

# Graphikprogrammierung in C++ Einführung

Marco Feuerstein

21 July 2006

chair for computer aided medical procedures

department of computer science | technische universität münchen

# Kursübersicht :: Graphikprogrammierung in C++

## Anwendung im Bereich Computer Graphics & Vision (Projektaufgabe)

Virtual Reality, Augmented Reality, 3D Spiel oder  
Special Effects

### OpenGL

- the rendering pipeline
- light /shading models
- simple geometric models

### Efficient software development

- IDE & debugging tools (Visual Studio 2005)
- version control tools (Subversion = svn)

### Qt

- advanced GUI and programming library

## Grundlegendes Wissen zur Programmiersprache C++

- Präprozessor, Compiler, Linker
- Funktionen, Klassen and Namensräume
- Speichermanagement: Zeiger & Referenzen
- STL (the standard template library) Generische Kontainer und Algorithmen
- Zeichenkette und Datenströme (strings and streams)

# Motivation :: Bildmanipulation



apply image filter



apply 2D homographies

# Motivation :: Special Effects



By Irfan Essa,  
Georgia Institute of Technology,  
Atlanta, Georgia



# Motivation :: Augmented Reality



# Organisatorisches :: Voraussetzungen für den Schein

- Einreichung der Aufgaben in Teamarbeit von jeweils zwei Teilnehmern bis spätestens Montag
- Mindestens 8 von 9 Aufgabenblättern wurden korrekt bearbeitet
- Aktive Mitarbeit bei den wöchentlichen Treffen
- Abgabe der Projektaufgabe
- Teamarbeit
- Bestehen eines 5min Testats

# Organisatorisches :: Zeitplan

- Wöchentliche Termine
  - **Montag 12:00** – Abgabe der Aufgaben
  - **Dienstag 16:00-18:00** – Wöchentliches Treffen
    - Besprechung des letzten Aufgabenblattes
    - Vortrag über neue Konzepte
    - Tipps & Tricks zum nächsten Aufgabenblatt
  - **Dienstag- Montag** – Aufgabenbearbeitung in den Teams
  - **Dienstag 16:00** – Neues Aufgabenblatt während des Kurses

# Organisatorisches:: Emails

- Tobias Sielhorst – [sielhors@cs.tum.edu](mailto:sielhors@cs.tum.edu)
- Christoph Bichlmeier – [bichlmei@cs.tum.edu](mailto:bichlmei@cs.tum.edu)
- Alle Teilnehmer und Betreuer  
[programmierpraktikum@mailnavab.in.tum.de](mailto:programmierpraktikum@mailnavab.in.tum.de)

# C++ vs JAVA

- Binärdateien statt Bytecode
- Programmierer muss sich um das Speichermanagement kümmern
- Code ist plattformunabhängig, Binärcode nicht
- Hardwarenah aber starke Abstraktionsmöglichkeiten; Fehlerchecks nur durch Compiler
- Schnell!
- Syntax
  - `if()`, `for()`, `while()`, Funktionsaufrufe gleich
  - `class:parentclass` statt `class extends parentclass`
  - `include` statt `import`