Geant 4

October 2006

Field Visualisation

Jane Tinslay, SLAC

Stanford Linear Accelerator Center

Outline

- Objectives
- Requirements
- Current Status
- Future plans

Objectives

- Develop generic field visualisation system
  - Automatic & interactive field line generation
  - 3D & 2D viewing
  - Animation
  - Offer range of visualisation methods - arrow plots, streamlines, line integral convolution…

Requirements

- Interactive capability very important
  - ➔ Explore potentially complicated field configurations
  - ➔ Field visualisation needs to be fast
- Occulusion a big problem in 3D field visualisation
  - ➔ Provide techniques to improve depth perception
- Multiple streamline seeding strategies
- Multiple streamline integration methods
- Multiple visualisation methods
- Integration into Geant4 framework
  - ➔ All drivers should have access to generated streamlines
  - ➔ Front end visualisation framework should be independent of field type

Current Status

- Reasonably early stage
- General flow visualisation approach
  - ➔ "Strategies for Interactive Exploration of 3D Flow Using Evenly-Spaced Illuminated Streamlines" Oliver Mattausch, Thomas Theul , Helwig Hauser , and Meister Eduard Groller
- Prototype automatic generation of evenly spaced streamlines
- Investigating streamline visualisation techniques
  - ➔ Use Open Inventor to implement illuminated streamlines

Evenly Spaced Streamlines

- “Creating Evenly-Spaced Streamlines of Arbitrary Density”, Bruno Jobard and Wilfrid Lefer
  - ➔ Simple algorithm to generate long, evenly spaced streamlines in a single pass
  - ➔ Generate seed points a distance d_{sep} from each streamline
  - ➔ Integrate forwards and backwards until streamline comes within a distance d_{test} to an existing streamline, where d_{test} = x%*dsep

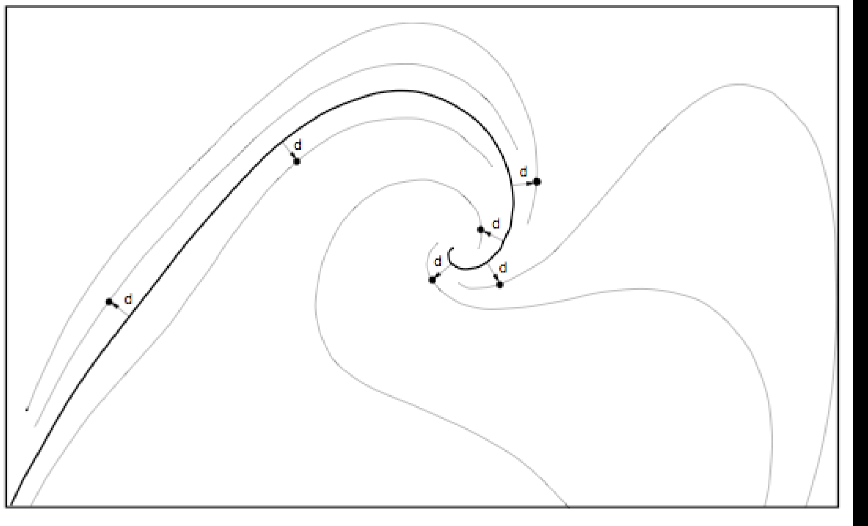


Fig. 3. streamlines are derived from the first (thick) one by choosing seed points (circles) at a distance $d = d_{sep}$ from it

- Use regular cartesian search grids to test validity of next seed/streamline point
- For simplicity, test distance between candidate point and sample points along existing streamlines
- Use G4ClassicalRK4 for actual streamline integration
- Hard code magnetic field configuration for testing

Streamline Visualisation

- Use standalone Open Inventor (Coin) app for testing
- Test case based on streamlines generated with G4LineCurrentMagField
  - ➔ 21 streamlines
  - ➔ 2947 vertices
- Rendering using cylinders too slow for interactive investigation of field
- Poor visual perception with regular line primitives
  - ➔ No illumination
- Illuminated lines produce reasonable effects
  - ➔ ~25 times faster than cylinders

Illuminated Streamlines

- "Interactive Visualization Of 3D-Vector Fields Using Illuminated Stream Lines" Malte Zockler, Detlev Stalling, Hans-Christian Hege
- Realistic illumination of line primitives through hardware based texture mapping
- Generalisation of Phong reflection model of diffuse, ambient and spectral terms. Light intensity given by:

$$I = k_a + k_d L \cdot N + k_s (V \cdot R)^n$$

  - ➔ V = viewing direction
  - ➔ R = reflection vector
  - ➔ N = shininess (width of highlights)

- For line primitives, define normal as the normal coplanar to L and tangent T

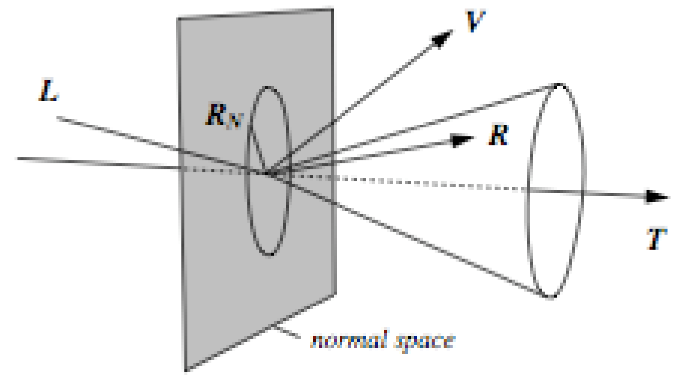


Figure 1: For line primitives there are infinitely many possible reflection vectors $\boldsymbol{R}$ lying on a cone around $\boldsymbol{T}$. For the actual lighting calculation we choose the one contained in the $\boldsymbol{L}$-$\boldsymbol{T}$-plane.

Ref: Zoller

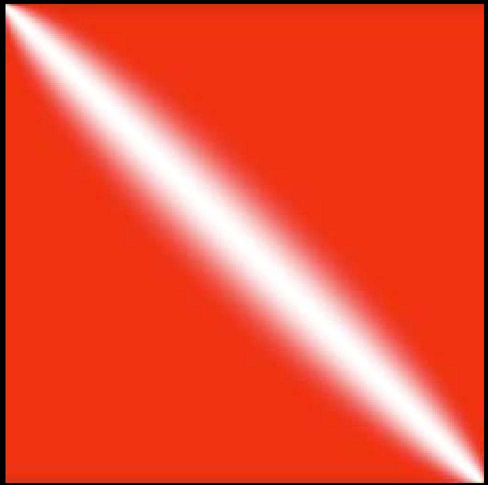
- Expensive to calculate new normal for each new view/light direction in software
  - ➔ Exploit texture mapping capabilities of graphics hardware
  - ➔ OpenGL

- Manipulate lighting equation such that light intensity is calculated using texture map & texture matrix
  - ➔ Texture coordinates given by tangent vector
  - ➔ Reload texture matrix with new viewing/lighting directions when necessary
  - ➔ Strealine colour defined through texture map

Texure matrix

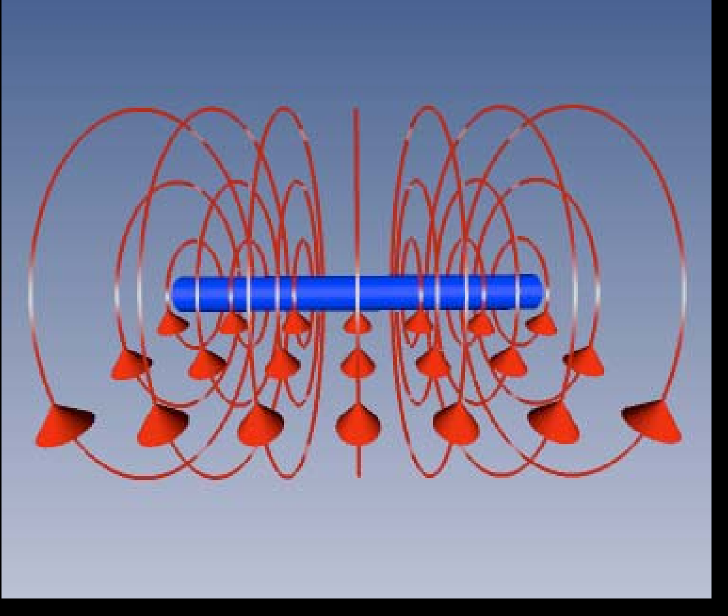
$$M = \frac{1}{2}\begin{pmatrix} L_1 & V_1 & 0 & 0 \\ L_2 & V_2 & 0 & 0 \\ L_3 & V_3 & 0 & 0 \\ 1 & 1 & 0 & 2 \end{pmatrix}$$

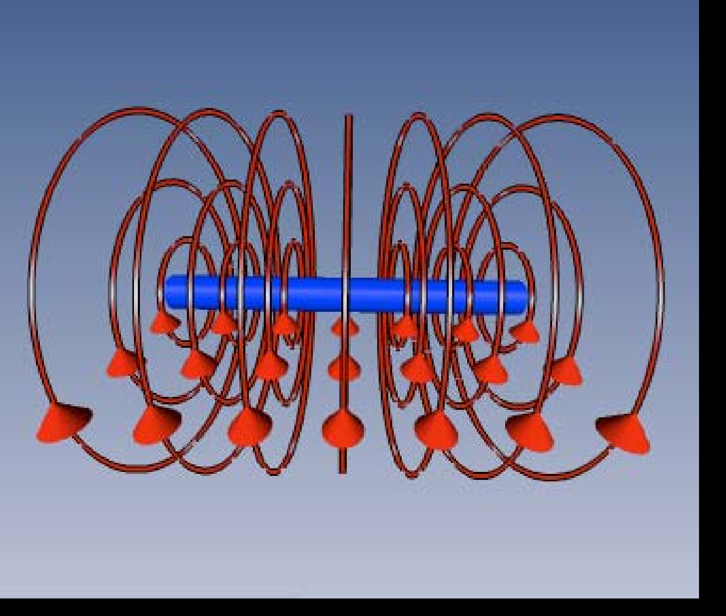
Example texture map



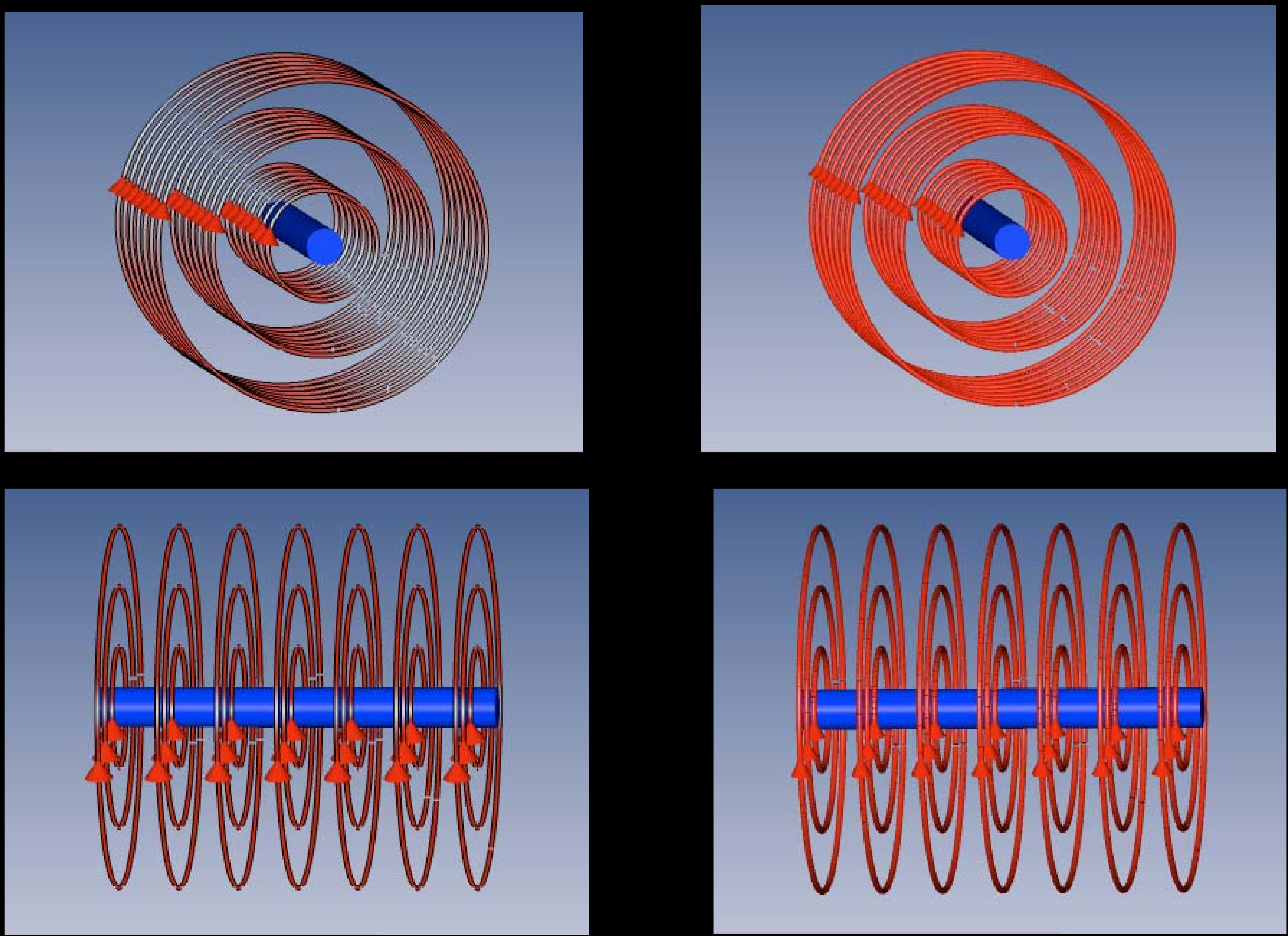
Haloing

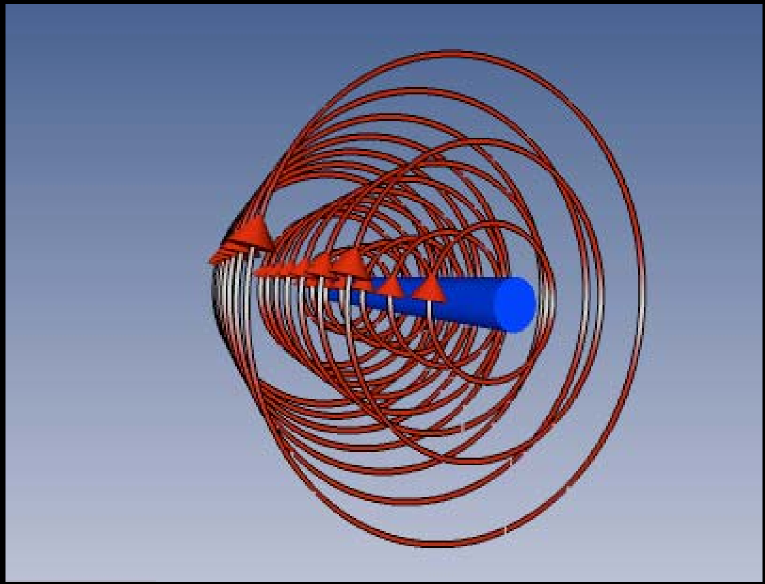
- Simple way to enhance depth perception
- Draw a larger black line behind streamline to form a black border

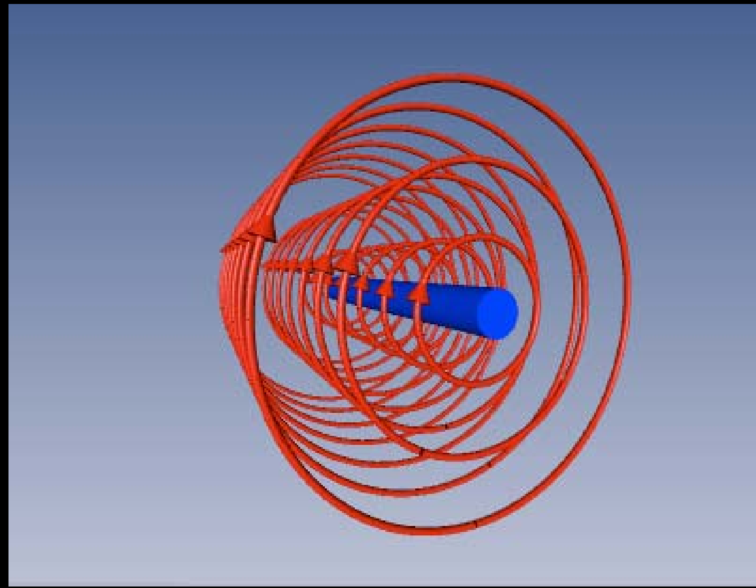


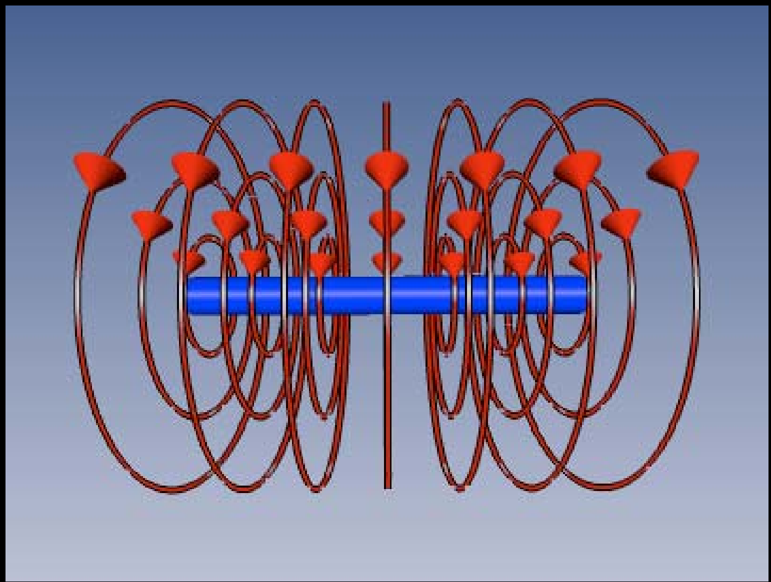


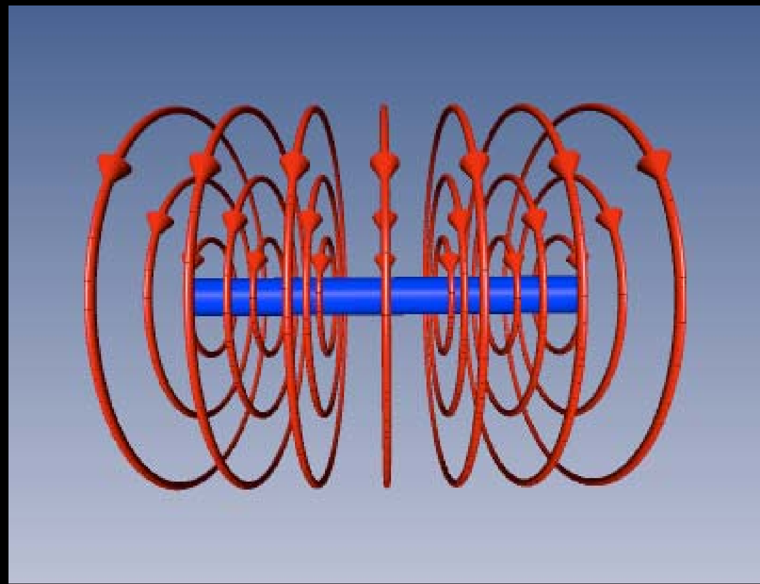
Line/Cylinder Comparisons











Conclusions & Future work

- Fast illuminated streamline visualisation promising
  - ➔ Understand artifacts
  - ➔ Perhaps create a toggle between line & cylinder rendering methods
- Investigate other methods enhancing visual perception for line primitives
  - ➔ Fog
  - ➔ Tapering
  - ➔ Transparency
  - ➔ Colour
  - ➔ …

- Investigate more seeding strategies
  - ➔ Spacing proportional to field strength
  - ➔ Seeding templates for known field configurations
  - ➔ Random seeding
- Streamline integration
  - ➔ Integration methods
  - ➔ Access to field in Geant4
- For December release, look at integrating BlineTracer example into visualisation system
- Aim to release some of new streamline based work in June release