Experimental Results on Quadrangulations of Sets of Fixed Points

Prosenjit Bose* Suneeta Ramaswami† Godfried Toussaint‡ Alain Turki§

Abstract

We consider the problem of obtaining "nice" quadrangulations of planar sets of points. For many applications "nice" means that the quadrilaterals obtained are convex if possible and as "fat" or squarish as possible. For a given set of points a quadrangulation, if it exists, may not admit all its quadrilaterals to be convex. In such cases we desire that the quadrangulations have as many convex quadrangles as possible. Solving this problem optimally is not practical. Therefore we propose and experimentally investigate a heuristic approach to solve this problem by converting "nice" triangulations to the desired quadrangulations with the aid of maximum matchings computed on the dual graph of the triangulations. We report experiments on several versions of this approach and provide theoretical justification for the good results obtained with one of these methods. The results of our experiments are particularly relevant for those applications in scattered data interpolation which require quadrangulations that should stay faithful to the original data.

1 Introduction

Finite element mesh generation is a problem that has received considerable attention in recent years, due to its numerous applications in a number of areas such as medical imaging, computer graphics, and Geographic Information Systems (GIS). Important applications also arise in the manufacturing industry, where a central problem is the simulation of processes such as fluid flow in injection molding, by solving systems of partial differential equations [11]. To make such tasks easier, the classical approach is to use the method of *finite elements*, where the bounding surface of the relevant object is partitioned into small pieces by using data points sampled on the object

*School of Computer Science, Carleton University, Ottawa, Ontario, Canada. Research supported by NSERC and FIR. This work was carried out while the author was at the Department of Computer Science, University of British Columbia. Email: `jit@scs.carleton.ca`

†Department of Computer Science, Rutgers University, Camden, NJ 08102, USA. Research supported by NSERC. This research was carried out while the author was at the School of Computer Science, McGill University. Email: `rsuneeta@camden.rutgers.edu`

‡School of Computer Science, McGill University, 3480 University Street, Montreal, Quebec, Canada. Research supported by NSERC. Email: `godfried@cs.mcgill.ca`

§IBM, Toronto, Canada. This research was carried out while the author was at the School of Computer Science, McGill University. Email: `alaint@ca.ibm.com`

surface. Each piece of the object (as given by the data points) defines a surface patch and these patches are then "stitched" together to form an approximation to the object surface. In *scattered data interpolation*, the goal typically is to construct a bivariate function that fits the sampled data, which is measured from the object surface. There are a variety of methods that yield functions with particular properties, depending on the application in mind. In these applications, the constructed mesh is required to stay faithful to the original sampled data.

The generation of a *mesh* from the sampled data points is, therefore, the first step in applications such as those mentioned above. For several decades the favored mesh has been the triangular mesh (or triangulation of the data points [15]), in which the finite elements are triangles. Consequently, triangulations of sets (such as sets of points, line segments, polygons, etc.) are well-studied and much is known about them [5]. However, in some situations, it is preferable that the finite elements be *quadrangles* (quadrilaterals) instead of triangles. For example, it has recently been shown that quadrangulations have several advantages over triangulations for the problem of scattered data interpolation [13, 24, 25, 26] and that improvements in elasticity analysis can be obtained in finite element methods by using quadrangles rather than triangles [2]. Moreover, convex quadrangles, while not always required, are preferred in certain applications [26, 28]. Note that the terminology "quadrangle" and "quadrangulation", which is standard in the literature on scattered data interpolation [13, 24, 25, 26], is equivalent to "quadrilateral" and "quadrilateralization". Quadrangle and quadrilateral are used interchangeably in the mathematics literature as well [14].

Unfortunately, not much is known about quadrangulations of point sets and good quadrangular meshes are harder to generate than good triangular meshes [22]. One of the reasons for this is that a quadrangulation of a set may not always exist, in which case additional points that are not part of the original input, called *Steiner points*, may need to be added. In other words, if edges are allowed to be inserted only between points of the given input set, then not all sets of objects (such as sets of points, simple polygons or polygons with points inside) admit a quadrangulation. For a survey on this topic see [38]. Engineers have developed a large body of mesh generation software for quadrangulations (see [33] for a comprehensive collection of references and pointers to currently available software). However, much of this work applies more to solid modeling, where the number of Steiner points is not an issue and a quadrangulation may add as many vertices as necessary to model an object. The focus of our work is on the construction of quadrangulations that modify the original data as little as possible, a requirement in some scattered data interpolation applications. We should point out that many interpolation schemes use neither triangles nor quadrilaterals, and among those that do, specific criteria for placing Steiner points may be specified. These criteria typically depend not only on geometry, but also on interpolatory criteria such as curvature and other derivative information. See [19] for a discussion of such issues.

The characterization of quadrangulations of sets of objects and the design of algorithms for their efficient computation using a small number of Steiner points have only just begun [6, 9, 10, 28, 29, 30, 36]. A set of points admits a quadrangulation without Steiner points if and only if the number of points on the convex hull is even. This was first shown in [39] and may have been known by Euler [12] (it was later rediscovered in [9, 10]). In [36], the authors show that there are simple polygons that may require $\Omega(n)$ Steiner points to be quadrangulated, and give efficient algorithms for quadrangulating an arbitrary simple polygon with a bounded number of Steiner points. In [28, 29, 30], the authors give complexity results and algorithms for convex quadrangulations of degenerate convex polygons, and related problems. In [6], the authors give an algorithm for quadrangulating a simple polygon using quadrilaterals with no large angles, and using a linear number of Steiner points. Note that the problem of determining if a set of points admits a convex quadrangulation without Steiner points is open [38].

Keeping the number of Steiner points as small as possible is an important consideration, since Steiner points change the measured data and increase the size of the input. It is worth pointing out that this view of Steiner points (as modifying the measured data) does not apply to solid modeling, where any number of Steiner points may be added to model the topology of the object under consideration. For scattered data interpolation, the measured data is sacred and hence Steiner points are seen as modifying the original data. However, minimizing the number of Steiner points is rarely the only requirement in practice. A number of other features of the quadrangular finite elements are very important as well. For instance, in a large number of applications, it is critical that the quadrangles be convex [28]. Furthermore, it is also often desirable that the quadrangles be *fat* (intuitively speaking, a fat quadrangle is one that is close to being a square); there are numerous ways of measuring fatness, some of which we will discuss later on in this paper. It is difficult to optimize all of these features simultaneously. For example, convex quadrangulations will require more Steiner points than if the requirement of convexity is removed. One of the goals of this paper is to give an experimental evaluation of this trade-off for random point sets. In particular, we construct quadrangulations that add as few Steiner points as possible and evaluate it for features such as convexity and fatness. On the other hand, we construct quadrangulations that have as many convex quadrangles as possible and assess the number of Steiner points such a quadrangulation may need. These experiments are particularly relevant to the area of scattered data interpolation.

The remainder of the paper is organized as follows. All definitions and notation appear in Section 2. In Section 3, we provide the relevant background on converting triangulations to quadrangulations via *matchings* in dual graphs of triangulations. We then discuss, in Section 4, the motivation behind our experiments. All the experimental results on quadrangulations of point sets appear in Section 5. Our heuristic approach gives very good results with regard to various quality measures for quadrangulations; we discuss in Section 6 the theoretical justification for some of our results.

2 Preliminaries and Definitions

In this section, we introduce relevant terminology and definitions, both geometric and graph-theoretic. These are standard and for more details on the former, we refer the reader to [32, 35] and for the latter, to [1, 8, 34].

A *graph* $G = (V, E)$ consists of a finite set of *vertices* (or *nodes*) V and a set of *edges* E, given by pairs of vertices from V. We consider only *undirected* graphs here, i.e., the pairs of vertices in E are unordered. A *weighted* graph is simply a graph whose edges have weights associated with them. A *planar* graph G is a graph that can be embedded (drawn) in the plane such that each vertex of G is a point on the plane, each edge of G is a simple curve between two vertices and no two edges of G intersect except possibly at the vertices. It is a well-known fact [17] that every planar graph

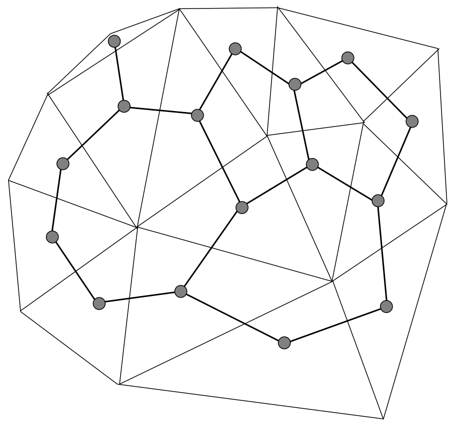

Figure 1: A triangulation of a set of points, and its dual

that has a planar embedding such that all edges are drawn as straight-line segments. Such a planar embedding is often referred to as a *planar subdivision*. A planar subdivision partitions a plane into several connected regions; the closures of such regions are called the *faces* of the graph.

Consider a planar subdivision which has the property that every face is classified in one of three ways: an *outer face*, an *object face* or *a hole*. The outer face is the only unbounded face. Bounded faces that do not belong to the object are called holes. By a *triangulation*, we mean a planar subdivision in which every object face is a triangle and every edge of the subdivision belongs to at least one object face. From now on, when we use the phrase “triangle of the triangulation”, we refer exclusively to an object face of the triangulation. A *quadrangulation* is defined in a manner similar to a triangulation, except that every object face is a quadrangle (quadrilateral). The *dual graph* of a triangulation is the graph in which there is a node for every triangle of the triangulation, and an edge between two nodes if the corresponding two triangles share a side.

A planar *convex* region R is a subset of the plane such that for any two points in R, the closed line segment connecting the two points lies entirely in R. Given a set S of points in the plane, the *convex hull* of S is the boundary of the smallest convex region containing S. A *triangulation of a set of points* S is a triangulation (without holes) that has the set S as the set of vertices and whose outer face is bounded by the convex hull of S. See Figure 1 for an example triangulation of a set of points, as well as its dual (shown in bold).

Given a graph $G = (V, E)$ (possibly weighted) with V as the set of nodes and E as the set of edges, a *matching* M on G is a subset of edges from E such that no two of them have a common node. The *maximum cardinality matching* of G is a matching of maximum size. The *maximum weight matching* in a weighted graph is a matching of maximum weight. A *perfect matching* is a matching of size $\lfloor |V|/2 \rfloor$ i.e., every node in V (except possibly one in the case that $|V|$ is odd) belongs to an edge of the matching.

3 Background

As mentioned in the introduction, the study of quadrangular mesh construction has started relatively recently, whereas computing triangulations is a very well-studied topic. Due to this fact, practitioners often use the approach of *converting* triangulations to quadrangulations [21, 23, 36, 37]. Several of these methods, however, are heuristic, may require a large number of Steiner points and often do not give precise bounds on the required number of Steiner points.

There are many different ways to obtain quadrangulations from triangulations by adding Steiner points (see [36] for some examples). We will assume that a *given* triangulation is to be converted into a quadrangulation, which implies that diagonals between a pair of points from the input set can only be deleted and not added. The idea behind obtaining a quadrangulation in this fashion is to pair up neighboring triangles to form quadrangles and one of the goals of the conversion is to add as few Steiner points as possible. In other words, we would like to find the maximum possible number of such pairings. This corresponds precisely to the problem of finding the maximum cardinality matching in the dual graph of the triangulation. It follows that a triangulation admits a quadrangulation without Steiner points if and only if the dual graph of the triangulation admits a perfect matching [36]. If the dual graph does not have a perfect matching, the resulting quadrangulation will have some leftover triangles.

This relation between quadrangulations and matchings in dual graphs was used in [36] to give efficient algorithms for quadrangulating simple polygons with a bounded number of Steiner points. In this paper, we will use this relation to design numerous experiments for constructing quadrangulations of sets of points. Furthermore, we also give tight upper bounds on the number of leftover triangles when using this approach to convert an *arbitrary* triangulation into a quadrangulation.

Our experiments, which will be described in detail in the following section, were conducted on three different types of triangulations. These are the *serpentine* triangulation, a triangulation constructed by a divide-and-conquer technique and the *Delaunay* triangulation. The serpentine triangulation was selected because the dual graph of such a triangulation always contains a perfect matching, which results in quadrangulations with no leftover triangles. The triangulation constructed by divide-and-conquer was selected because in constructing such a triangulation, the point set is equitably subdivided in a recursive manner. The Delaunay triangulation was selected because it possesses several desirable properties; for example, it is the triangulation that maximizes the minimum angle and satisfies the empty circle property (see [31]). In a related paper on quadrangulating planar point sets [10], the authors use two types of triangulations that have perfect matchings, serpentine and *hamiltonian* [4], to show that the former triangulation results in better quadrangulations. Our paper extends these results by demonstrating that if we start with a triangulation with good properties, using maximum matchings results in good quadrangulations with few leftover triangles, even if the dual graph is not guaranteed to have perfect matchings.

In the following, we provide brief descriptions of the three triangulations used in our experiments.

- A *serpentine* (or *spiral*) triangulation [9, 10] of a set of points is defined as follows: First construct a convex spiral from the set of points S as follows. Start with a vertical half-ray

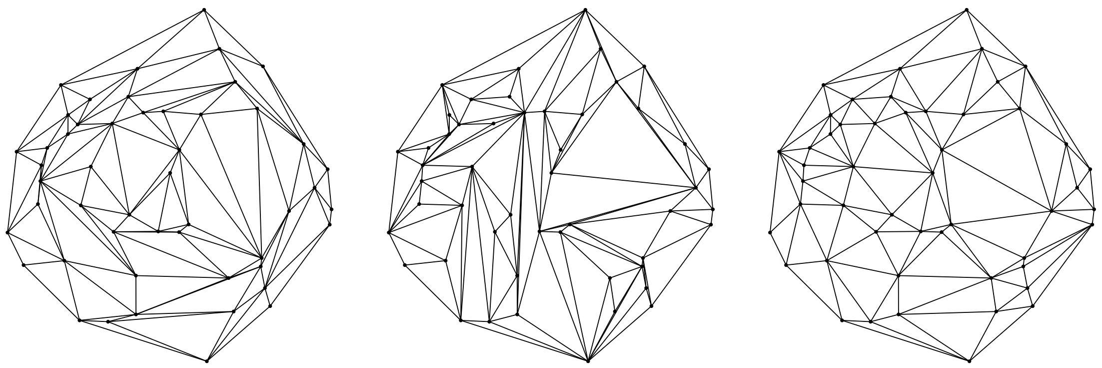

Figure 2: (a) Serpentine triangulation (b) HVP-triangulation (c) Delaunay triangulation

placed at a leftmost point on the convex hull of S, mark the point, and rotate the half-ray in the clockwise direction until it hits an unmarked point. Mark this point, which is the second point of the spiral chain. We now have a new direction for the half-ray. To find the third point on the spiral chain, repeat the procedure with the new half-ray anchored at the second point of the spiral chain, and so on until all the points in S are marked. Spiral chains are closely related to convex layers (also known as the *onion peeling* of a set), and one can be found from the other in linear time [35]. Once the spiral chain has been constructed, the triangulation is computed using a rotating calipers method. The serpentine triangulation can thus be constructed in $O(n \log n)$ time (see [9] for details). The serpentine triangulation of 50 points is shown in Figure 2(a).

- The triangulation constructed by divide-and-conquer is built as follows. First, the point set is partitioned into two halves by a vertical line. Each half is triangulated recursively where the dividing line alternates between horizontal and vertical. In the merge step, the two triangulated halves are "stitched" up along the dividing line. This method of partitioning the points is the same as the partitioning method used in the construction of *kd-trees*. Since these triangulations are constructed by repeated horizontal and vertical partitionings, we shall henceforth refer to them as *HVP-triangulations*. See Figure 2(b) for an example of such a triangulation of the same set of 50 points used in Figure 2(a).

- The *Delaunay* triangulation of a set of points possesses many desirable properties and can be constructed in many different ways (see [31]). Some useful properties include the empty-circle property (i.e. the circumcircle of a triangle in the Delaunay triangulation does not contain any other points in the point set), and the maxmin-angle property (i.e. the Delaunay triangulation of a point set is the triangulation that maximizes the value of the minimum angle). The Delaunay triangulation is often the triangular mesh of choice among practitioners. See Figure 2(c) for an example of the Delaunay triangulation of the same 50 points used in Figure 2(a).

Observe that the serpentine triangulation dual has a perfect matching because, by construction, the dual contains a hamiltonian path. The duals of the other two triangulations will not, in general, have perfect matchings.

4 Motivating the Experiments

Now that we have established the relevant definitions, we can describe the motivation behind the set of experiments that we conducted. The basic idea behind all the experiments is to construct a quadrangulation of a point set by finding matchings in the dual graph of a triangulation. Note that the quadrangulation may have leftover triangles. Our experiments were designed to evaluate two important requirements in quadrilateral meshes: there should be as few leftover triangles as possible, and the quadrangles must be convex.

Computing maximum cardinality matchings minimizes the number of unmatched nodes. This implies that the corresponding quadrangulation contains the minimum number of leftover triangles. However, maximum cardinality matchings give us no guarantees on the shape of the quadrangle, and hence the quadrangulation may contain a large number of non-convex quadrangles. In order to find quadrangulations that contain many convex quadrangles, we compute the maximum *weighted* matching of the dual graph, where the weight of an edge is high if the quadrangle corresponding to that edge is convex.

The following variables determined each experimental run:

Number of points Point sets of size 50, 100, 200, 500, and 1000 were used.

Method used to randomly generate the point set The points were generated randomly using three different distributions: uniform on a disk, uniform inside a square, and Gaussian (or standard normal).

Triangulation algorithm Three types of triangulations were used: Delaunay triangulation, HVP-triangulation, and serpentine triangulation. These are described in Section 3. To compute the Delaunay triangulation of a point set, we used Steve Fortune's implementation of his $O(n \log n)$ time algorithm [18]. The `C` source code is publicly available at AT&T Bell Labs. `C` code for the algorithms to compute the HVP and serpentine triangulations were developed by us.

Matching algorithm Maximum cardinality matching, and maximum weighted matching algorithms were applied. The best known algorithm for computing the maximum cardinality matching of a graph with vertex set V and edge set E is an $O(|E|\sqrt{|V|})$ algorithm given by Micali and Vazirani [27]. This algorithm was implemented in Pascal by Steven Crocker for the implementation challenge held by DIMACS in 1985, and is available by anonymous ftp from `dimacs.rutgers.edu`. Source code for maximum weighted matching was developed by Edward Rothberg for the same implementation challenge, and was also available at the same site (`/pub/netflow/matching/weighted/solver-1/`). This code is an implementation of Gabow's $O(n^3)$ algorithm [20] for computing the maximum weighted matching of an n-node graph.

All other code, such as the driver, graphics, and statistics code necessary for the experiments, was developed by us. Each trial run, as determined by the above variables, was run one hundred times in order to collect statistical data on the following quality measures for the quadrilateral mesh generated:

- the number of leftover triangles
- the number of convex quadrangles
- the average fatness of a quadrangle

The third quality measure requires further discussion. There are several ways to measure the fatness of a quadrangle; for example, some standard ways are via the maximum angle of the quadrangle (the smaller it is, the more fat the quadrangle), or the minimum angle (the larger it is, the more fat the quadrangle). However, these two measures alone do not adequately reflect the fatness of a quadrangle, since of two quadrangles with exactly the same minimum and maximum angles, one may be fat and the other long and skinny (for example, a square and a rectangle with the long edge much longer than the short one). Hence we use one more measure of fatness, namely, the ratio of the maximum distance and the minimum distance between any two vertices of the quadrangle. This number will vary from $\sqrt{2}$ to arbitrarily large values. One weakness of this measure of fatness is that it is unbounded, and influenced by how close the vertices of the quadrangle are. This results in a lot of variance over many quadrangles. Therefore, we choose to find the median, rather than average, value of this measure.

5 Experimental Results

5.1 Maximum Cardinality Matchings

In this section, we show the results obtained by running the maximum cardinality matching algorithm on the Delaunay, serpentine, and HVP-triangulations. Recall that this approach seeks to minimize the number of leftover triangles in the resulting quadrangulation. All graph plots of data obtained from this set of experiments appear in Appendix A.

Figure 3 shows the quadrangulation obtained from the **Delaunay triangulation** of 500 points. The graphs in Figure 9 (Appendix A) show the fraction of quadrangles in the resulting quadrangulation that are convex (left), and the fraction of triangles of the triangulation that are matched (right). The variance is indicated by vertical bars. Observe that near-perfect matchings are obtained for the larger data sets.

Figure 10 contains the graphs of the mean maximum angle (left), and the mean minimum angle (right) of the quadrangles in the quadrangulation. Observe that the quadrangles are convex on average and the variance is small. The median maximum ratio of the distances between pairs of vertices of a quadrangle is shown in Figure 11. The quadrangulation performs well in terms of this quality measure as well.

Figure 4 shows the quadrangulation obtained from the **serpentine triangulation** of 500 points. The graphs in Figure 12 show the fraction of quadrangles in the resulting quadrangulation that are convex (left), and the fraction of triangles of the triangulation that are matched (right). Recall that since serpentine triangulation duals have perfect matchings, which explains the graph on the right hand side. However, the fraction of convex quadrangles is much smaller than for Delaunay

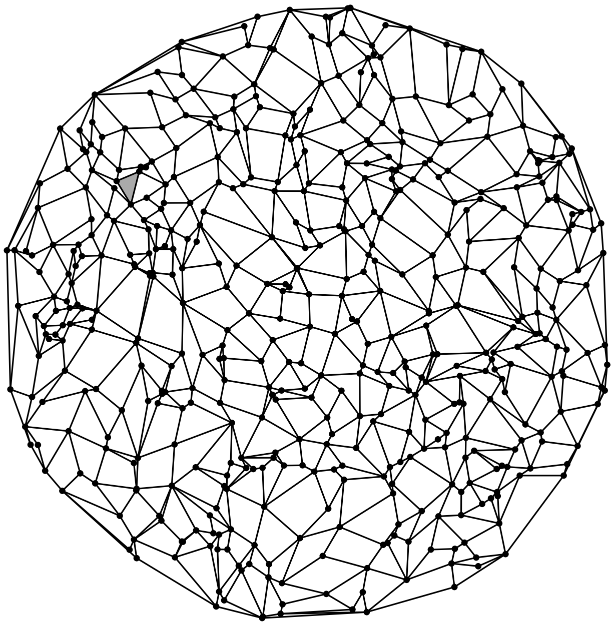

Figure 3: Quadrangulation obtained by computing a maximum cardinality matching on the Delaunay triangulation dual. Unmatched triangles are shaded.

triangulations, a fact that is clearly visible in the difference between Figures 3 and 4. The mean maximum angle, and the mean minimum angle of the quadrangles in the quadrangulation are shown in Figure 13. Again, the results are significantly worse than for Delaunay triangulations, with the mean maximum angle being greater than 180 degrees for larger data sets, and the mean minimum angle being smaller than 25 degrees. The serpentine triangulation also performs poorly in terms of the maximum ratio fatness measure, as the graph in Figure 14 shows.

Finally, Figure 5 shows the quadrangulation of 500 points, obtained from the **HVP-triangulation**. This triangulation has the worst performance in terms of convexity of the quadrangles, as is apparent from the graph in Figure 15 (left). However, we obtain near-perfect matchings in this case as well. This triangulation also has the worst results for the average angle measures, with the maximum angle being greater than 189 degrees, and the minimum angle being less than 23 degrees on average. However, for the maximum ratio fatness measure (Figure 17), this quadrangulation performs better than the one obtained from serpentine triangulation.

5.2 Maximum Weighted Matchings

In this section, we show the results obtained by running the maximum weighted matching algorithm on the Delaunay, serpentine, and HVP-triangulations. Recall that this approach seeks to maximize the number of convex quadrangles in the resulting quadrangulation. All graph plots of data obtained from this set of experiments appear in Appendix B.

Figure 6 shows the quadrangulation obtained from the Delaunay triangulation of 500 points. Of all our experimental runs, maximum weighted matchings on Delaunay triangulations gave us

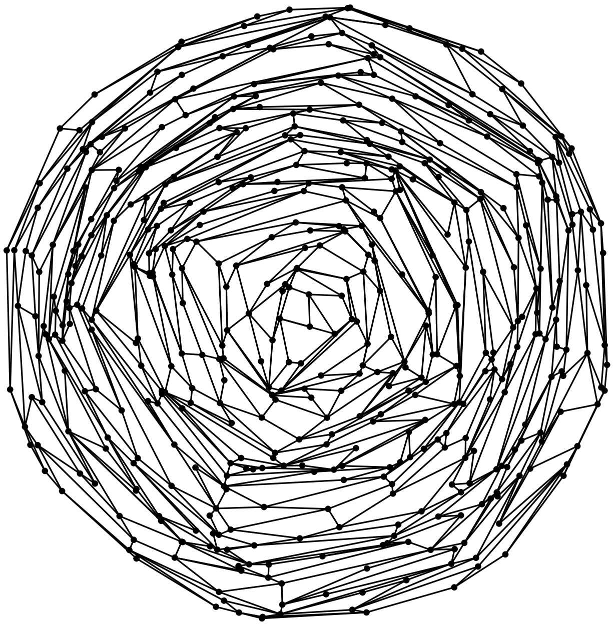

Figure 4: Quadrangulation obtained by computing a maximum cardinality matching on the Serpentine triangulation dual. Unmatched triangles are shaded.

the best overall results in terms of the number of leftover triangles, as well as element quality. As Figure 18 (Appendix B) clearly shows, the number of leftover triangles is not very high, even though the weight was biased entirely in favor of convex quadrangles (a weight of 10,000 was used for dual graph edges that represent convex quadrangles). Furthermore, Figures 19 and 20 show that all fatness measures indicate a high-quality quadrangulation as well, with mean maximum angle less than 145 degrees, mean minimum angle greater than 40 degrees, and a fairly low maximum inter-vertex distance ratio.

Figure 7 shows a quadrangulation obtained from the serpentine triangulation of 500 points. It can be seen that the number of leftover triangles is quite high. Since the matching is biased entirely in favor of dual graph edges that give convex quadrangles, this indicates that in serpentine triangulations, the fraction of pairs of triangles that form convex quadrangles is relatively low. Furthermore, the element quality is quite poor as well, with all fatness measures being generally worse than those obtained for Delaunay triangulations.

Finally, Figure 8 shows the quadrangulation obtained from the HVP-triangulation of 500 points. The results for the fraction of matched triangles, and the fraction of convex quadrangles is slightly worse than those for serpentine triangulations (Figure 24). However, the maximum and minimum angle fatness measures, shown in Figure 25, are slightly better than those for the serpentine triangulation , and Figure 26 shows that the distance ratio measure is significantly better for HVP-triangulations.

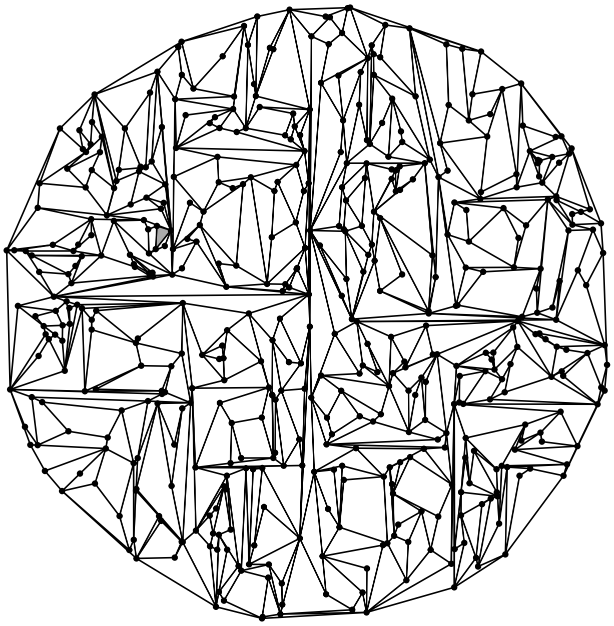

Figure 5: Quadrangulation obtained by computing a maximum cardinality matching on the HVP-triangulation dual. Unmatched triangles are shaded.

5.3 Summary of Results

The following table summarizes the results for the case of 1000 points. The maximum weighted matching results shown in the table use a weight of 10,000 (1) for dual graph edges representing convex (non-convex) quadrangles.

Triangu-lation	**Matching**	**Percent matched**	**Percent convex**	**Mean max angle**	**Mean min angle**	**Median ratio**
Delaunay	Cardinality	100	82	2.7	.72	3.4
	Weighted	99	98	2.5	.78	3.4
HVP	Cardinality	100	47	3.3	.36	4.6
	Weighted	95	81	2.9	.47	4.6
Serpentine	Cardinality	100	61	3.2	.21	6.7
	Weighted	97	90	3.0	.27	6.6

Table 1: Summary of results for 1000 points

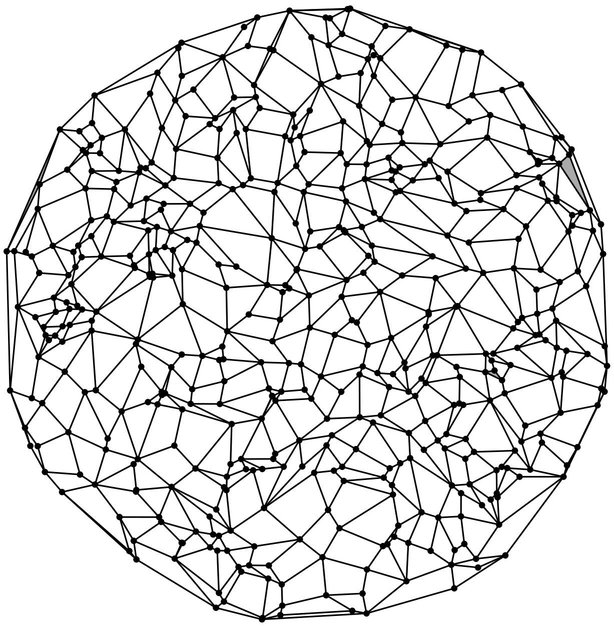

Figure 6: Quadrangulation obtained by maximum weighted matching on the Delaunay triangulation dual. Unmatched triangles are shaded.

6 Analysis of Results on Matchings

One of the most striking results in our experimental work is the fact that the maximum cardinality matching gave perfect, or near perfect, matchings in all the cases. While this is not surprising for serpentine triangulations (whose duals admit a hamiltonian path, as pointed our earlier), it is unexpected for the Delaunay and HVP-triangulations. However, we are able to explain this phenomenon. We show that the number of leftover triangles in a maximum cardinality matching of the dual graph of *any* triangulation of a point set depends only on the number of convex hull vertices. For randomly generated point sets in a disc or square, this number is relatively small [16].

In fact, the number of unmatched nodes in a maximum cardinality matching of the dual graph of a triangulation of a point set cannot exceed a third of the number of convex hull vertices. We prove this bound below.

A *k-edge coloring C* of a graph G is an assignment of one of k colors to each edge of G such that no two adjacent edges have the same color. Note that each color class of edges in a k-edge coloring of a graph represents a matching. This link will be exploited in order to bound the maximum number of unmatched triangles.

Given a point set P, let $T(P)$ represent an arbitrary triangulation of P. Every face of $T(P)$ is a triangle except possibly the outer face. Let CH represent the number of vertices on the outer face of $T(P)$, i.e. the number of vertices on the convex hull.

Let $T^*(P)$ represent the dual graph of $T(P)$. Let MT^* represent a maximum matching in $T^*(P)$. Let MT represent the edges in $T(P)$ that are dual to the edges MT^* in $T^*(P)$ but are *not* on the

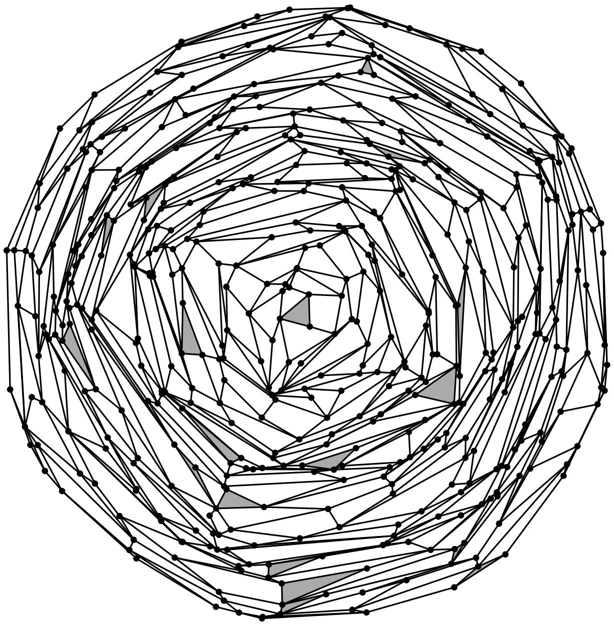

Figure 7: Quadrangulation obtained by maximum weighted matching on the Serpentine triangulation dual. Unmatched triangles are shaded.

convex hull of P.

Lemma 6.1 *Removal of the edges MT from $T(P)$ results in a graph where every internal face is either a triangle or a quadrangle. If a face is a triangle, then the vertex v^* in the dual $T^*(P)$ representing the triangle is unsaturated (i.e. is not adjacent to an edge in the matching MT^*) or v^* is saturated but adjacent to the vertex in $T^*(P)$ representing the external face.*

Proof: Consider an edge e^* of MT^* whose dual edge is not on the convex hull of P. Let the two vertices adjacent to e^* be v_i^* and v_j^*. These two vertices correspond to two triangles, t_i and t_j, respectively, in $T(P)$. The two triangles in $T(P)$ share an edge e that is the dual of e^*. Removal of the edge e from $T(P)$ results in a quadrangle. None of the 4 edges forming this quadrangle are removed since no other edge of MT^* is adjacent to v_i^* or v_j^*. Therefore, the removal of an edge in MT from $T(P)$ can only create a quadrangle.

Since all of the faces in $T(P)$ are originally triangles, the lemma follows.

■

Notice that reducing the number of unsaturated vertices in the matching MT^* of the dual $T^*(P)$ reduces the number triangles remaining after the edges MT are removed from $T(P)$. The fewer triangles that remain, the fewer Steiner points are needed to complete the quadrangulation. This crucial link will be exploited in order to convert a triangulation into a quadrangulation.

We need to establish a relation between the number of unsaturated vertices in $T^*(P)$ and the number of Steiner points that must be added to convert $T(P)$ into a quadrangulation once the edges of MT are removed. This link is established by studying maximal planar graphs.

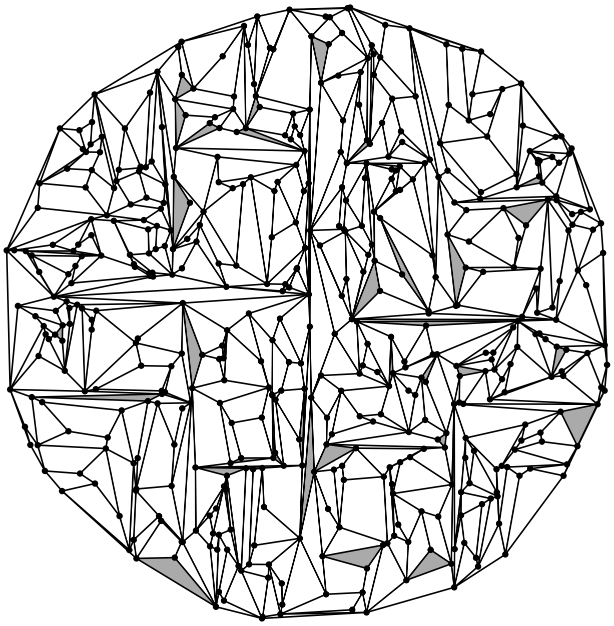

Figure 8: HVP-triangulation: Quadrangulation obtained by maximum *weighted* matching on the HVP-triangulation dual. Unmatched triangles are shaded.

Definition 1 *A maximal planar graph is a connected planar graph in which every face (including the external face) is a triangle.*

By adding CH - 3 edges between vertices on the convex hull, $T(P)$ can be converted into a maximal planar graph G. Since the vertex set of G and $T(P)$ is the same, by Euler's formula for planar graphs, G has CH - 3 more faces than $T(P)$. By considering properties of the dual G^* of G, we will bound the number of Steiner points added in order to quadrangulate P through the removal of edges from $T(P)$.

The dual G^* of G is a 3-regular, bridgeless graph.

Lemma 6.2 *The dual G^* of a maximal planar graph G is 3-regular and bridgeless (i.e. contains no cut edge).*

Proof: Since every face in G is a triangle, every vertex in G^* has degree three.

Suppose that G^* had a cut edge e^*. Note that the faces adjacent to a vertex v in G form a cycle in G^*. Let edge e with end points a and b in G be the dual of edge e^*. The faces adjacent to a form a cycle C in G^*. However, e^* is in cycle C contradicting the fact it is a cut edge.

■

Theorem 6.1 *Every bridgeless, 3-regular planar graph is 3-edge-colorable.*

A proof of the above theorem appears in [8] where it is shown to be equivalent to the four-color theorem [3]. Theorem 6.1 implies that the edges of a bridgeless, 3-regular planar graph can be partitioned into three sets such that each set is a perfect matching. Therefore, we conclude with the following.

Theorem 6.2 *The number of unmatched nodes in a maximum cardinality matching of the dual graph of a triangulation of a point set cannot exceed a third of the number of convex hull vertices.*

7 Concluding Remarks

Intuition seems to suggest that if you start with a good triangulation, the strategy of pairing up triangles to form quadrangles should lead to good quadrangulations. The experiments conducted support this intuition. As the data in Table 1 clearly shows, Delaunay triangulations consistently give the best results in terms of element quality. In particular, the maximum weighted matching algorithm run on the Delaunay triangulation dual has the best overall output with 99% of the triangles matched, 98% of the resulting quadrangles convex, mean maximum angle of 2.5 radians (143.2 degrees), mean minimum angle of .78 radians (44.7 degrees), and a maximum inter-vertex distance ratio median of 3.4. Note that the weighted serpentine seems to be a close second. However, the minimum angle and median ratio for these quadrangulations are much worse than for the Delaunay. Even though the weighted serpentine triangulation has a high number (90%) of convex quadrangles, these quadrangles are long and skinny, which is undesirable in practice.

References

[1] R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows: Theory, Algorithms and Applications*. Prentice Hall, Englewood Cliffs, NJ, 1993.

[2] D. J. Allman. A quadrilateral finite element including vertex rotations for plane elasticity analysis. *International Journal for Numerical Methods in Engineering*, 26:717–730, 1988.

[3] K. Appel and W. Haken. Every planar map is 4-colorable. *Ill Journal of Mathematics*, 21:429 – 567, 1977.

[4] E. Arkin, M. Held, J. Mitchell and S. Skiena. Hamiltonian triangulations for fast rendering. *Algorithms-ESA '94*, J. van Leeuwen, ed., Utrecht, NL, LNCS 855, 36-47.

[5] M. Bern and D. Eppstein. Mesh generation and optimal triangulation. In F. K. Hwang and D.-Z. Du, editors, *Computing in Euclidean Geometry*. World Scientific, 1992.

[6] M. Bern and D. Eppstein. Quadrilateral meshing by circle packing. *6th Int. Meshing Roundtable*, Park City, Utah, pages 7-19, 1997.

[7] P. Bose, T. Shermer, G. Toussaint, and B. Zhu. Guarding Polyhedral Terrains. *Computational Geometry: Theory and Applications*, 7:173 – 186, 1997. Also available as Technical Report 92.20, McGill University, 1992.

[8] A. Bondy and U. S. R. Murty. *Graph Theory with Applications*. Elsevier Science, 1976.

[9] P. Bose and G. T. Toussaint. No quadrangulation is extremely odd. In *Proc. of the International Symposium on Algorithms and Computation*, pages 372–281, Cairns, Australia, December 4-6 1995.

[10] P. Bose and G. T. Toussaint. Characterizing and efficiently computing quadrangulations of planar point sets. *Computer Aided Geometric Design*, 14:763-785, 1997.

[11] J. Bown. *Injection Moulding of Plastic Components*. McGraw-Hill, New York, 1979.

[12] W. G. Brown. Historical note on a recurrent combinatorial problem. *American Mathematical Monthly*, 2:973-977, 1965.

[13] C. K. Chui and M.-J. Lai. Filing polygonal holes using C^1 cubic triangular spline patches. *Computer Aided Geometric Design*, 17:297–307, 2000.

[14] H. S. M. Coxeter. *Projective Geometry* Springer-Verlag 1987.

[15] L. De Floriani, B. Falcidieno, and C. Pienovi. Delaunay-based representation of surfaces defined over arbitrarily shaped domains. *Computer Vision, Graphics and Image Processing*, 32:127–140, 1985.

[16] L. Devroye, Expected time analysis of algorithms in computational geometry. *in Computational Geometry*, Editor, G. T. Toussaint, North-Holland, pages 135-151, 1985.

[17] I. Fáry. On straight line representation of planar graphs. *Acta Sci. Math. Szeged*, 11:229–33, 1948.

[18] S. Fortune. A sweepline algorithm for Voronoi diagrams. *Algorithmica*, 2:153-174, 1987.

[19] R. Franke. Scattered Data Interpolation: Tests of Some Methods. *Mathematics of Computation*, 38(157):181-200, 1982.

[20] H.N. Gabow. Implementation of Algorithms for Maximum Matching on Nonbipartite graphs. Ph.D. Thesis, Stanford University, 1973.

[21] E. Heighway. A mesh generator for automatically subdividing irregular polygons into quadrilaterals. *IEEE Transactions on Magnetics*, 19(6):2535–2538, 1983.

[22] B. Joe. Quadrilateral mesh generation in polygonal regions. *Computer Aided Design*, 27(3):209–222, March 1995.

[23] B. P. Johnston, J. M. Sullivan, and A. Kwasnik. Automatic conversion of triangular finite meshes to quadrilateral elements. *International Journal of Numerical Methods in Engineering*, 31(1):67–84, 1991.

[24] M-J. Lai. Scattered data interpolation and approximation using bivariate C^1 piecewise cubic polynomials. *Computer Aided Geometric Design*, 13:81–88, 1996.

[25] M-J. Lai. Convex preserving scattered data interpolation using bivariate C^1 cubic splines. *J. Comput. Applied Math.*, 119:249–258, 2000.

[26] M. J. Lai and L. L. Schumaker. Scattered data interpolation using C^2 supersplines of degree six. *SIAM Journal on Numerical Analysis*, 34(3):905-921, 1997.

[27] S. Micali and V. Vazirani. An $O(\sqrt{|V|}|E|)$ algorithm for finding a maximum matching in general graphs. *Proc. 21st Annual IEEE Symp. on Foundations of Comp. Sci.*, pages 17–27, 1980.

[28] M. Muller-Hannemann and K. Weihe, Minimum strictly convex quadrangulations of convex polygons. *Proc. 13th ACM Symposium on Computational Geometry* pages 193-200, 1997.

[29] M. Muller-Hannemann and K. Weihe, Improved approximations for minimum-cardinality quadrangulations of finite-element meshes. *Proc. 5th European Symposium on Algorithms (ESA'97)*, Springer Lecture Notes in Computer Science 1284, pages 364-377, 1997.

[30] M. Muller-Hannemann and K. Weihe, On the discrete core of quadrilateral mesh refinement. *International Journal on Numerical Methods in Engineering*. Vol. 46, pages 593-622, 1999.

[31] A. Okabe, B. Boots, K. Sugihara, S-N. Chin. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Chichester, John Wiley, 2000.

[32] J. O'Rourke. *Computational Geometry in C*. Cambridge University Press, 1994.

[33] S. Owen. A survey of unstructured mesh generation technology. `http://www.andrew.cmu.edu/user/sowen/survey/index.html`.

[34] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall Inc., 1982.

[35] F. Preparata and M. I. Shamos. *Computational Geometry*. Springer-Verlag, 1985.

[36] S. Ramaswami, P. Ramos, and G. T. Toussaint. Converting triangulations to quadrangulations. *Computational Geometry: Theory & Applications*, 9:257-276, 1998.

[37] V. Srinivasan, L.R. Nackman, J-M. Tang, and S.N.Meshkat. Automatic mesh generation using the symmetric axis transformation of polygonal domains. *Proceedings of the IEEE (Special Issue on Computational Geometry)*, 80(9):1485–1501, 1992.

[38] G. Toussaint. Quadrangulations of planar sets. *Proceedings of 4th International Workshop on Algorithms and Data Structures (WADS '95)*, invited paper, August 16-18, pages 218-227, 1995.

[39] N. von Fuss. Solutio quaestionis, quot modis polygonum n laterum in polygona m laterum per diagonales resolvi queat. *Nova Acta Sciientiarum Imperialis Petropolitanae*, vol. 9, 1791.

A Graphs of results from maximum cardinality matchings

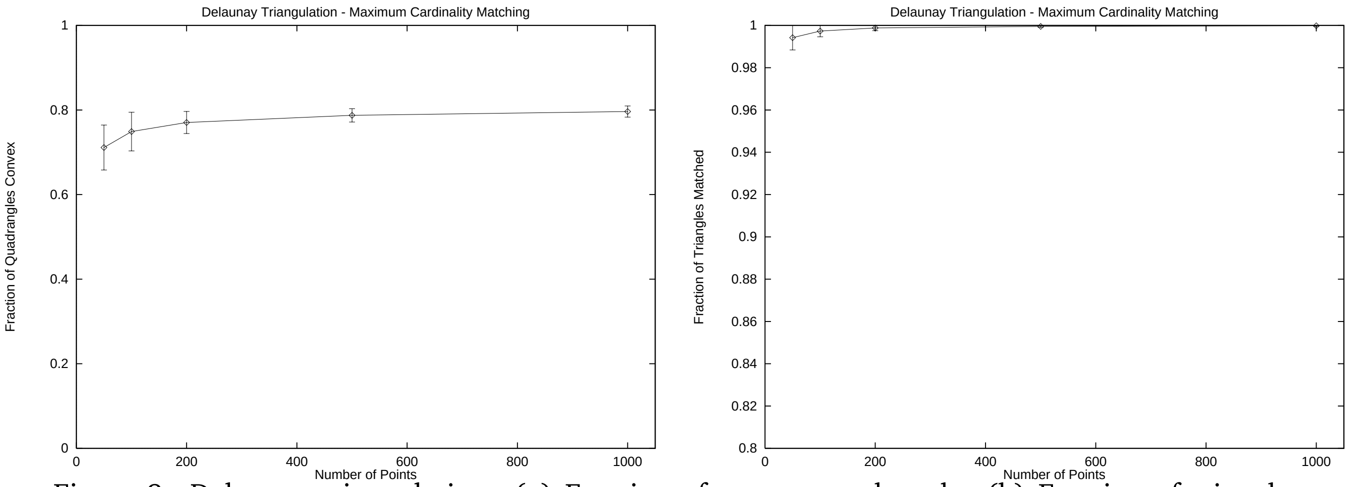

Figure 9: Delaunay triangulation: (a) Fraction of convex quadrangles (b) Fraction of triangles matched

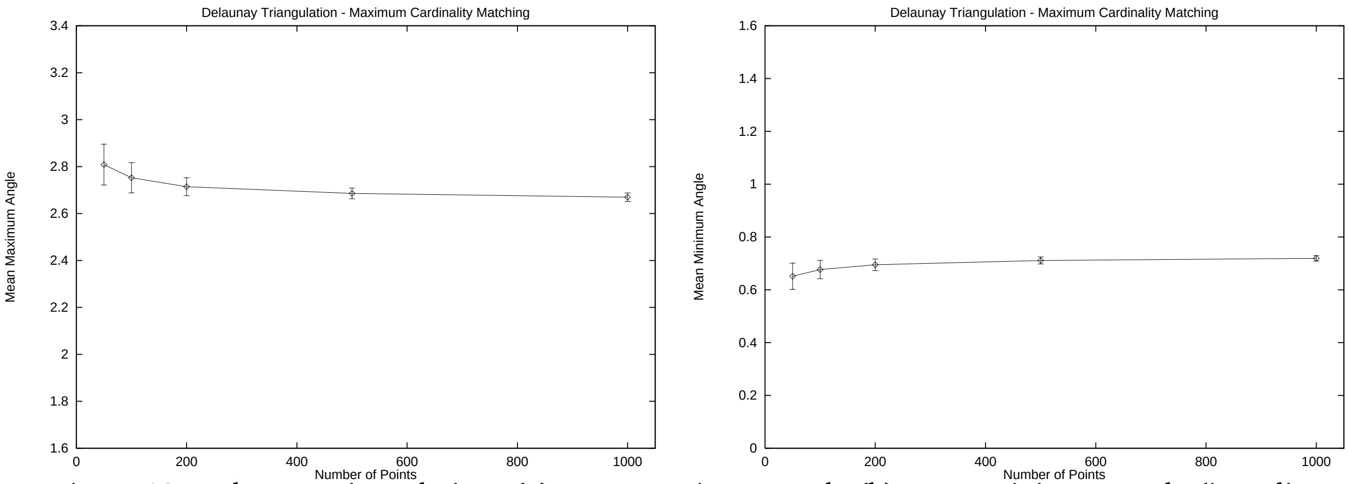

Figure 10: Delaunay triangulation: (a) Mean maximum angle (b) Mean minimum angle (in radians)

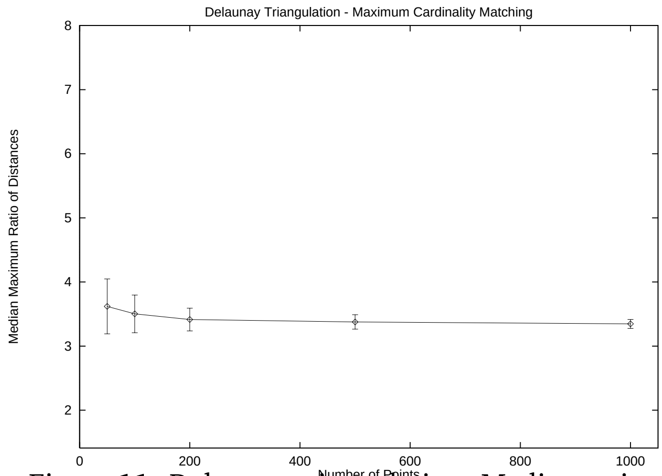

Figure 11: Delaunay triangulation: Median ratio

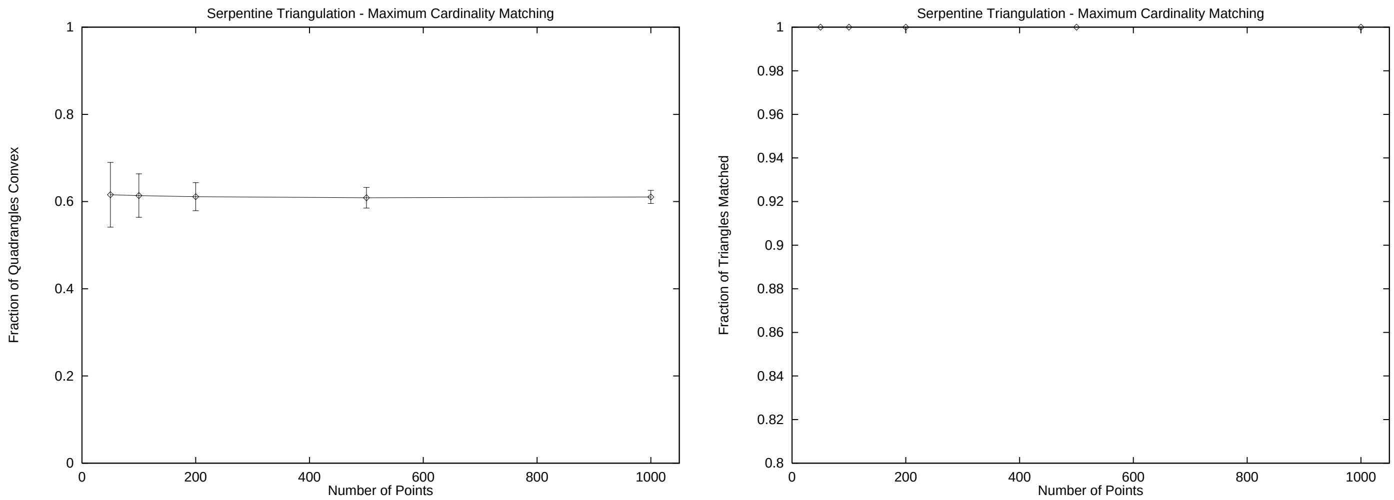

Figure 12: Serpentine triangulation: (a) Fraction of convex quadrangles (b) Fraction of triangles matched

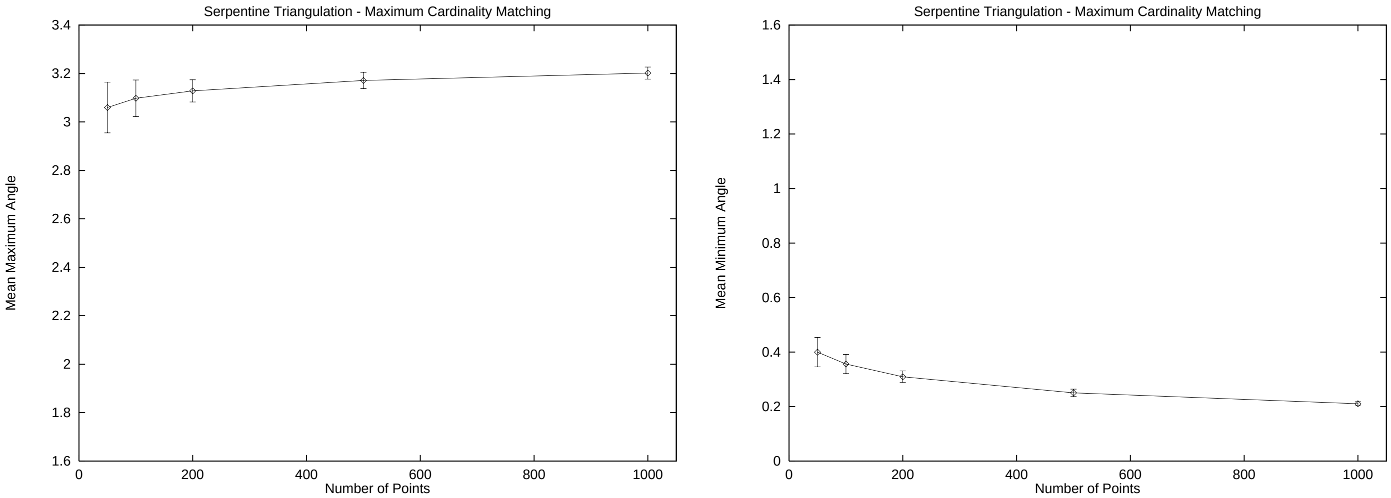

Figure 13: Serpentine triangulation: (a) Mean maximum angle (b) Mean minimum angle (in radians)

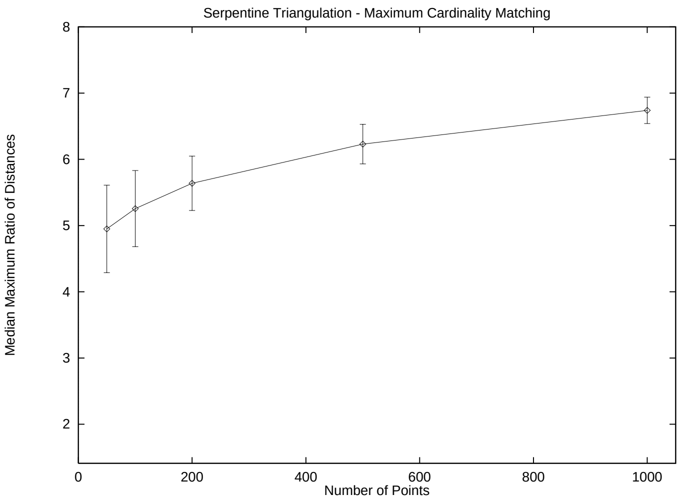

Figure 14: Serpentine triangulation: Median ratio

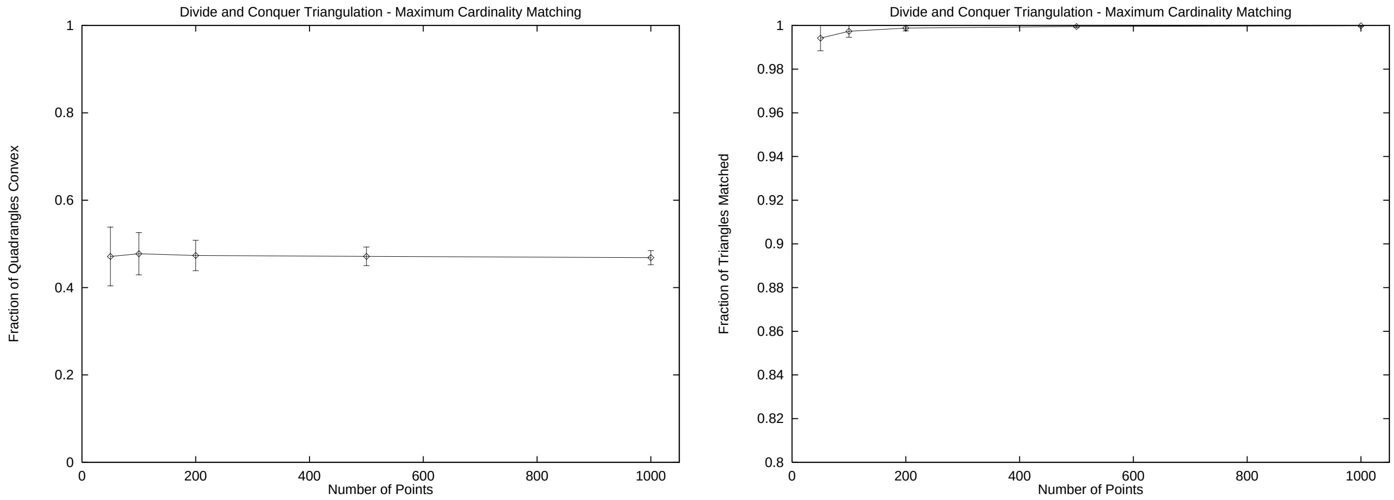

Figure 15: HVP-triangulation: (a) Fraction of convex quadrangles (b) Fraction of triangles matched

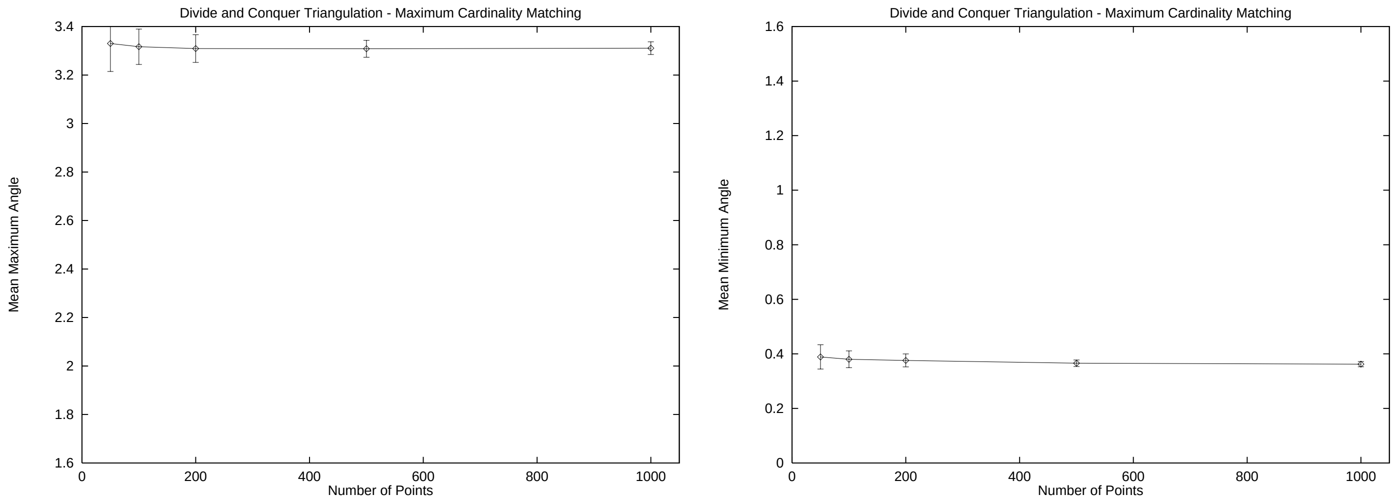

Figure 16: HVP-triangulation: (a) Mean maximum angle (b) Mean minimum angle (in radians)

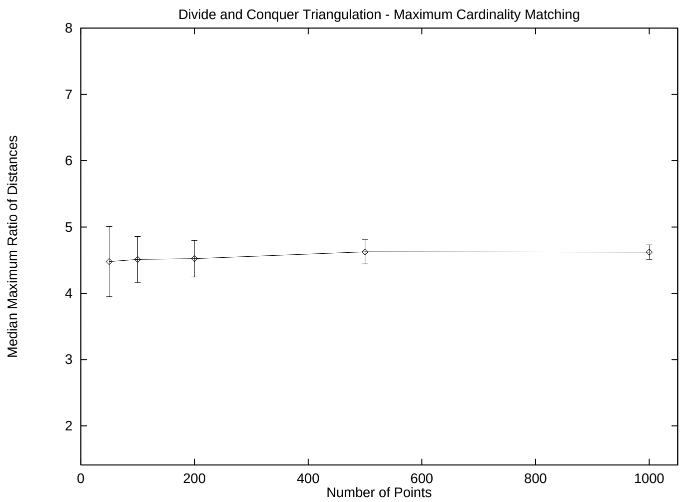

Figure 17: HVP-triangulation: Median ratio

B Graphs of results from maximum weighted matchings

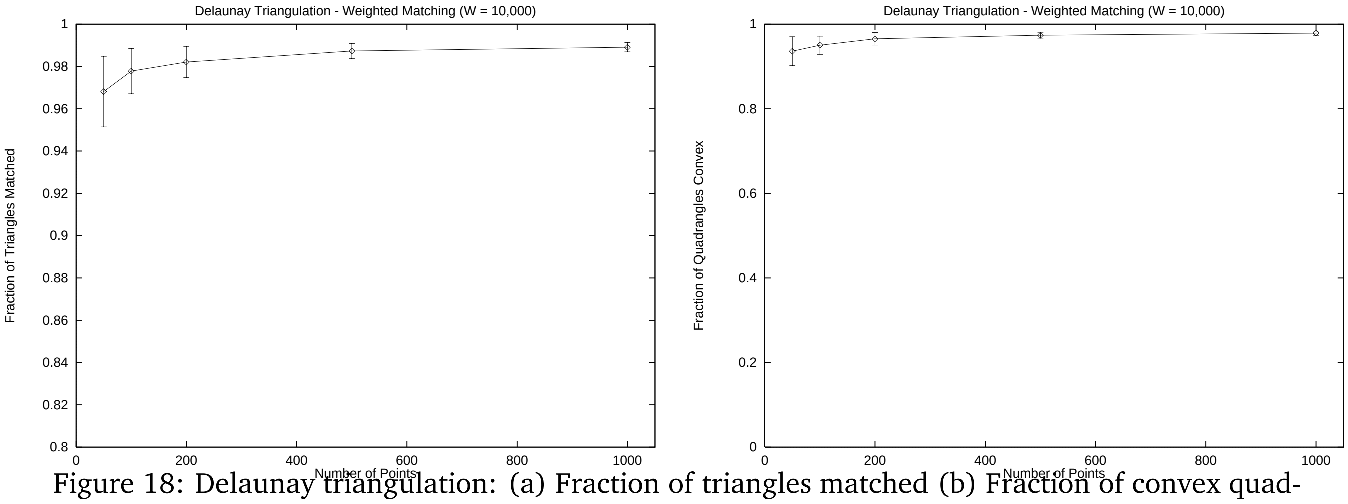

Figure 18: Delaunay triangulation: (a) Fraction of triangles matched (b) Fraction of convex quadrangles

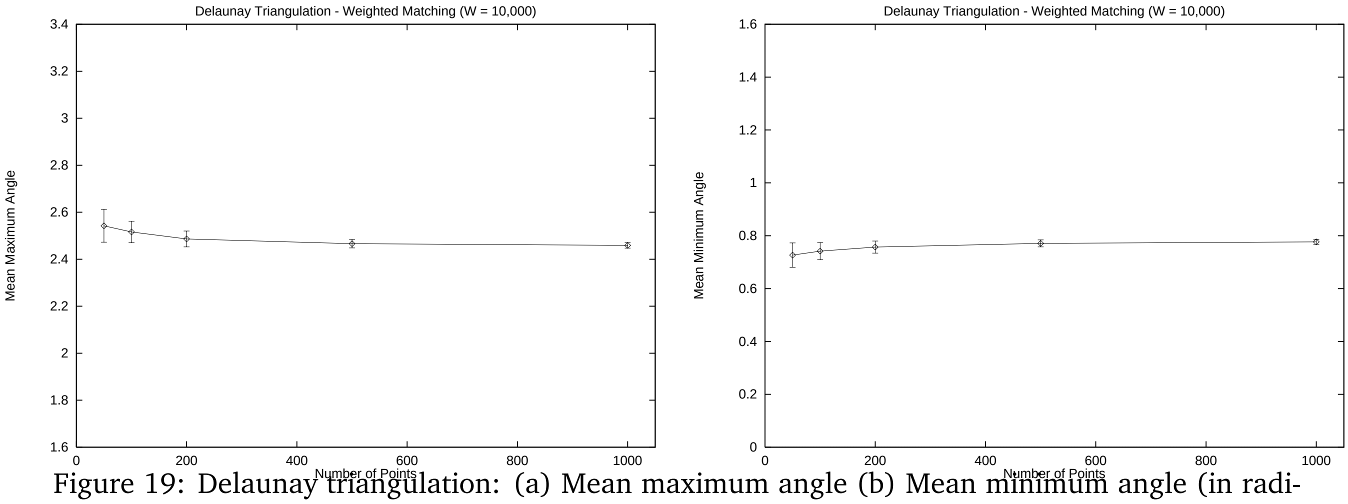

Figure 19: Delaunay triangulation: (a) Mean maximum angle (b) Mean minimum angle (in radians)

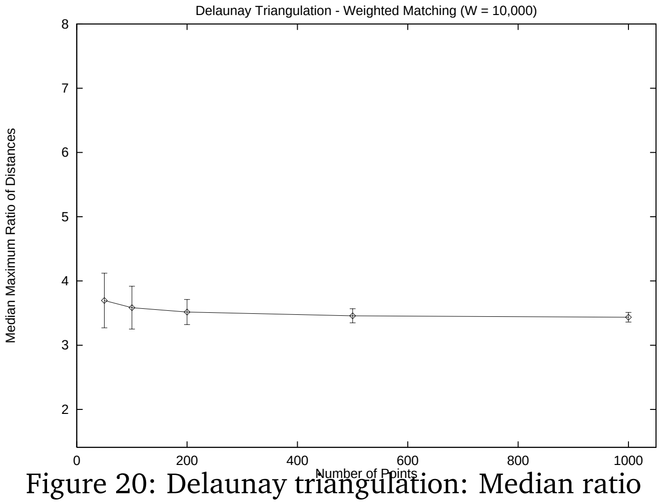

Figure 20: Delaunay triangulation: Median ratio

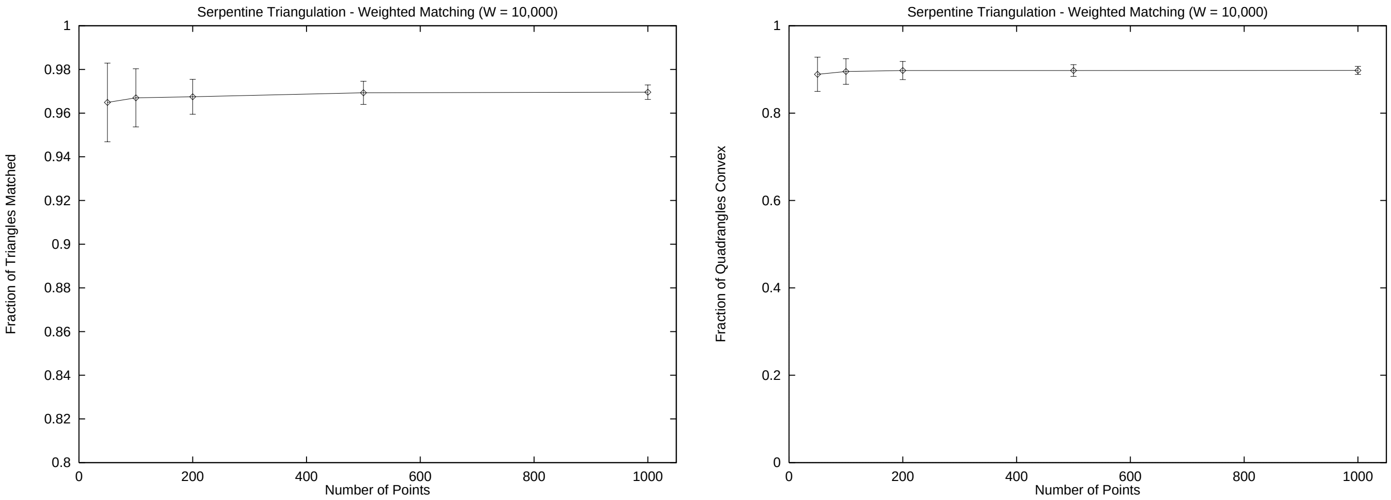

Figure 21: Serpentine triangulation: (a) Fraction of triangles matched (b) Fraction of convex quadrangles

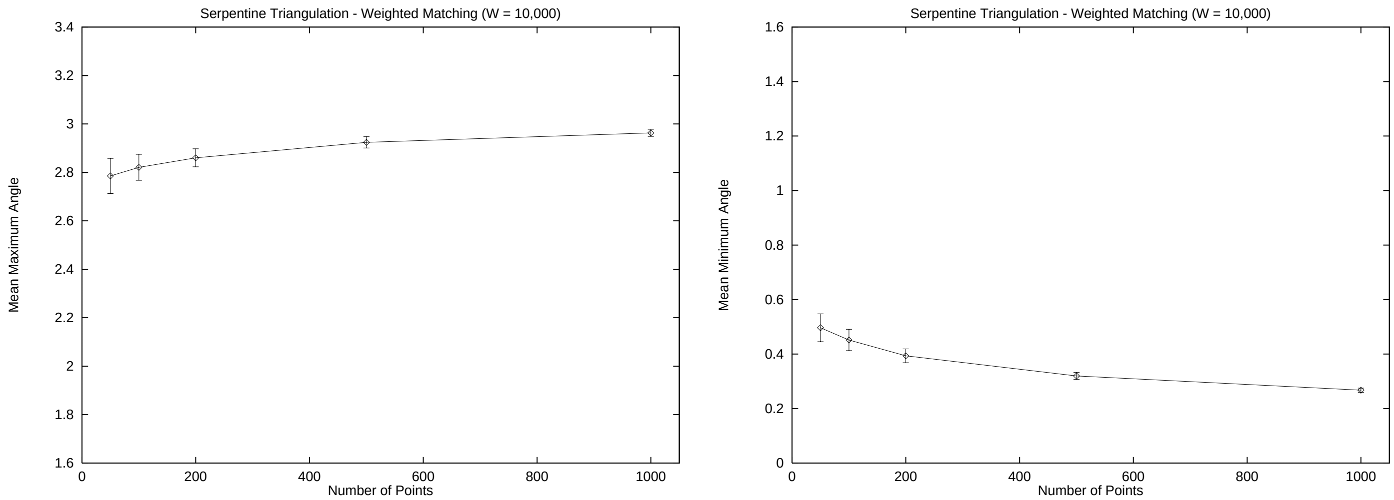

Figure 22: Serpentine triangulation: (a) Mean maximum angle (b) Mean minimum angle (in radians)

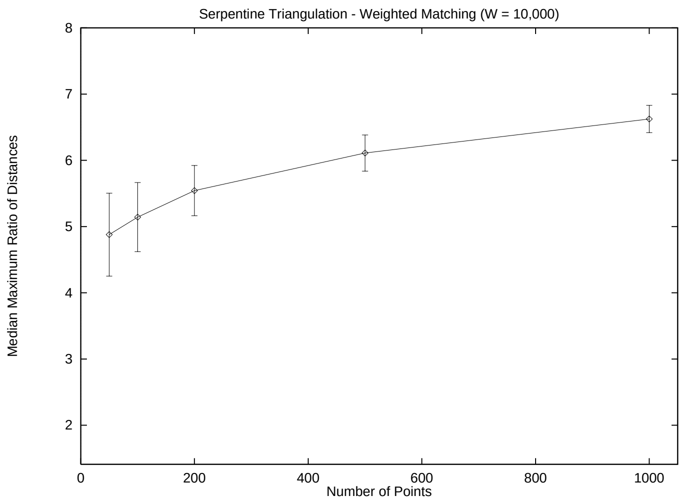

Figure 23: Serpentine triangulation: Median ratio

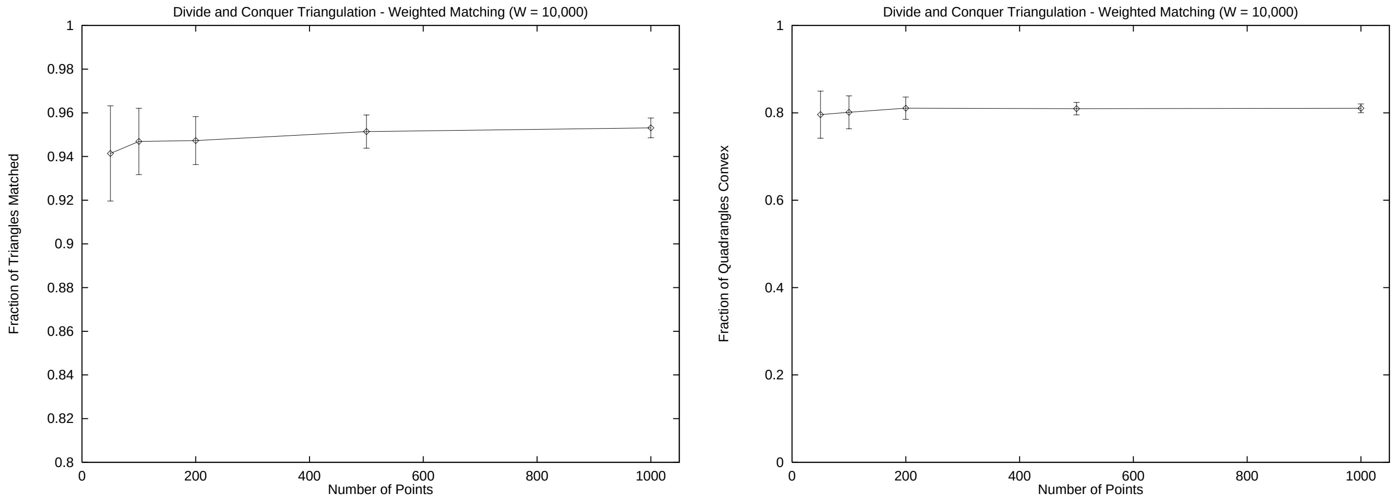

Figure 24: HVP-triangulation: Fraction of (a) triangles matched (b) convex quadrangles

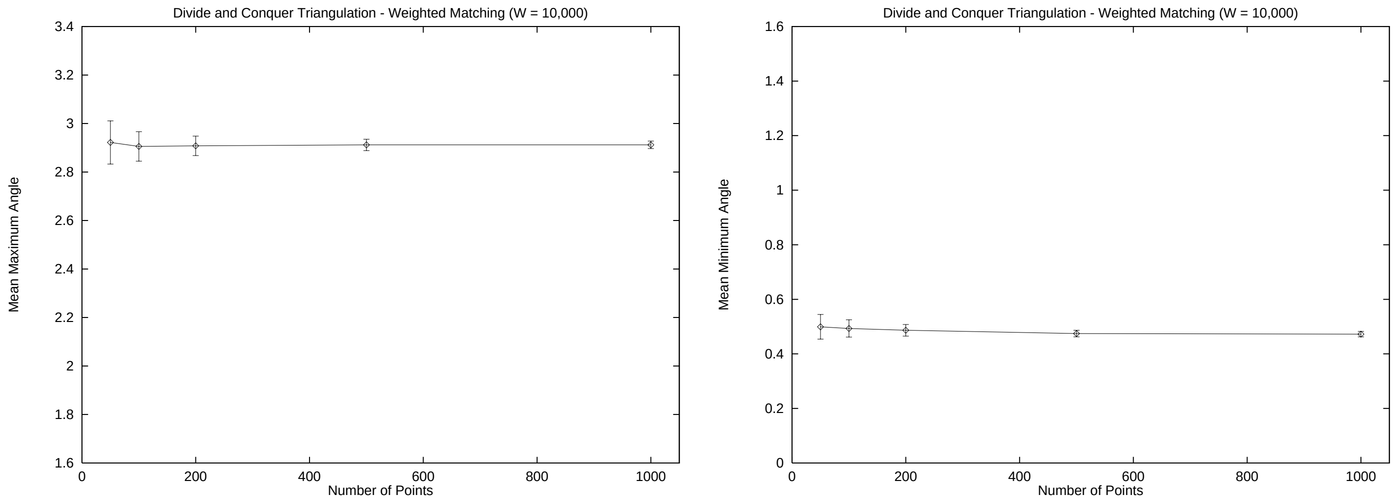

Figure 25: HVP-triangulation: (a) Mean maximum angle (b) Mean minimum angle (in radians)

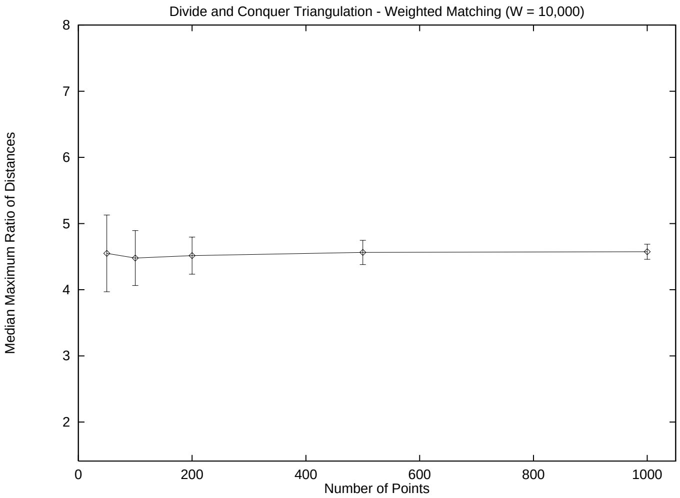

Figure 26: HVP-triangulation: Median ratio