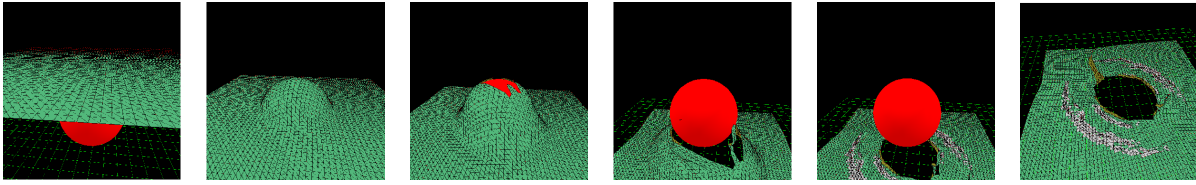


# Collision Detection for Objects with Changing Mesh Topology (sketches\_0346)

David Knott

University of North Carolina at Chapel Hill



## 1 Introduction and Motivation

Deformable objects that break, rip, tear, or otherwise change topology are common in animation and simulation [Molino et al. 2004]. Similarly, schemes for multiresolution geometry such as subdivision surfaces can arbitrarily change the connectedness of a mesh. However, changes in topology alter many of the basic assumptions of collision detection algorithms. We present an expansion on our previous work on collision detection for deformable objects [Govindaraju et al. 2005], taking into account possible changes in the connectedness and topology of the underlying geometric model.

## 2 Background

We extend the concept of chromatic decomposition in order to take into account changes in mesh connectedness. Chromatic decomposition is a colouring of the polygons of a mesh, such that the self-collision problem may be transformed into N-body collision detection. A valid chromatic decomposition satisfies two constraints:

- No polygons of the same colour are adjacent to each other
- No polygon is adjacent to two polygons of any other colour.

The mesh is decomposed into sets, where each set contains polygons of the same colour. Collision detection is performed using pair-wise tests between sets. A set has a unique ordering for each other set that it is tested against (see [Govindaraju et al. 2005]).

## 3 Our Extended Approach

When polygons are added or deleted from the mesh, or polygons are disconnected from each other (such that they no longer share a vertex), the extended dual graph of the mesh is altered. An alteration in the extended dual graph has two implications:

- The colouring of the of the graph may become invalid. This occurs only when new polygons are added to the mesh or new connections are made between existing polygons. It cannot occur when polygons are deleted from the mesh or polygons are disconnected from each other.
- Sets may need to be reordered for pair-wise set collision detection. This occurs when polygon-polygon connections are either added to or removed from the mesh.

Graph colouring is an NP-Hard problem. This means that mesh recolouring and set reordering are the two most expensive operations associated with addition or removal of polygons in the mesh. We therefore propose that some form of local recolouring scheme is necessary in order to maintain interactive performance.

Fortunately, polygon addition and removal can be shown to impose strictly local requirements on mesh recolouring. These constraints can be divided into three categories:

- New graph edges between the polygon and its one-ring
- New graph edges between the polygon and its two-ring
- New graph edges between every pair of polygons in the new polygon's one-ring (if they are not already connected)

The polygons in the new polygon's two-ring have new connections only to the new polygon. Since the new polygon must be recoloured, there is no necessity to recolour any polygons in the two-ring (or beyond). The one-ring must be recoloured, since new graph edges have been introduced between all one-ring polygons.

Polygon addition and graph recolouring proceeds as follows:

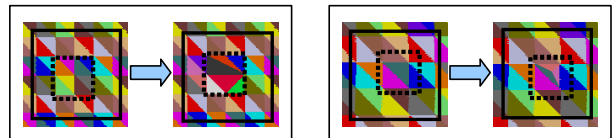
- Introduce new edges to the graph
- Recolour the one-ring of the new polygon in a greedy fashion
- Colour the new polygon with the first available colour
- Create new colouring sets if the previous two recolouring steps resulted in an increase in the number of colours

Furthermore, if more than one polygon is added or deleted, the recolouring is still strictly local. When a connected cluster of polygons is removed, there is still no recolouring required. When a connected cluster of polygons is added, recolouring is necessary only for the cluster and the one-ring of the cluster.

The locality of recolouring also means that the time required to recolour a mesh is independent of its resolution. Performance of recolouring is instead dependent on the number of polygons in the one-ring of the affected area, which is in general quite small.

## 4 Recolouring Examples

Examples of recolouring due to polygon addition and removal are shown below. The added/deleted polygons are shown inside the dotted lines. The region that requires recolouring is roughly demarcated by the solid lines. The first example shows recolouring due to an edge-collapse scheme. The second example shows recolouring due to a simple subdivision scheme. The recolourings required by other subdivision schemes are similar.



## 5 Results and Future Work

The images at top show simulated cloth that tears as it falls over a sphere-like object. The potentially colliding triangles of the mesh are highlighted in white.

We have found graph recolouring to be quite efficient, requiring only a few tens of microseconds for a single polygon addition or deletion. Set re-ordering is more costly, but still an order of magnitude faster than the collision detection itself.

With more advanced subdivision surface or progressive mesh schemes, additional heuristics could be added to the recolouring and set reordering method. For instance, the recolouring requirements could be encoded in the progressive mesh itself.

Finally, it is quite possible to combine our method with sophisticated simulation techniques such as [Molino et al. 2004] to handle tears and fractures which are not limited to mesh edges.

## References

- GOVINDARAJU, N. K., KNOTT, D., JAIN, N., KABUL, I., TAMSTORF, R., GAYLE, R., LIN, M. C., AND MANOCHA, D. 2005. Interactive collision detection between deformable objects using chromatic decomposition. In *Proc. of SIGGRAPH 2005*.
- MOLINO, N., BAO, Z., AND FEDKIW, R. 2004. A virtual node algorithm for changing mesh topology during simulation. In *Proc. of SIGGRAPH 2004*.