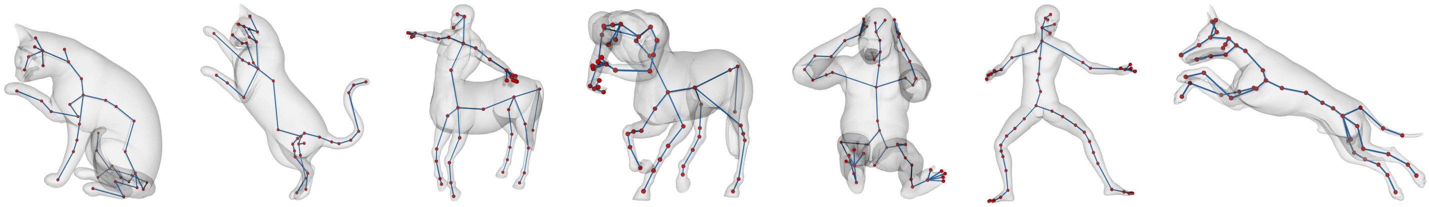
An Efficient PILP Algorithm for 3D Region Guarding and Star Decomposition

Wuyi Yu
Xiamen University

Xin Li
Louisiana State University



1 Introduction

We propose an efficient algorithm to compute 3D region guarding and star decomposition. A star decomposition partitions a 3D region to a set of sub-regions, each of which is visible from an interior point and is called a star shape. Star decomposition is closely related to the well known gallery guarding problem [Chvatal 1975]. Despite their broad applications in graphics and robotics, 3D region guarding and star decomposition are highly challenging and have been little explored.

On a geometric region M, the optimal guarding problem seeks a smallest set of points (guards) $\{g_i\} \in M$ to which the entire region $\forall p \in \partial M$ is *visible*. Solving optimal guarding problem is difficult. Given a specific 2D region M, finding optimal (fewest necessary) guards is shown to be NP-hard. [Lien 2007] presents a greedy algorithm for approximately guarding point cloud data. To our best knowledge, no other practical guarding computation algorithm has been developed for general 3D regions represented by polyhedra. We suggest an efficient optimization framework for this problem.

2 Our Approach

Given a 3D region M bounded by a triangle mesh $\partial M = \{V, F\}$, where V and F are sets of the vertices and triangles, we say a point q on ∂M is *visible* to a point $p \in M$, if the line segment $\overline{pq}$ connecting p and q lies totally inside M, namely, it only intersects ∂M on q: $\overline{pq} \bigcap \partial M = \{q\}$. For a 3D region M discretely represented by a tetrahedral mesh, from its candidate vertex set, we can formulate the *optimal gallery guarding* problem to cover the entire ∂M. Our intuition is to use medial axes (curve skeletons) as candidates because the skeleton usually has great visibility to the object boundary and hierarchical skeleton can be effectively computed to reduce the problem size. Therefore, we extract the shape skeletons of a sequence of progressively simplified meshes, then apply a multi-level optimization for efficient problem solving.

2.1 Visibility Detection

A face $f_j \in F$ is visible to a point p if all its vertices are visible. We need to detect the visible faces set of a given skeletal node. If a line segment $\overline{pv_i} \bigcap \partial M = \{q\}$, and $|\overline{pq}| < |\overline{pv_i}|$, then v_i is not visible from p. Enumerating $\overline{pv_i}$ for $\forall v_i \in V$ to check its intersections with all triangles in F takes $O(n_V \cdot n_T) = O(n^2)$ complexity. We develop a *sweep algorithm* using spherical coordinates to pre-sort all faces and only check ones incident to pv_i. This improves the efficiency of detecting the visible region of each p to $O(n \log n)$.

2.2 Optimization Algorithms for Guarding

When the visibility of all skeletal nodes is computed, picking a node set of minimum size that covers all boundary faces can be converted to a set-covering problem, which is also NP-complete.

A **greedy** solution for set-covering is to iteratively pick the skeletal node with the largest visible region on uncovered elements, remove covered faces from F, repeat this until F is empty. The greedy algorithm is efficient and its optimality is bounded by $O(\log C)$, C being the size of the optimal solution.

Table 1: *Runtime Table of Guarding Computation. $\#V$ is vertex number of ∂M, $\#G$ is number of computed guards, T is the computation time in seconds.*

Model	$\#V$	$\#G$			T(s)		
		ILP	Greedy	PILP	ILP	Greedy	PILP
Armadillo	20,002	–	38	30	–	590.8	601.4
Female	10,002	13	18	14	2,046.2	279.1	300.3
Male	10,002	14	16	15	3,074.3	312.6	330.8
Greek	9,994	15	22	18	4,122.4	307.4	312.9
David	9,996	16	22	17	107,391.2	245.1	248.2

An **optimal** solution can be computed by $0-1$ programming, also called Integer Linear Programming (ILP). For every skeleton point $p_i, i = 1, \ldots, m$, assign a variable x_i such that $x_i = 1$ if p_i is picked, and $x_i = 0$ if not. The *objective function* to minimize is then $\sum_{i=1}^{m} x_i$. To cover the entire ∂M, for $\forall j = 1, \ldots, n$, $f_j \in F$ is visible to a set of nodes $Q^j = \{p_{(j,1)}, \ldots, p_{(j,k)}\}$, then at least one node in Q^j should be chosen. So the *constraints* are $x_j = \{0, 1\}$, and $\sum_{p_k \in Q^j} x_k \geq 1, \forall j \in \{1, \ldots, n\}$. ILP gets the optimal guarding, but has exponential complexity.

We develop a progressive integer linear programming **(PILP)** framework for models with big sizes. We progressively simplify the boundary mesh ∂M into coarser resolutions. In the coarsest level, we solve the optimal guards by ILP. Then we progressively move to finer levels with more details: on each level, we map guards to the finer skeleton, ignore least significant guards, remove covered regions, and solve ILP again to find necessary new guards. As details increase, new guards are added until the finest resolution of ∂M is covered. PILP improves computational efficiency for several orders of magnitude over ILP, but still get very nice approximate optimization results (see Table 1). We perform experiments on large amount of models (see some in the teaser and more in the video).

2.3 Star Decomposition

We can use the optimal guarding points to compute star decomposition. Guarding points are natural seeds for a region-growing segmentation scheme. We can grow different regions simultaneously with the restriction of preserving their star-property.

3 Applications

We are exploring applications of region guarding and star decomposition, in such as shape interpolation, shape matching/retrieval, and robotics for optimal autonomous environment inspection, as illustrated in the accompanied video.

References

CHVATAL, V. 1975. A combinatorial theorem in plane geometry. *Journal of Combinatorial Theory Series B 18*, 39–41.

LIEN, J.-M. 2007. Approximate star-shaped decomposition of point set data. In *Eurographics Symposium on Point-Based Graphics*.