# *Druid*: Use of Crossing-State Equivalence Classes for Rapid Relabeling of Knot-Diagrams Representing 2½ D Scenes

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

## Introduction

In our earlier work, we developed *Druid* [2], a drawing program which permits construction of 2½ D scenes. *A 2½ D scene* is fundamentally 2D, but represents relative depths of surfaces. Conventional drawing programs use a layered representation which limits them to DAG-based scenes (Fig 1).

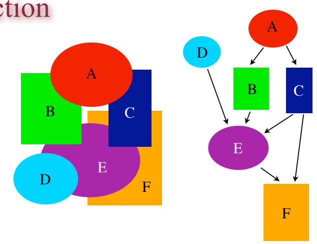



**Figure 1.** A surface DAG typical of a conventional layer-based drawing program.

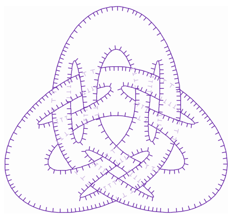


**Figure 2.** A *labeled knot-diagram* representation permits scenes of interwoven surfaces.

*Druid* represents 2½ D scenes with a *labeled knot-diagram* (Fig. 2) [1], which assigns a *sign of occlusion* to every boundary (shown hashed), to state which side of the boundary the surface lies on, and a depth index to every boundary segment.. This representation permits interwoven scenes.

A *legal labeling* is one in which every crossing honors the *labeling scheme* (Fig. 3), which specifies constraints on the relative depths at a crossing.

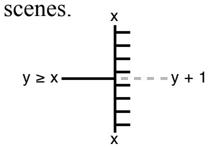



**Figure 3.** The *labeling scheme* constrains relative depths at a crossing.

## Three Labeling Methods

Occasionally, *Druid* must find a new legal labeling, *e.g.*, after a *surface-flip* user-interaction, in which the user inverts the relative depth ordering of two surfaces within an area of overlap (Fig. 5). It is desirable that the new labeling be a *minimum difference* labeling with respect to the labeling preceding the flip.

There are three methods for relabeling a figure:

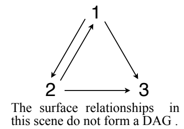
1. Perform a tree search (our original method).
2. Perform the same search using equivalence classes as a search constraint.
3. Maintain the equivalence classes without a search and deduce the resulting segment depth changes directly.

Although we have developed a number of search optimization techniques, *Druid's* capability using Method 1 remained limited due to long search times for complex drawings.

## Definition of Concepts

In the following definitions, examples refer to Fig. 4.

- ***2½ D scene*** - a scene of surfaces (surfaces shown numbered) which may overlap or interweave, *e.g.*, Figs. 1, 2, 4, 5, and 6.
- ***Boundary segment*** - a section of a boundary joining two crossings.
- ***Region*** - a partitioning of the canvas along boundary segments (regions shown lettered). Every region is covered by zero or more surfaces, *e.g.*, region *k* is covered by surfaces *1* and *3*.



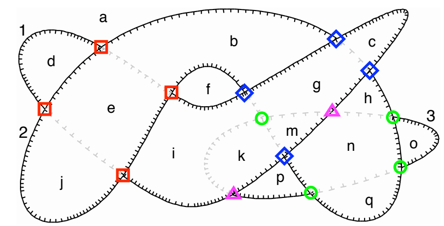




**Figure 4.** This figure shows a scene of interwoven surfaces with a number of features labeled.

- ***Superregion*** - a set of contiguous regions covered by a single surface, *e.g.*, { *b*, *g*, *h*, *n* } for surface *2*.
- ***Shared superregion*** - the maximum superregion common to two surfaces, *e.g.*, { *g*, *m* } for surfaces *1* and *2*.
- ***Corner of a shared superregion*** - a crossing where adjacent segments of a shared superregion's border belong to different surfaces, *e.g.*, corners for shared superregion { *m*, *n* } of surfaces *2* and *3* are marked with green circles.

## Crossing-State Equivalence Class

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. Crossing-state equivalence classes are marked with unique shapes/colors at the crossings in Fig 4.

## Crossing-State Equivalence Class Rule

Let *X* and *Y* be the two surfaces whose boundaries intersect at a crossing. The crossing can only be in one of two states. Either *X* is above *Y* or *Y* is above *X*.

**The *Crossing-State Equivalence Class Rule* states:**

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

The rule is proven in [3].

Consider the superregion { *m*, *n* } shared by surfaces *2* and *3*. The only segment interior to the shared superregion is part of the boundary of surface *1*. Therefore, surfaces *2* and *3* cannot change relative depth along that boundary segment. Thus, all corners of { *m*, *n* } (marked with green circles) must be in the same state.

## Results

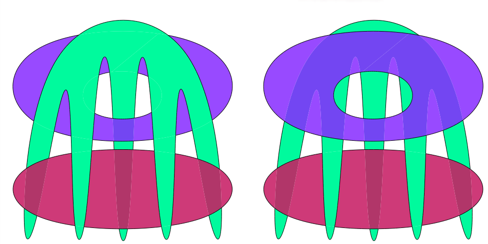


**Figure 5.** This figure shows a drawing before and after a shared superregion is flipped.

| Method | Crossing-state search space size |
|---|---|
| 1 | $2^{40}$ (for 40 crossings) |
| 2 | $2^{7}$ (for 7 equivalence classes) |
| 3 | N/A, *i.e.*, 0 (there is no search) |

**Table 1.** Crossing-state search space sizes for the three relabeling methods applied to Fig. 5.

The *labeling space* consists of all crossing-state labelings crossed with all segment depth labelings. Table 1 only shows the crossing-state search space sizes for Fig. 5. Method 2 is more efficient than Method 1 by a factor of $2^{33}$, or 8,589,934,592. Method 3 is even better.

Table 2 shows relabeling running times for the flip shown in Fig. 5 on a 1.6GHz G5 PowerMac. We observe that Method 2 is adequate for most drawings. Method 3 can extend *Druid's* capability even further however.

| Method | Time (secs) |
|---|---|
| 1 | 45.19s |
| 2 | .15s |
| 3 | <.01s |

**Table 2.** Relabeling running times for the three methods applied to Fig. 5.

## Example

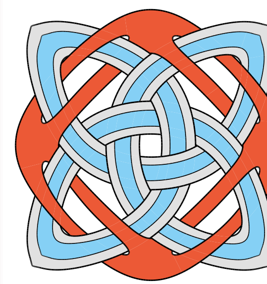


**Figure 6.** A fairly complex scene that our original system could not handle in reasonable running times. The new system performs very well, with response times on the order of a few seconds..

Fig. 6 shows a scene with 256 crossings and 64 equivalence classes.

*Druid* can find the equivalence classes and label this figure from scratch in 6.04 seconds. Method 1 fails to find any legal labeling in a reasonable time (the search was terminated after a few minutes).

Subsequent surface-flips are instantaneous.

## Conclusions

In our earlier work, we developed *Druid*, a system for constructing interwoven 2½ D scenes. Past versions of *Druid* relied on a tree search to find a new labeling following many user-interactions. Even with substantial optimization techniques, this search hindered *Druid's* scalability.

We have discovered a topological trait of 2½ D scenes which we call the crossing-state equivalence class rule. Exploitation of this trait can alleviate the need to search in some situations, and can dramatically reduce the search space in remaining situations that require a search. Thus, we have vastly extended the complexity of drawings that users of *Druid* can construct.

## References

[1] Lance R. Williams. *Perceptual Completion of Occluded Surfaces*. Ph.D. dissertation. Univ. of Massachusetts at Amherst, 1994.

[2] Keith Wiley and Lance R. Williams. *Representation of interwoven surfaces in 2½ D drawing*. In *CHI Proceedings*, 2006.

[3] Keith Wiley and Lance R. Williams. *Use of Crossing-State Equivalence Classes for Rapid Relabeling of Knot-Diagrams Representing 2½ D Scenes*. CSUSC, 2006.