Ensuring the Semantic Correctness of Complex Regions

Mark McKenney, Alejandro Pauly,
Reasey Praing, & Markus Schneider*

CISE Department, University of Florida

*This work was partially supported by the National Science Foundation under grant number NSF-CAREER-IIS-0347574.

Overview

Introduction

Related Work

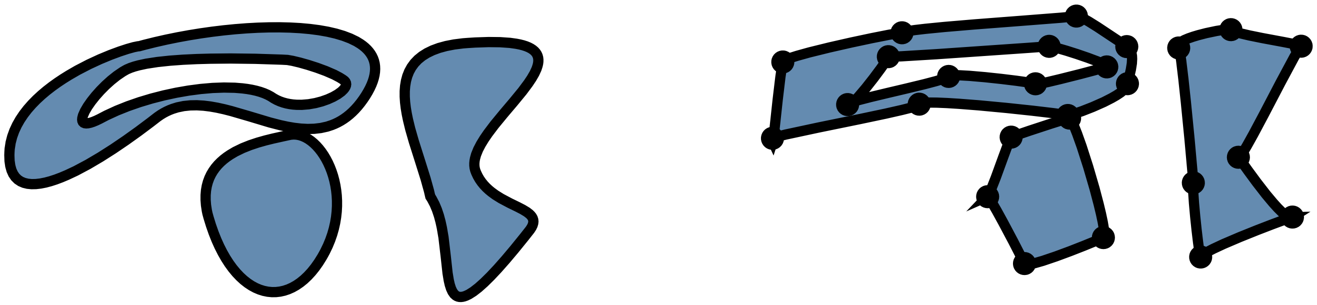
Complex Regions

Computing the Cyclic Structure of Complex Regions

Conclusion

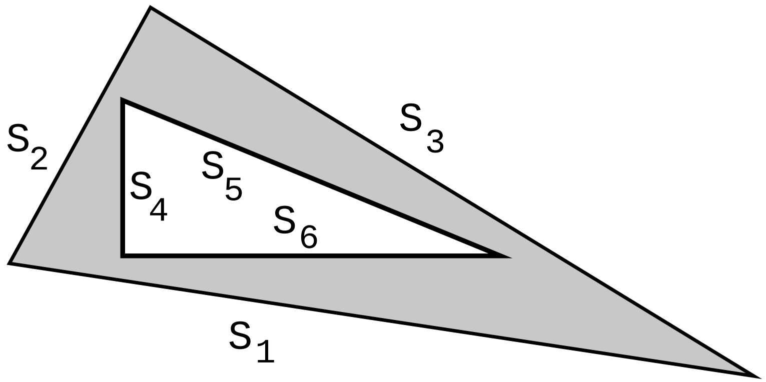
Introduction

Complex Regions



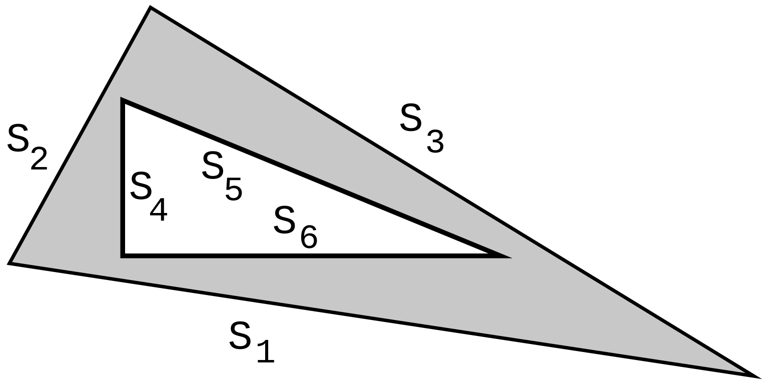
- Multiple faces
- Faces may meet at single points
- Faces may contain holes
- Face interiors may not overlap

Views of Complex Regions: Sequential View



- A complex region is an *ordered sequence of segments*
- Corresponds to input of *plane sweep algorithms*
- Easy to locate a particular *segment*
- Strucutural components are not represented

Views of Complex Regions: Component View



- Segments are grouped by structural compnents
- Holes are associated with their surrounding face
- Useful in visualization and manipulation
- Easy to locate a particular *cycle* (face or hole)

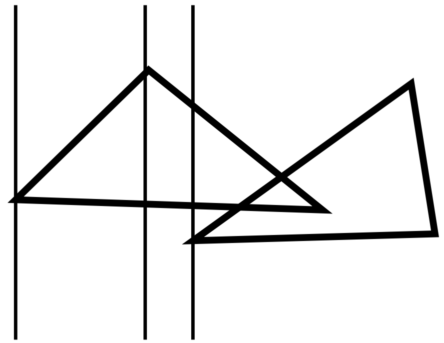
Validation of Complex Regions

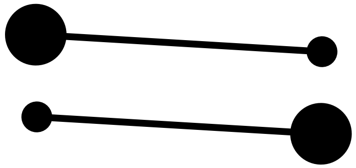
- Need to validate whether a given complex region is semantically and topologically correct.
- If it is given in component view, then the cyclic structure of the region is known and can be used for validation.
- Otherwise, the cyclic structure must be computed explicitly from a sequential view.
- Complex regions in sequential view are usually resulted from set operations such as intersection, union, and difference.
- Goal: Define an efficient algorithm to compute the cyclic structure of a complex region from its sequential view and at the same time validate its correctness.

Related Work

Related Work

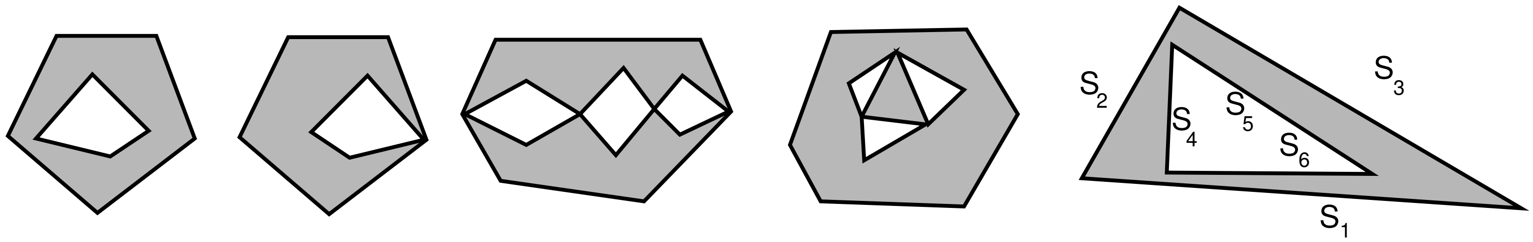
- No existing solution to our problem introduced.
- There are some works on polygon detection from a set of intersecting segments. But this problem is different from ours in that:
  - There are no restrictions on the polygons that are detected.
  - In our problem, we must consider holes and outer cycles separately.
  - All cycles must adhere to the definition of complex regions.
- The well known Plane Sweep paradigm can be used as a basis for our solution.





Complex Regions

Complex Region Data Type



(a) (b) (c) (d) (e)

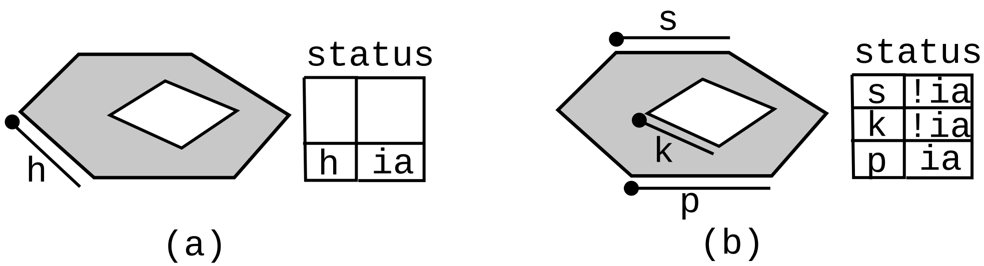
- Unique representation: there is exactly one valid semantic interpretation of the cyclic structure of a complex region.
- Our algorithm takes into account all the special cases.
- The ordered sequence of halfsegments for region (d) is $\langle h_1^l, h_2^l, h_6^l, h_4^l, h_4^r, h_5^l, h_2^r, h_3^l, h_5^r, h_6^r, h_3^r, h_1^r \rangle$.
- Each halfsegment has an *interior above* flag indicating if the region interior lies above or below the halfsegment

Computing the Cyclic Structure of Complex Regions

The Algorithm: Overview

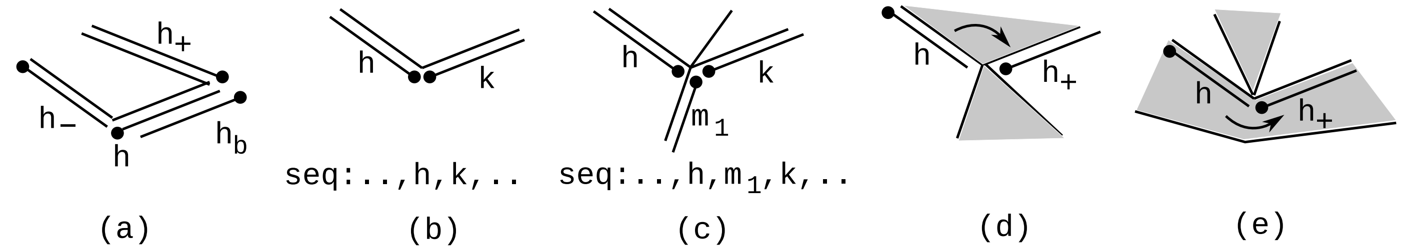
- Find an unvisited segment h in halfsegment order
- Determine if h is a outer or hole cycle
- Walk the cycle
  - Visit all segments belonging to the cycle containing h (mark them)
  - Walk is different depending on outer/hole cycle
  - Ensure the cycle is valid
- Repeat

Classifying Outer and Hole Cycles

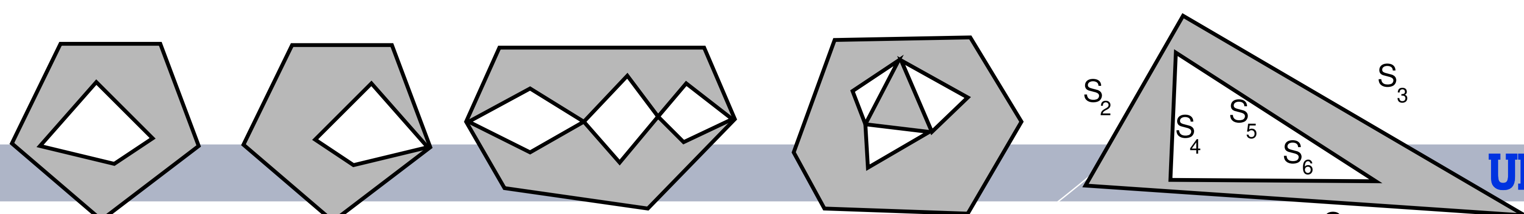


- Processing the smallest halfsegment h of the sequence (a) and k of a hole cycle (b).
- The smallest halfsegment always belongs to the outer cycle.
- The smallest halfsegment not yet visited is a left halfsegment
- The smallest halfsegment not yet visited is the leftmost halfsegment of a cycle
- The cycle walking process marks all halfsegments of this outer cycle as visited.
- The next yet to be visited halfsegment k is identified as belonging to a hole cycle.

Walking Cycle

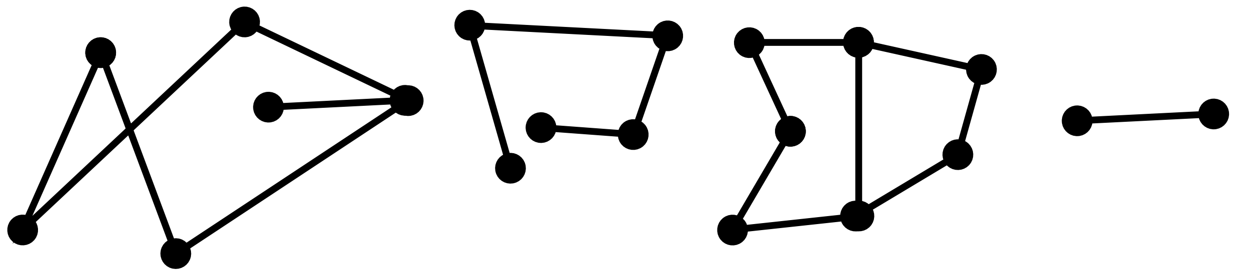


- Given a halfsegment, how to find the next in a cycle?
  - Calculate current halfsegments *brother*
  - Find the brother in the list of halfsegments
  - All halfsegments with the same dominating point are grouped
  - Find next halfsegment rotating clockwise around dominating point
  - (rotate counter-clockwise if walking a hole)
- Special cases
  - Anchored holes
    - If a point is encountered twice, construct a hole
  - Chain of suasages



Validation

- Invalid cases
  - Cycles that do not close (cannot find next halfsegment)
  - Cuts (will visit a halfsegment twice)
  - Faces that share a segment (will visit a halfsegment twice)
  - Intersecting segments (will be found by plane sweep)



The Algorithm

Algorithm 1: The algorithm for deriving the component view of a region.

Input: Sequence of unannotated halfsegments H
Output: Sequence H with fully annotated halfsegments

```
while not end of sweep do
    Advance sweep line to h. h is the left-most halfsegment yet to be annotated;
    Using sweep line status, determine h as part of an outer cycle or a hole cycle;
    NewCycle(h); Visit(dp(h));
    if h belongs to a hole then
        Using sweep line status, retrieve halfsegment f from its owning outer cycle;
        Owns(h, f);
        Set cycle walk mode to use counter-clockwise adjacency;
    else
        Set cycle walk mode to use clockwise adjacency;
    end
    /* Begin walking the cycle */
    c ← h+;
    while c ≠ h do
        if Visited(dp(c)) then
            q ← c; c ← c−; NewCycle(c); Owns(c, h);
            while dp(c) ≠ dp(q) do
                /* Trace back anchored hole */
                info(c−) ← info(c); c ← c−;
            end
        else
            info(c) ← info(h); Visit(dp(c)); c ← c+;
        end
    end
end
```

Complexity

- Store halfsegments in an array
- Use parallel arrays to indicate visited segments
- Plane sweep portion takes $O(n \log n)$
- Finding the brother of a halfsegment
  - Use binary search ($O(n \log n)$)
- Finding next clockwise or counterclockwise halfsegment around a point
  - Constant time

Conclusions and Future Work

- Developed an efficient algorithm that correctly determines the cyclic structure of a region represented as a list of halfsegments
- Algorithm also ensures the region is valid
- Eliminates the need to store the cyclic structure of a region

Thanks!