CS 294-5
Meshing and Triangulation in Graphics, Engineering, and Modeling (Autumn 1999)
Examination 2
(20% of final grade)

This is an open-book exam. It is due at midnight the evening of Wednesday, December 15, 1999. You may submit your answers by email, under my office door, or through the U. S. Postal Service. If you choose the latter, your submission should be postmarked no later than December 15. Answers submitted after the deadline will **not** be accepted. You may consult papers and other references, but you may not receive assistance from other people (except me). You are welcome to ask me to clarify anything in the readings, the lectures, or this exam that you don't fully understand.

There are 28 points worth of questions below. Answer exactly 20 points worth. If you answer questions totalling more than 20 points, I will randomly select answers to ignore, bringing the total down to 20 points. If I ignore a question that you answered correctly and grade one that you got wrong, tough luck.

[1] Colliding fronts in paving (1 point). A simple way to check for colliding fronts in an advancing front triangular mesh generator is to maintain a constrained Delaunay triangulation of the void where vertices have not yet been generated. Why can't this idea be used in the paving algorithm without difficulty?

[2] Quadrilateral doublets (1 point). A *doublet* in a quadrilateral mesh is a pair of elements that share two adjacent sides. A doublet is considered topologically invalid, because at least one of the two quadrilateral elements must have an angle of 180° or more in any planar embedding. Demonstrate that, by adding twist chords to the dual of the mesh, you can (topologically) eliminate all doublets from a mesh.

[3] Rivara refinement (1 point). Is the second Rivara refinement algorithm discussed by Bern and Plassmann guaranteed to terminate if applied to refining the 9-simplices of a 9-dimensional simplicial mesh? Explain your answer.

[4] Paving and smoothing (1 point). When smoothing is applied after mesh generation, it affects only the geometry, and not the topology, of the mesh. However, the paving algorithm uses smoothing during meshing, and not just as a postprocess. Does smoothing affect the paving algorithm's choice of where to place the next element? Explain your answer.

[5] Constrained Delaunay tetrahedra (2 points). Let X be a three-dimensional PLC, and let t be a tetrahedron whose vertices are in X. Suppose that no vertex of X lies in t except the four vertices of t, and no segment of X lies in the interior of t. Suppose that each of the four faces of t is either constrained Delaunay or is contained in a constraining facet of X. Prove that t is constrained Delaunay.

[6] Element-nested quadrilateral mesh refinement (2 points). The following statement is true for any triangular mesh M: "Given any element e in M, it is possible to refine e while maintaining element-nesting by subdividing no more than c elements of M, where c is a constant independent of the mesh." (We can accomplish this by bisecting e and a neighbor sharing a common edge.)

Is the same statement true of any quadrilateral mesh M? Explain your answer.

[7] Subfacet splitting in Delaunay refinement (2 points). The three-dimensional Delaunay refinement algorithm discussed in the lecture notes generates elements whose circumradius-to-shortest edge ratios are no greater than two. To achieve this bound, the algorithm requires that encroached subfacets not be split in an arbitrary order. Instead, a subfacet may be split only if the orthogonal projection of an encroaching vertex lies in the subfacet. If encroached subfacets *are* split in an arbitrary order, what is the best bound on circumradius-to-shortest edge ratio that can (without extraordinary efforts) be proven using the same proof techniques employed in the lecture notes? Explain your answer.

[8] Minimum weight quadrilateralizations (2 points). Suggest how Step 5(a) of Montague and Dickerson's LMT-skeleton algorithm can be modified to help identify edges of the minimum weight quadrilateralization of a point set (whose convex hull has an even number of edges).

[9] Topological minimum weight triangulations (2 points). Given a weighted graph G whose nodes have *not* been assigned coordinates, consider the problem of finding a subgraph T of G whose total weight is minimum, under the constraint that T is the set of edges in some topological triangulation of the sphere, where the triangulation's vertices are the nodes of G.

Let's modify Dickerson and Montague's algorithm so that it attempts to eliminate some edges that are not in the topological minimum weight triangulation (TMWT). Topologically, there is no such thing as an "empty edge" or an "empty triangle," so every edge of G is initially in *candEdges*, and every triangle whose edges are in G is initially in *candTris*. There is no convex hull, so we skip Step 4 of Dickerson and Montague's LMT-skeleton algorithm.

A *certificate* of the edge $e = (u, v)$ is a pair of triangles $\triangle uvw, \triangle uvx \in$ *candTris* such that e is locally minimal with respect to $\triangle uvw$ and $\triangle uvx$: that is, either $(w, x) \notin G$ or the weight of (w, x) is greater than the weight of e. (Note that (u, v) and (w, x) are the diagonals of the quadrilateral $uwvx$.)

For each edge $e \; in$ *candEdges*, if e has no certificate, then remove e from *candEdges*, and remove any triangle that has e for an edge from *candTris*. Perform several passes through *candEdges*, stopping only after no edge is eliminated during an entire pass.

If a TMWT T of G exists, we would like to say that every edge of T is in *candEdges*. Unfortunately, the algorithm is flawed: it may mistakenly eliminate edges of T from *candEdges*. What is the flaw? Why doesn't Dickerson and Montague's algorithm for geometric MWTs have this flaw?

[10] Hierarchical point location in Delaunay triangulations (2 points). Devillers describes a point location method that uses a node-nested sequence of Delaunay triangulations. Suppose the finest triangulation has n nodes and thus $\mathcal{O}(n)$ triangles. Say that the *overhead* of Devillers' method is the amount of additional memory—above and beyond the storage space used to hold the finest triangulation—allocated for point location data structures. Devillers' point location method has $\mathcal{O}(n)$ expected overhead, and $\mathcal{O}(\log n)$ expected running time for one point location operation.

How can Devillers' method be modified so that its overhead is $o(n)$ (that is, asymptotically less than linear space), but its running time is still $\mathcal{O}(\log n)$? (Hint: change the sizes of the triangulations used for point location.)

[11] Good-quality surface simplification I (2 points). Surface simplification algorithms (Hoppe; Garland and Heckbert) aren't normally concerned with the skinniness of the triangles they produce. But if one wants to simplify a scanned model in preparation for tetrahedralizing it, one may wish to avoid surface triangles with excessively small or large angles.

How can spacing function based coarsening be integrated with surface simplification so that the simplified mesh is likely to have good-quality triangles? (Hint: don't compute a maximal independent set.)

[12] Good-quality surface simplification II (2 points). Another way we might simplify surfaces without creating skinny triangles is smoothing.

(a) Suppose that each time an edge is contracted, the resulting vertex is placed by optimization-based smoothing, with the goal of maximizing the minimum sine of the angles of the triangles incident on the vertex. Why is this a bad idea?
(b) Why is the final position of the smoothed vertex sensitive to the position it initially had before the optimization algorithm was applied?

[13] Quadric error metrics and the convex hull (2 points). If a triangulated surface (embedded in three dimensions) is not perfectly flat, its convex hull has volume, even if the surface does not enclose a volume. Let H be the convex hull of a triangulated surface. If we contract an edge in the surface, and use Garland and Heckbert's quadric error metrics to choose a location for the new vertex, is the vertex guaranteed to lie in H? Explain your answer.

[14] Another view of quadric error metrics (2 points).

(a) Explain how Garland and Heckbert's quadric error metrics can be used to identify the position of a vertex v that minimizes the sum of square distances to the vertices in v's preimage.

(b) This use of quadrics can be combined with Heckbert and Garland's use, so that the location of a vertex v depends on a weighted average of the sum of square distances to planes associated with v's preimage, and the sum of square distances to vertices in v's preimage. What problem associated with Garland and Heckbert's method is solved this way? (Hint: Think of a perfectly flat triangulated surface.)

[15] Dependencies in view-dependent meshes (2 points). Explain how a view-dependent progressive mesh can be inadvertently encoded so that the finest mesh must be reconstructed in its entirety before one (particular) small portion of it can be viewed. Why doesn't this unfortunate circumstance happen often in practice?

[16] Maximizing the area ratio (2 points). Let the *area ratio* of a triangulation be equal to the area of its largest triangle divided by the area of its smallest triangle. Given a simple polygon P, describe a cubic-time algorithm that finds a triangulation of P whose area ratio is at least as great as the area ratio of any other triangulation of P.