Informatica Grafica

Laurea Specialistica in Ingegneria Informatica

- Il corso di lezioni intende fornire una visione approfondita degli aspetti matematici, geometrici e algoritmici della grafica sintetica al calcolatore

- Il corso di lezioni utilizza il linguaggio PLaSM come ambiente di sviluppo rapido di esempi e di applicazioni

- Il corso di esercitazioni fornisce una introduzione alla programmazione grafica SVG

Programma

- Elementi di geometria e algebra, di geometria poliedrale e differenziale.
- Trasformazioni affini, definizioni e proprieta‘. Rappresentazione dei tensori.
- Grafi gerarchici della scena, algoritmi di traversal e graph culling, ottimizzazione.
- Generalita‘ su ambienti grafici: GKS, PHIGS, OpenGL, VRML, SVG.

- Pipeline grafiche 2D e 3D. Primitive e attributi. Sistemi di coordinate, trasformazioni modeling, view orientation, view mapping, trasformazione prospettica e di dispositivo.

- Proiezioni prospettiche e parallele. Tassonomia delle proiezioni. Modelli di vista. Rendering: algoritmi approssimati ed esatti di rimozione delle superfici nascoste: ordinamento in profondita', alberi BSP, algoritmo z-buffer.

- Modelli locali di illuminazione, modelli di colore, shading di Gouraud e di Phong

- Elementi di animazione. Shape morphing e inbetweening. Spazio delle configurazioni,

Materiale didattico

- E. Angel, OpenGL, A Primer, seconda edizione, Pearson & Addison-Wesley, 2005
- A. Paoluzzi, Geometric Programming for Computer Aided Design, Wiley, 2003
- Installers del linguaggio PLaSM per Linux, Mac OS X, Windows

Scalable Vector Graphics (SVG)

XML Graphics for the Web

SVG is a language for describing two-dimensional graphics and graphical applications in XML.

SVG 1.1 is a W3C Recommendation and forms the core of the current SVG developments.

SVG 1.2 is the specification currently being developed as is available in draft form

The SVG Mobile Profiles: SVG Basic and SVG Tiny are targetted to resource-limited devices and are part of the 3GPP platform for third generation mobile phones. SVG Print is a set of guidelines to produce final-form documents in XML suitible for archiving and printing. sXBL is a binding language for SVG content.

Overview

SVG is a platform for two-dimensional graphics.

It has two parts: an XML-based file format and a programming API for graphical applications.

Key features include shapes, text and embedded raster graphics, with many different painting styles.

It supports scripting through languages such as ECMAScript and has comprehensive support for animation.

SVG is used in many business areas including

- Web graphics,
- animation,
- user interfaces,
- graphics interchange,
- print and hardcopy output,
- mobile applications and
- high-quality design.

SVG is a **royalty-free vendor-neutral open standard** developed under the W3C Process.

It has *strong industry support;*

Authors of the SVG specification include:

Adobe, Agfa, Apple, Canon, Corel, Ericsson, HP, IBM, Kodak, Macromedia, Microsoft, Nokia, Sharp and Sun Microsystems.

SVG viewers are deployed to over 100 million desktops, and there is a broad range of support in many authoring tools.

SVG builds upon many other successful standards such as

- XML (SVG graphics are text-based and thus easy to create),
- JPEG and PNG for image formats,
- DOM for scripting and interactivity,
- SMIL (Synchronized Multimedia Integration Language) for animation
- CSS for styling

Applications of SVG in industry

- **Mobile**

  In 2001 the mobile phone industry chose SVG as the basis for its graphics platform.
  Many leading companies joined the SVG effort to produce the *SVG Tiny* and *SVG Basic* profiles, collectively called **SVG Mobile**

- **Print**

  The combination of rich graphical features, comprehensive text support and resolution independence in SVG produce a **format suited to printing**.

  Leading print hardware companies are currently developing the *SVG Print* specification: a version of SVG specifically suited to hard-copy output

- **Web Applications**

  Web-based applications are increasing in popularity. Developers are often limited by browser incompatibilities and missing functionality.

  With powerful scripting and event support, SVG can be used as a platform upon which to build graphically rich applications and user interfaces

- **Design and Interchange**

  SVG is well suited to the **high-end graphical design market** common in the Aerospace, Transportation, Automotive and Telecommunication industries.

  The extensibility of XML allows SVG diagrams to have **embedded metadata** in proprietary formats without affecting the presentation

- For example, a CAD program could export to SVG for **online display**, but **embed data** within the file that facilitates future editing or roundtripping.
- Also, since many design tools support import and export of SVG, it can be used as an **interchange format** between applications.

- **GIS and Mapping**

  Geographic Information Systems have very specific requirements: **rich graphics** features, support for **vector** and **raster** content and the ability to handle a **very large amount** of data.

  SVG is well-suited to this market and many GIS systems provide SVG export

- SVG is a perfect complement to the **OpenGIS** consortium's **GML** format. GML, also XML-based, describes geographical elements such as rivers and roads.
- It can be converted into SVG using an **XML pipeline for online display**

Technical Details

SVG is a language for describing two-dimensional graphics in XML.

SVG allows for three types of graphic objects:

- **vector graphic paths** consisting of straight lines and curves,
- **images** and
- **text**

- Graphical objects can be
  - **grouped**,
  - **styled**,
  - **transformed** and
  - **composited**

  into previously rendered objects.
- Text can be in any **XML namespace** suitable to the application

- The feature set includes **nested transformations**, **clipping** paths, **alpha** masks, **filter** effects, **template** objects and **extensibility**.
- SVG drawings can be **dynamic** and **interactive**.
- The **Document Object Model** (DOM) for SVG, which includes the full XML DOM, allows for straightforward and efficient **vector graphics animation** via scripting

SVG Implementations

This is the official W3C list of implementations of Scalable Vector Graphics (SVG)

SVG Viewers | Mobile SVG Viewers | Native SVG Editors | SVG-exporting Editors | SVG Converters | Server-side SVG generators | Test Suite | SVG Overview

SVG Viewers

A number of stand-alone SVG Viewers exist. These can be though of as SVG-only browsers, in the same way as older browsers were HTML-only browsers. These SVG Viewers include an XML parser; a CSS parser; a CSS cascading, specificity and inheritance engine; and an SVG rendering engine to draw the graphics. They may offer print capabilities, in addition to display on screen.