$ for each $x \in \mathbb{R}^2$. Observe that for points x outside the convex hull of A, this area is always zero. If we have two finite unions of line segments A and B, the reflection metric $d_{\mathtt{R}}$ is defined as:

$$d_{\mathtt{R}}(A,B) = \frac{\int_{\mathbb{R}^2} |\rho_A(x) - \rho_B(x)|\ dx}{\int_{\mathbb{R}^2} \max(\rho_A(x), \rho_B(x))\, dx}.$$

The reflection metric can be generalised to finite complexes of $d-1$ dimensional algebraic hyper-surface patches in d dimensions. For this, we refer to [14]. Here, we focus at the computation of the reflection metric for finite unions of segments in the plane.

In Section 2, we emphasised the fact that the reflection metric is robust for deformation, blur, cracks, and noise. This property can be derived from the definition without much difficulty. Given a fixed viewpoint, the change in the area of the visibility star caused by each of the above effects is proportional. The area of the reflection-visibility star, $\rho_A(x)$, changes at most twice as much as the area of the visibility star. This pointwise behaviour of the ρ_A-function is preserved as we integrate it, showing robustness of the reflection metric.

Now, we apply the results from the previous sections to compute the reflection metric. We assume A and B are unions of n_A and n_B line segments, respectively. having at most k edges in their visibility graphs. The reflection metric can be rewritten as follows:

$$d_{\mathtt{R}}(A,B) = \frac{2\int_{\mathbb{R}^2} |r(x)|\ dx}{\int_{\mathbb{R}^2} |p(x)|\ dx + \int_{\mathbb{R}^2} |q(x)|\ dx + \int_{\mathbb{R}^2} |r(x)|\ dx},$$

where $p(x) = \rho_A(x)$, $q(x) = \rho_B(x)$, and $r(x) = p(x) - q(x)$. The functions p, q and r are piecewise rational functions in two variables. With piecewise we mean that there is a finite number of triangles covering the support of the function, such that the restriction of the function to each such triangle is a rational function in two variables. The functions p and q are quotients of polynomials of degrees $O(n_A)$ and $O(n_B)$, respectively. The function r is a quotient of polynomials having degree $O(n_A + n_B)$. We adopt a model of computation in which the absolute value of a rational function in two variables, can be integrated over a triangular domain in $\Theta(d)$ time, where d is the maximum degree of the polynomial numerator and denominator.

The computation of the integrals of the rational functions p, q and r proceeds as follows. Let k_A and k_B denote the number of visibility edges corresponding

to A and B respectively. First, we compute the visibility graphs of A and B, taking times $O(n_A \log(n_A) + k_A)$ and $O(n_B \log(n_B) + k_B)$, respectively. Using the algorithm sketched in Section 3, the event segments that correspond to the reflection visibility partition, can be found in $O(s_A \log(n_A))$ and $O(s_B \log(n_B))$ time, where $s_A = \Omega(k_A)$ and $s_B = \Omega(k_B)$ are the number of event segments for A and B, respectively. Then, we compute a trapezoidal decomposition for the union of both event segment collections in time $O((s_A + s_B) \log(n_A + n_B) + v)$, where v is the number of intersections. We integrate the absolute values of p, q, and r by summing the partial integrals over all trapezoids (each trapezoid is a union of two triangles). In our model of computation, this takes $\Theta(n_A + n_B)$ time for each trapezoid. Since the summation of partial integrals dominates the overall complexity, we arrive at the following result.

Theorem 7 *Let A and B each be unions of n_A and n_B segments, respectively. Using randomisation, the reflection distance $d_{\mathbb{R}}(A, B)$ can be computed in $O(r(n_A + n_B))$ time, where r is the complexity of the overlay of the reflection-visibility partitions of A and B.*

7 Conclusion

We presented a new metric for pattern matching, the reflection metric. This metric is invariant under the group affine transformations and can therefore be used for affine shape recognition. The reflection metric is robust for pattern-defects such as deformation, blur, cracks, and noise. It can be generalised to finite unions of algebraic hyper-surface patches in any dimension.

The reflection metric is defined in terms of reflection visibility. Trans-visibility and reflection-visibility are stronger than visibility. Reflection-visible points are always trans-visible, and trans-visible points are always visible. We analysed the partitions corresponding to the visibilities, starting at normal visibility, proceeding with trans-visibility, and ending with reflection-visibility. In both steps new types of events emerged, making the resulting partitions more complex.

Constructions show that the worst-case combinatorial complexity for each of the three types of partitions is $\Omega(n^4)$. Using randomised incremental construction, each of the three corresponding arrangements can be built in $O((n + k) \log(n) + v)$ time, where k is the number of visibility edges, and v is the number of intersections in the arrangement. The structure of reflection-visibility partitions can be used to compute the reflection metric for two collections of segments in $O(r(n_A + n_B))$ randomised time, where r is the complexity of the overlay of two reflection-visibility partitions.

References

[1] P. K. Agarwal and M. Sharir. On the number of views of polyhedral terrains. *Discrete and Computational Geometry*, 12:177–182, 1994.

[2] B. Aronov, L. J. Guibas, M. Teichmann, and L. Zhang. Visibility queries in simple polygons and applications. In *ISAAC*, pages 357–366, 1998.

[3] T. Asano, T. Asano, L. Guibas, J. Hershberger, and H. Imai. Visibility of disjoint polygons. *Algorithmica*, 1:49–63, 1986.

[4] P. Bose, A. Lubiw, and I. Munro. Efficient visibility queries in simple polygons. In *Proc. 4th Canadian Conference on Computational Geometry*, 1992.

[5] K. W. Bowyer and C. R. Dyer. Aspect graphs: An introduction and survey of recent results. *Int. J. of Imaging Systems and Technology*, 2:315–328, 1990.

[6] I. Chakravarty and H. Freeman. Characteristic views as a basis for three-dimensional object recognition. In *Proc. SPIE: Robot Vision*, pages 37–45, 1982.

[7] M. de Berg, D. Halperin, M. Overmars, and M. van Kreveld. Sparse arrangements and the number of views of polyhedral scenes. *Int. J. Computational Geometry & Applications*, 7(3):175–195, 1997.

[8] H. Everett. *Visibility graph recognition.* PhD thesis, Department of Computer Science, Univerity of Toronto, 1990.

[9] S. K. Ghosh. On recognizing and characterizing visibility graphs of simple polygons. *Discrete Computational Geometry*, 17(2):143–162, 1997.

[10] S. K. Ghosh and D. M. Mount. An output-sensitive algorithm for computing visibility graphs. *SIAM J. Computing*, 20:888–910, 1991.

[11] Z. Gigus, J. Canny, and R. Seidel. Efficiently computing and representing aspect graphs of polyhedral objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13:542–551, 1991.

[12] N. Grimal, J. Hallof, and D. van der Plas. Hieroglyphica, sign list. http://www.ccer.theo.uu.nl/hiero/hiero.html, 1993.

[13] L. J. Guibas, Rajeev Motwani, and Prabhakar Raghavan. The robot localisation problem. *SIAM J. Computing*, 26(4), 1997.

[14] M. Hagedoorn and R. C. Veltkamp. Measuring resemblance of complex patterns. In *Discrete Geometry for Computer Imagery*, 1999.

[15] D. J. Kriegman and J. Ponce. Computing exact aspect graphs of curved objects: Solids of revolution. *Int. J. of Computer Vision*, 5:119–135, 1990.

[16] D. T. Lee. *Proximity and Reachability in the plane.* PhD thesis, University of Illinois at Urbana-Champaign, 1978.

[17] Y. Lin and S. S. Skiena. Complexity aspects of visibility graphs. *Int. J. Computational Geometry & Applications*, 5:289–312, 1995.

[18] K. Mulmuley. *Computational Geometry: An introduction through randomized algorithms.* Prentice Hall, 1994.

[19] W. H. Plantinga and C. R. Dyer. Visibility, occlusion and the aspect graph. *Int. J. of Computer Vision*, 5:137–160, 1990.

[20] M. Pocchiola and G. Vegter. Topologically sweeping visibility complexes via pseudotriangulations. *Discrete Computational Geometry*, 16:419–453, 1996.

[21] E. Welzl. Constructing the visibility graph for n-line segments in $O(n^2)$ time. *Inform. Process. Lett.*, 20:167–171, 1985.