Hierarchical Streamarrows for the Visualization of Dynamical Systems

Helwig Löffelmann, Lukas Mroz, and Eduard Gröller

Overview: Introduction / Streamarrows / Hierarchical Streamarrows

Institute of Computer Graphics, Vienna University of Technology

Dynamical Systems – Necessary Terms

- **Continuous vs. discrete –** differential vs. difference equations

$$\boxed{\dot{x} = f(x, p)} \qquad \Delta x_i = f(x, p) \qquad x \in \Re^n, p \in \Re^m$$

- **Phase space –** $\Re^n$: each axis is assigned one variable of the dynamical system
- **Trajectory –** streamline of the dynamical system, depends on initial condition x_0

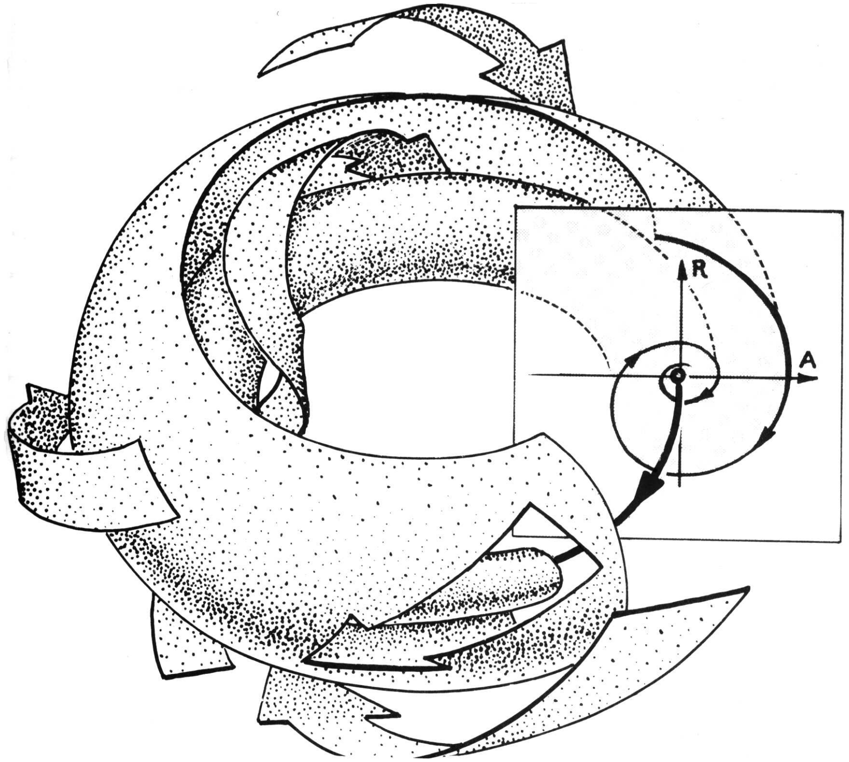
$$\underline{T(t, x_0)} = x_0 + \int_0^t f(\underline{T(\tau, x_0)}, p)\, d\tau$$

Streamarrows by Abraham & Shaw, 1992

Streamsurface: solution of the dynamical system, based on *a set of initial conditions*, e.g., a line segment in phase space

Streamarrows: Using streamlines and timelines to model arrow-shaped patches, cut out of a streamsurface

"Inverse" streamarrows: cutting streamarrows out of a streamsurface

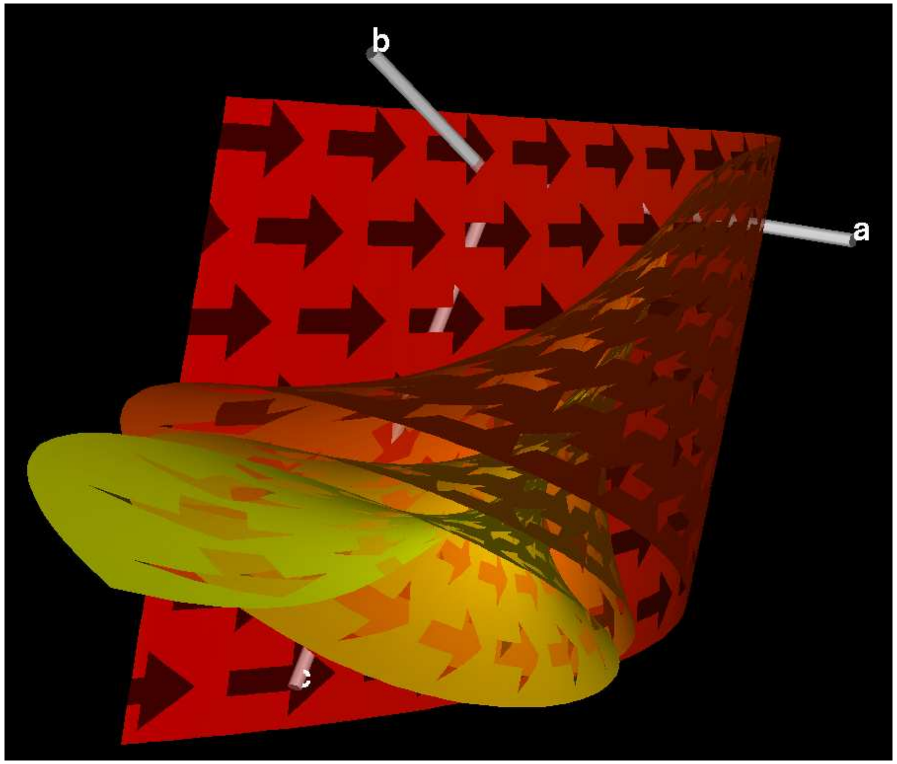


Streamarrows by Löffelmann et al., 1996

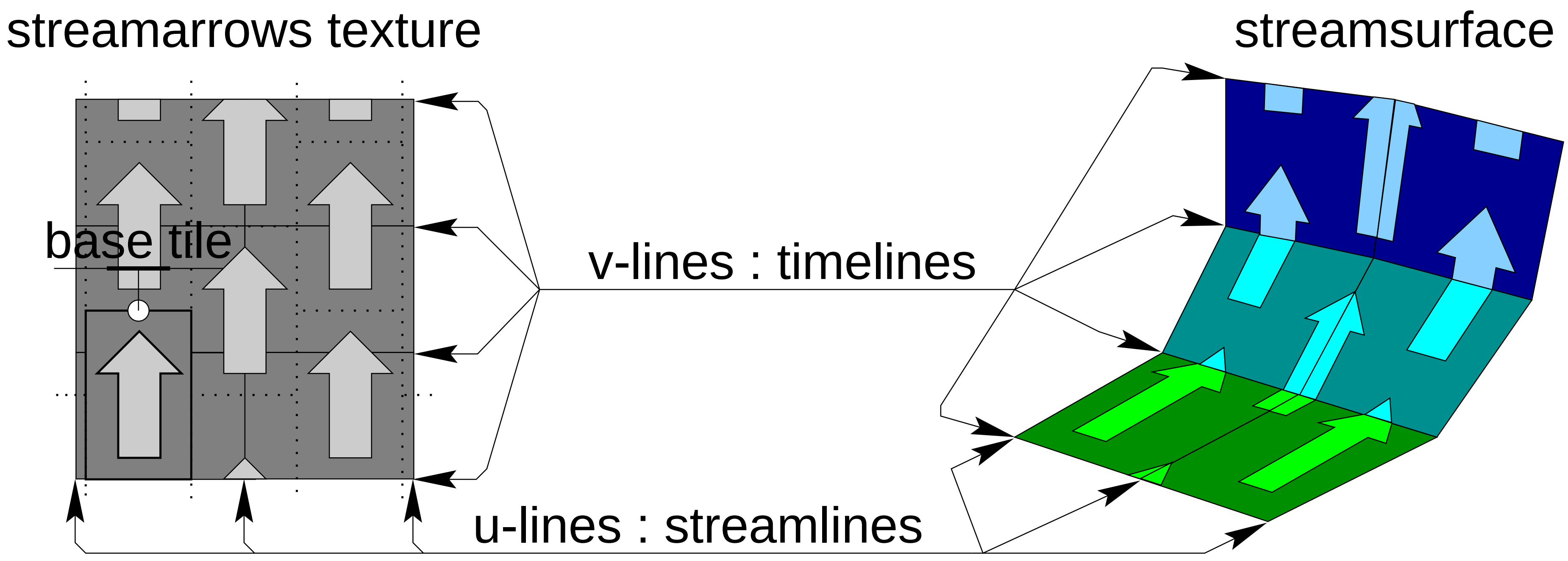
Separation algorithm:

Streamarrows $\leftrightarrow$ remaining streamsurface portions

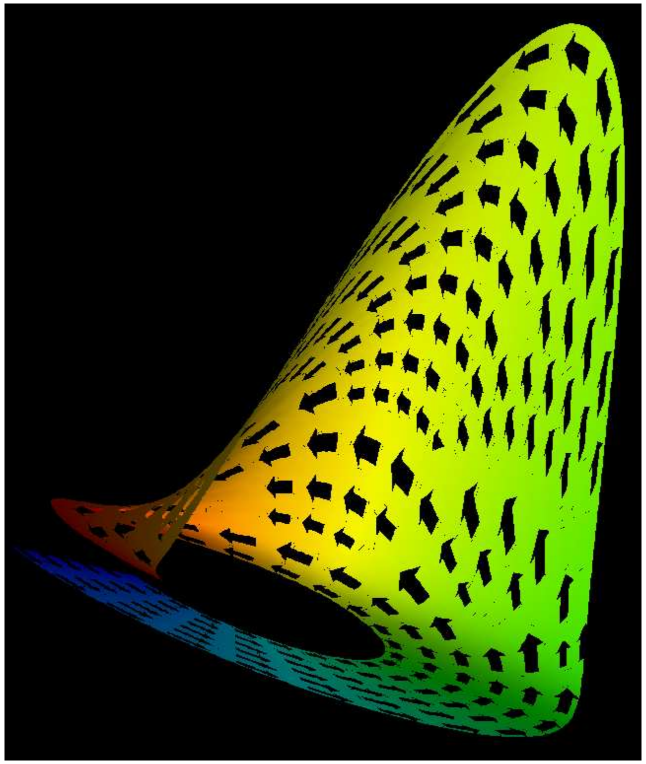
+ less occlusion
+ better than uniform transparency
+ additional visualization of, e.g., flow direction and velocity
– suboptimal in regions with a lot of divergence or convergence

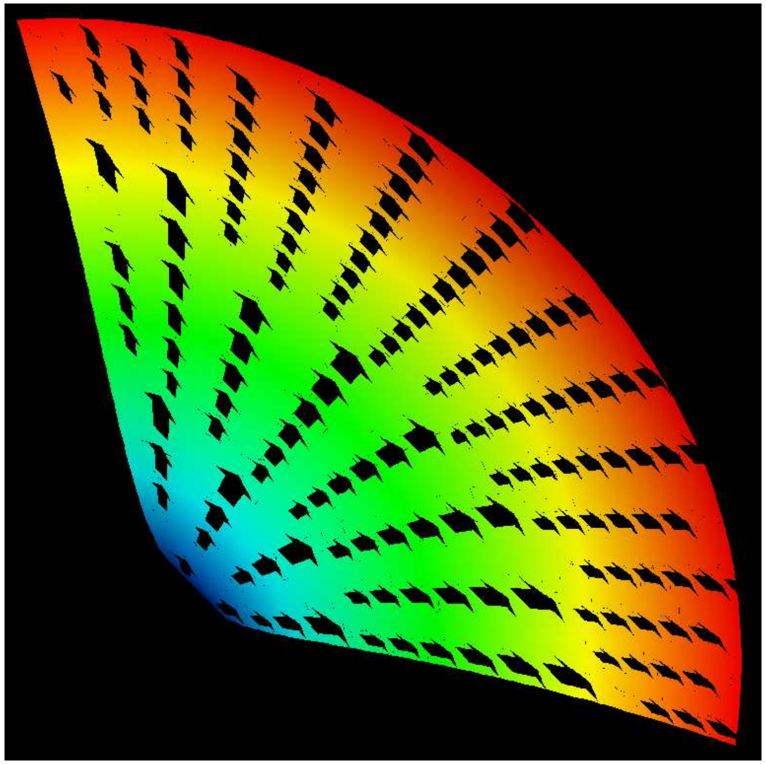


Streamarrows Texture and Mapping

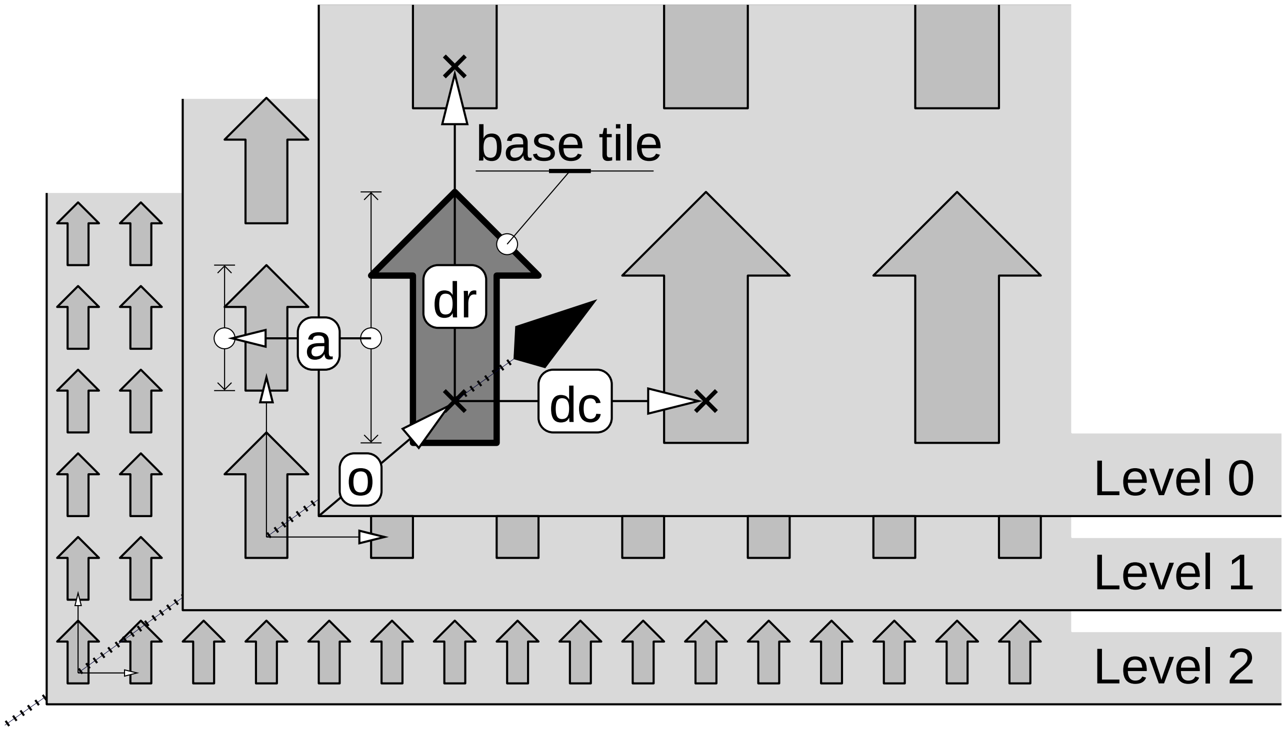


Hierarchical Streamarrows





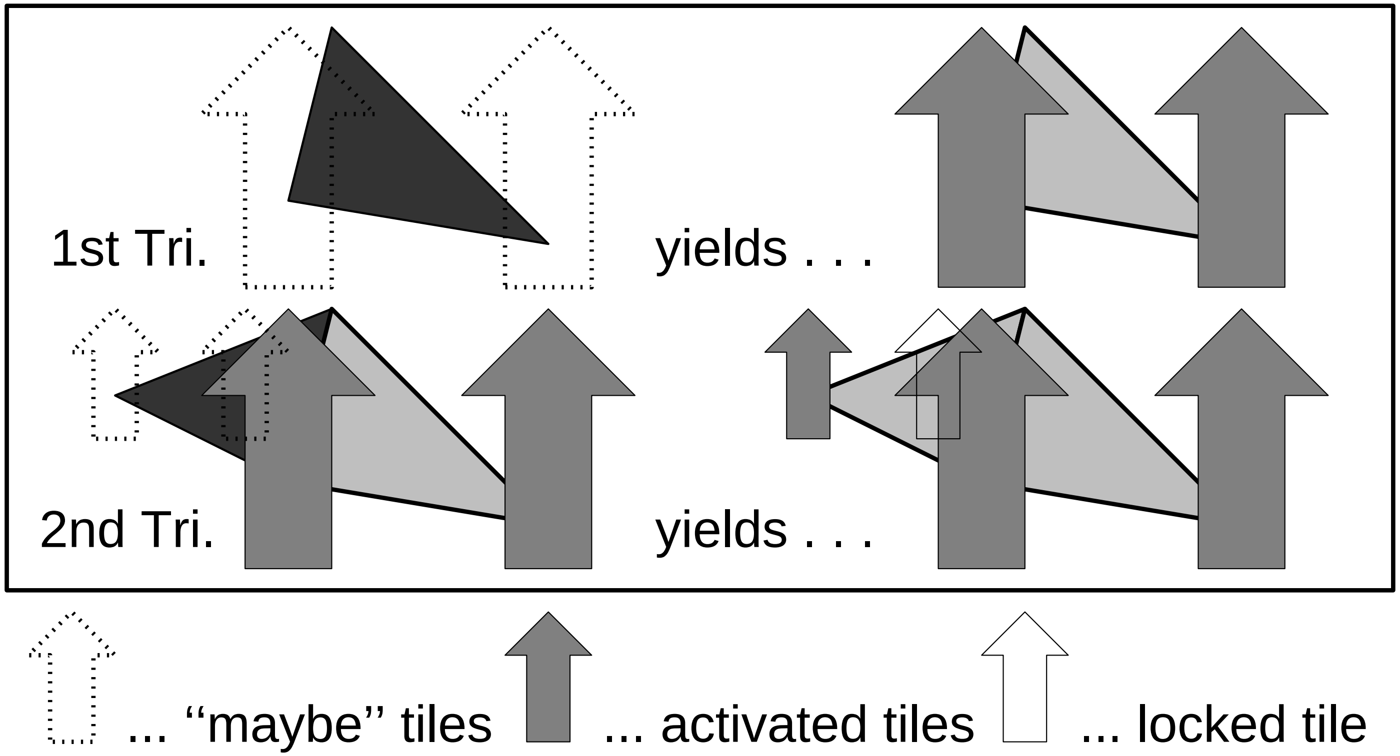
The Hierarchical Streamarrows Texture



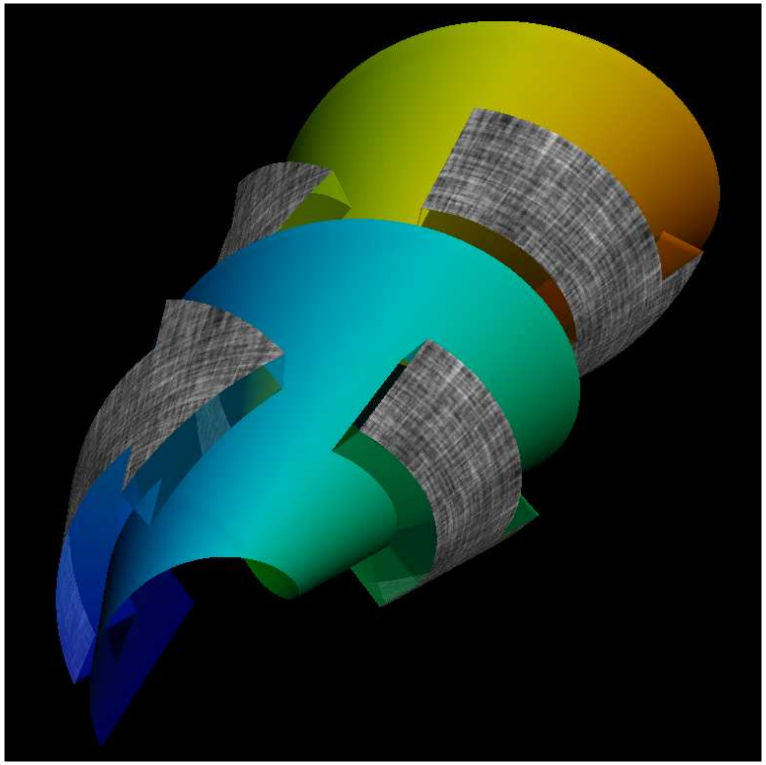
The New Separation Algorithm

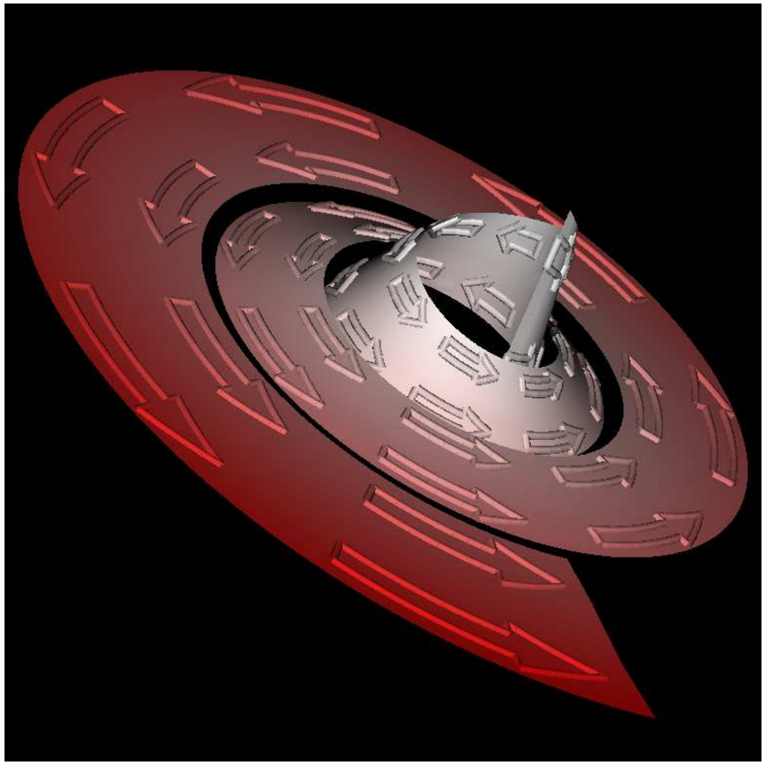
```
activeTiles = {}  // . . . . . . . . . . . . . IDs of active tiles
lockedTiles = {}  // . . . . . . . . . . . . . IDs of locked tiles
FOR ALL Triangles tri DO:
| level:=findLevelOfTriangle(tri)  // . get most appropriate level
| tiles:=getMaybeTiles(tri,level)  // . . .  get overlapping tiles
| FOR ALL Tiles tile IN tiles DO:
| | IF NOT (tile.active OR tile.locked) THEN:
| | | IF overlap(tile,activeTiles) THEN: tile.lock
| | | ELSE:                              tile.activate
| intersect(tri,activeTiles)  // . . . . . . . . do the separation
```

Activating and Locking – An Example

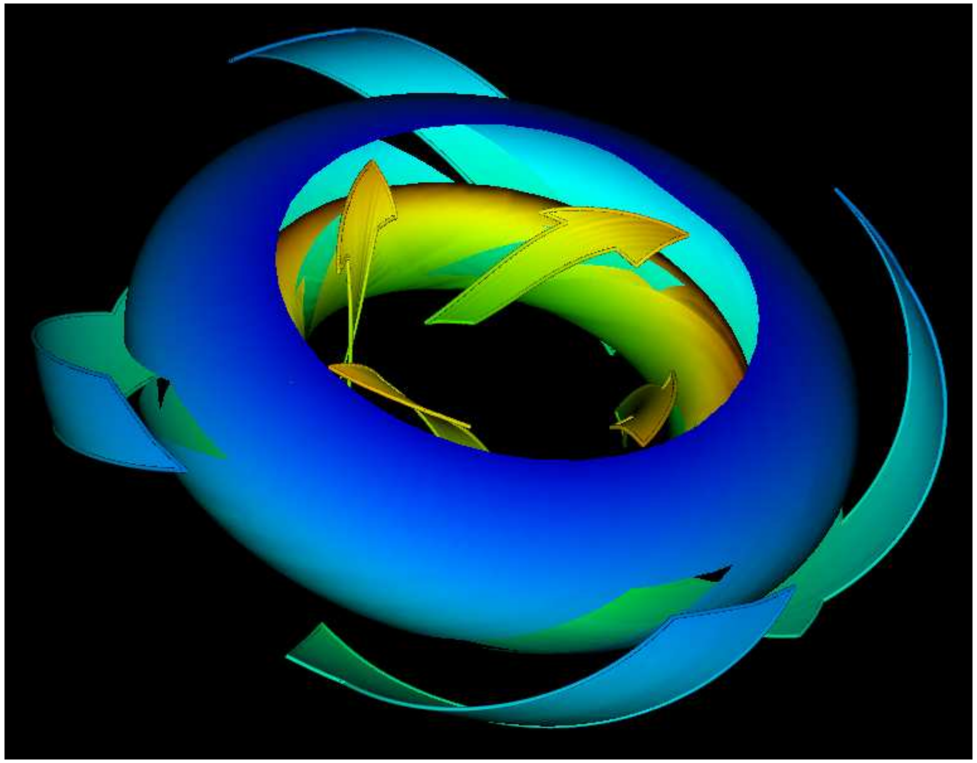


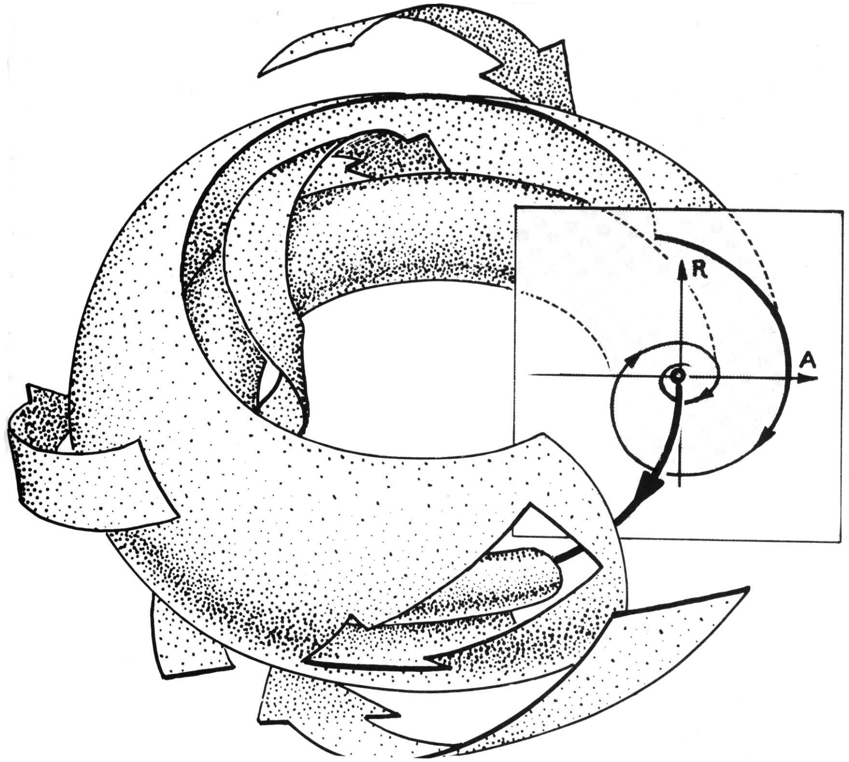
Streamarrows – 3D Extensions





Streamarrows – Now and Then :-)





Summary and Conclusions

- **Streamarrows –** less occlusion when streamsurfaces are used
- **Local properties –** enhanced visualization, more information encoded
- **Hierarchical streamarrows –**
  stable technique, works fine even for ill-shaped streamsurfaces
- **3D extensions –** additional improvements (lifted streamarrows, 3D arrows)
- **Future work –** further extensions to streamsurface textures