Use of Crossing-State Equivalence Classes for Rapid
Relabeling of Knot-Diagrams Representing 2\!/2D Scenes

Keith Wiley and Lance R. Williams
Department of Computer Science, University of New Mexico, Albuquerque, NM 87131

In our previous research, we have demonstrated a sophisticated computer-assisted drawing program called Druid, which permits easy construction of 2\!/2D scenes. A 2\!/2D scene is a representation of surfaces that is fundamentally two-dimensional, but which also represents the relative depths of those surfaces in the third dimension. This paper improves Druid's efficiency by exploiting a topological trait of 2\!/2D scenes called a crossing-state equivalence class. This paper describes this trait and how it is used by Druid.

I. INTRODUCTION

Our research focuses on the development of an advanced drawing program called Druid [1], which permits the construction of 2\!/2D scenes. A 2\!/2D scene is a representation of surfaces that is fundamentally two-dimensional, but which also represents the relative depths of those surfaces in the third dimension. To accomplish this, Druid uses labeled knot-diagrams to represent surfaces [2].

This paper describes a topological constraint on legal labelings which we call the crossing-state equivalence class rule. Use of this constraint allows us to rapidly relabel a knot-diagram representing a 2\!/2D scene.

II. DEFINITION OF KEY CONCEPTS

Fig. 1 shows a 2\!/2D scene of interwoven surfaces. A section of a boundary joining two crossings is termed a boundary segment. We observe that the canvas is partitioned into disjoint regions separated by boundary segments. In Fig. 1, the regions of the canvas are labeled with letters. We observe that every region is covered by zero or more surfaces (numbered in Fig. 1). For example, region k is covered by surfaces 1 and 3 while region m is covered by surfaces 1, 2, and 3.

To define and prove the crossing-state equivalence class rule, we first define the following terms:

- A superregion is a set of contiguous regions covered by a single surface. For example, in Fig. 1, \{ b, g, h, n \} is a superregion of surface 2.
- A border of a superregion is the set of boundary segments which define its perimeter.
- A shared superregion is the maximum superregion common to two surfaces, e.g., \{ g, m \} is a shared superregion of surfaces 1 and 2.
- A corner of a shared superregion is a crossing where adjacent boundary segments of the border belong to different surfaces. In Fig. 1, corners corresponding to the shared superregion \{ m, n \} common to surfaces 2 and 3 are marked with circles.

The corners of a shared superregion comprise the crossing-state equivalence class for that shared superregion. Notice that every crossing in a drawing is a corner of some shared superregion. Consequently, every crossing is a member of some crossing-state equivalence class.

III. REDUCING GENERAL 2\!/2D SCENES TO SIMPLE 2\!/2D SCENES

A simple surface is a surface with a single boundary component which does not intersect itself, i.e., a Jordan curve. Two steps are required to reduce a general 2\!/2D scene to a simple 2\!/2D scene. First, any surface with multiple boundary components (a surface containing holes) must be converted into a surface with a single boundary component. Second, any self-overlapping surfaces must be converted into a set of non-self-overlapping surfaces.

We perform both surface conversions using cuts [1]. A cut is analogous to a scissor cut through a surface from one boundary to another. When two boundaries are connected by a cut, they are joined into a single boundary component (Fig. 2). Likewise, a self-overlapping surface with a single boundary component can be cut into multiple smaller surfaces which abut and such that no surface in the final scene self-overlaps (Fig. 3).

Fig. 2: A cut connects two boundaries of a single surface into a single boundary for that surface.
IV. THE CROSSING-STATE EQUIVALENCE CLASS RULE

Let $X$ and $Y$ be the two surfaces whose boundaries intersect at a crossing. We observe that the crossing can only be in one of two states. Either surface $X$ is above surface $Y$ or surface $Y$ is above surface $X$.

**Theorem** All crossings in a crossing-state equivalence class must be in the same state.

**Proof** We first prove the above theorem for simple surfaces. Because any general 2/1D scene can be reduced to a simple 2/1D scene, this suffices to prove the theorem in the general case. We begin by observing the following:

- We observe that for every region there is a total depth ordering of the surfaces which cover that region.
- The total depth ordering of adjacent regions is identical except for the addition or deletion (depending on the sign of occlusion) of the surface whose boundary segment separates the two regions.
- It follows that the relative depth of two surfaces in adjacent regions remains the same if the boundary segment which divides the regions belongs to neither surface.
- It follows that the relative depth of two surfaces is constant within a shared superregion.
- The relative depth of the two surfaces whose boundaries intersect at a crossing is the same as the relative depth of those surfaces in the region they corner.

Consequently, the relative depth ordering of two surfaces at every crossing in a crossing-state equivalence class must be the same. □

For example, in Fig. 1, consider the superregion $\{ m, n \}$ shared by surfaces 2 and 3. The only segment interior to the superregion is part of the boundary of surface 1. Therefore, the relative depths of surfaces 2 and 3 cannot change along that boundary segment.

V. RESULTS

Fig. 4 shows a drawing of low complexity before (left) and after (right) two surfaces are flipped with respect to one another. There are three methods by which Druid can relabel a drawing, as follows:

1. Perform a tree search to find a new labeling.
2. Perform the same search but use the equivalence classes as a constraint during the search process.
3. Update the equivalence classes without a search and deduce the segment depth changes directly.

Method 1 is our original method. Method 2 provides significant improvements and is only adequate for drawings of high complexity. Method 3 is even better however. Table I shows the relabeling running times for the three methods on a 1.6GHz G5 PowerMac.

<table>
<thead>
<tr>
<th>Method</th>
<th>time (secs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>45.19s</td>
</tr>
<tr>
<td>2</td>
<td>.15s</td>
</tr>
<tr>
<td>3</td>
<td>&lt;.01s</td>
</tr>
</tbody>
</table>

**Table I:** Running times for three possible relabeling methods following the flip shown in Fig. 4.

Notice that Method 2 provides adequate improvement. In fact, it is within our tolerances for reasonable turnaround times. In other words, Method 2 provides sufficient performance in this example and Method 3 is unnecessary. It would be difficult to show an example that demonstrates the benefits of Method 3 over Method 2 because such an example would have to be extremely complex.

VI. CONCLUSION

In earlier work, we developed a system called Druid which permits the construction of interwoven 2/1D scenes. The new Druid system exploits a topological constraint on 2/1D scenes which we call the crossing-state equivalence class rule, and consequently can relabel knot-diagrams much more rapidly than the old system. This vastly extends the complexity of drawings that users of Druid can construct.