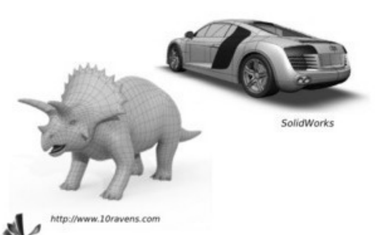
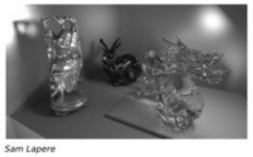
Application domains of CG

Modeling



Animation

Rendering



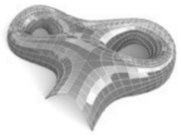
Applications of 3D

Emerging domain

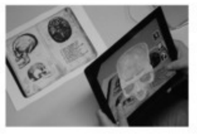
- Entertainment (Video Games, Cinema, ...)

- Simulation, Computing, Data Analysis

- Design, prototyping, reconstruction

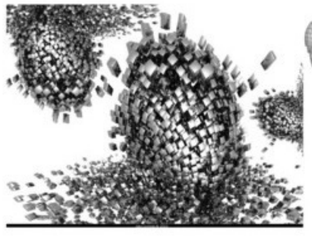


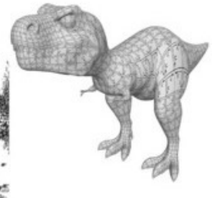
- Interaction (VR, Augmented Reality, ...)



Entertainment: Video Games

Entertainment: Cinema





Entertainment

Jobs areas

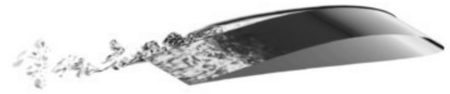
Development
C++
Shader/Rendering
GPU
Scripting
Automatization
Pipe-line

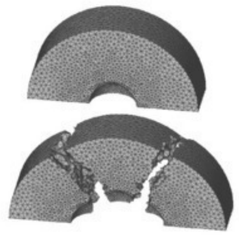
Internships

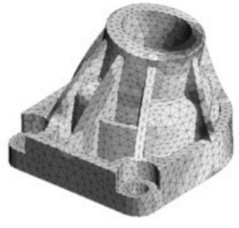
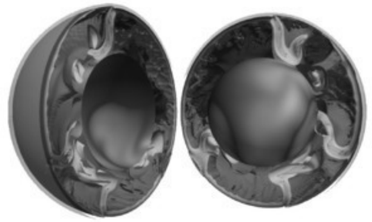
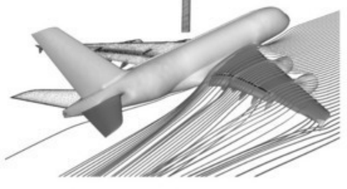
Occulsion computation
(GameLoft)
Android Dev.
(AppSolute)
Version management
(Asobo)
Non Playing Characters
(Ubisoft)
Dev. The Crew
(Ivory Tower)

UBISOFT ASOBO STUDIO appsolute Ivory Tower GAMELOFT

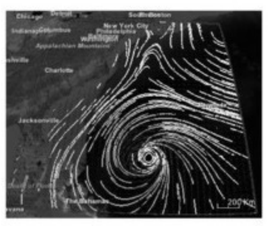
Simulation: Computing

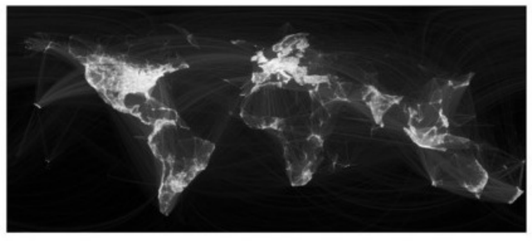
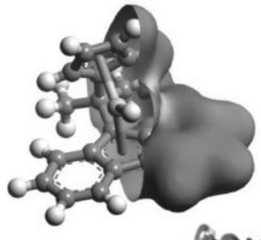


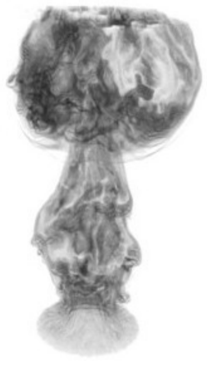
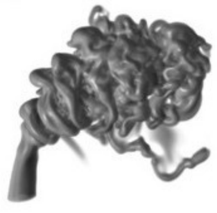


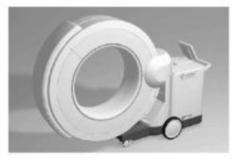
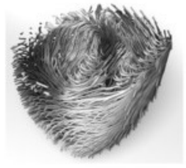
Simulation: Scientific visualization



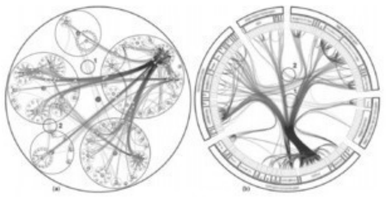
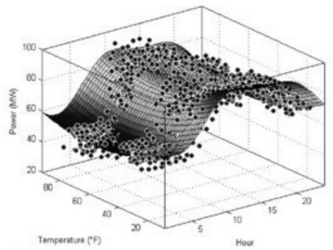
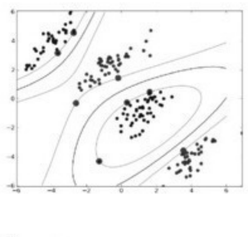




Simulation: Medical Imaging

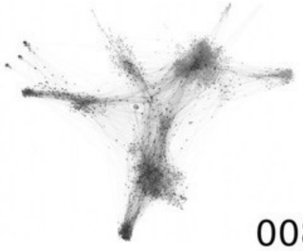






Data Analysis: Big Data

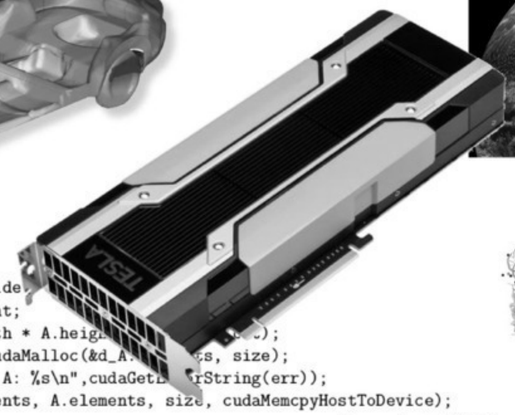
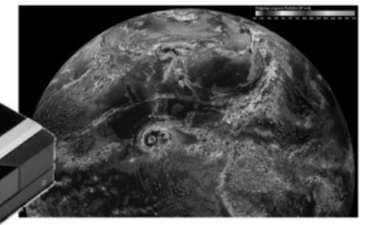
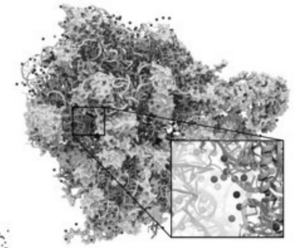






008

Data analysis: GPGPU

```
Matrix d_A;
d_A.width = d_A.stride
d_A.height = A.height;
size_t size = A.width * A.heig                    );
cudaError_t err = cudaMalloc(&d_A.                s, size);
printf("CUDA malloc A: %s\n",cudaGet              rString(err));
cudaMemcpy(d_A.elements, A.elements, size, cudaMemcpyHostToDevice);

Matrix d_B;
d_B.width = d_B.stride = B.width;
d_B.height = B.height;
size = B.width * B.height * sizeof(float);
err = cudaMalloc(&d_B.elements, size);
printf("CUDA malloc B: %s\n",cudaGetErrorString(err));
```

009

Simulation, Computing, Data Analysis

Jobs areas

Computing
Physics/medical domain
Data Analysis
Optimization
Scientific visualization
(scalar/vector fields, ...)
Big Data
GPU

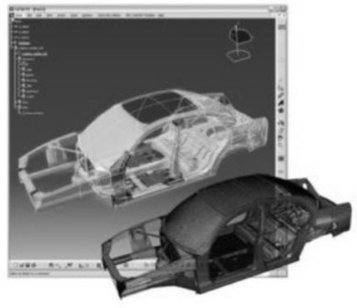
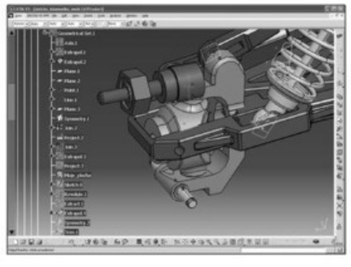
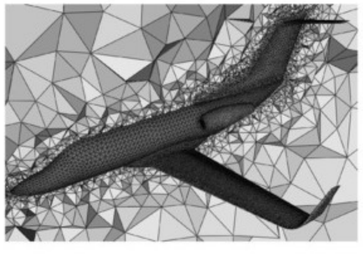
Internships ex.

Brain imaging
(Scient. Comp. II)
3D tooth visualization
(Michigan Univ.)
3D Visualization
(GE Healthcare)
Registration
(Michelin)
Data analysis
(L'Oreal)
Hair Simulation
(LIRIS)

SCI www.sci.utah.edu · GE Healthcare · L'ORÉAL · MICHELIN · LIRIS

010

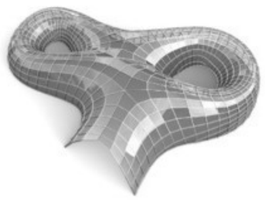
Design: CAD

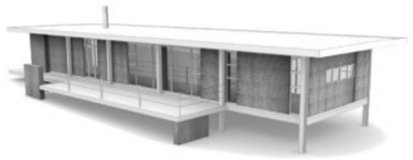






011

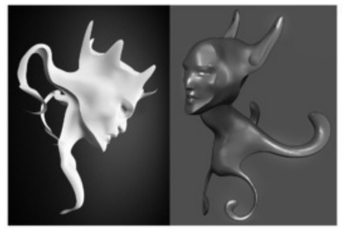
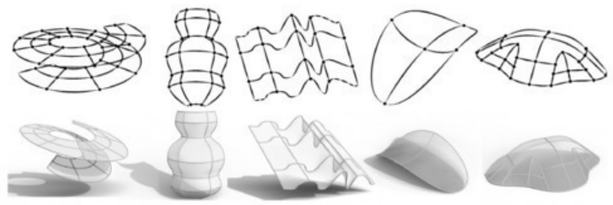
Design: Architecture

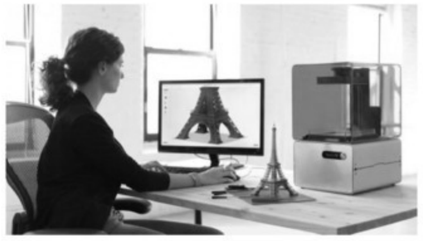
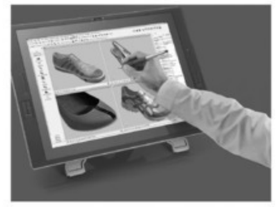







Design: Prototyping



Design: 3D Reconstruction



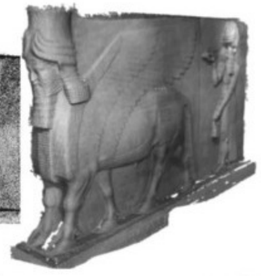




Design: Cultural heritage





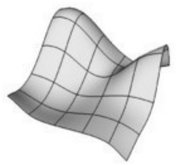





Design

Jobs areas

Mesh
Lasers
Reconstruction
Interaction, HCI
Mecanics
Curve & Surfaces



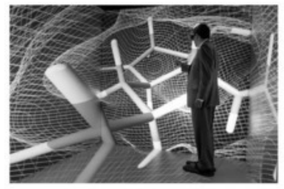
Internships ex.

3D Reconstruction
(Kagoshima Univ.)
Virtual sculpting
(LIRIS)
3D scenes
(Technodigit)
3D video capture
(Thales)
Collision handling
(Delcam)
3D Modeling
(Rauscher)
3D Glass
(Rodenstock)

鹿児島大学 Kagoshima University · Technodigit · LIRIS · LOHMANN & RAUSCHER · Delcam · RODENSTOCK · THALES

Interaction: Virtual Reality (VR)







Interaction: Augmented Reality







Interaction

Jobs areas

3D Reconstruction
Fast simulation
HCI
Lasers
Smartphone dev.

Internships ex.

Aug. Reality for Medical
(Kitware)
Aug. Reality for city modeling
(Wikitude)
Sketching Software
(Dassault)
Virtual city
(LIRIS)
A350 air system modeling
(Airbus)
Android aug. reality system
(CEA)

Kitware · AIRBUS · DASSAULT SYSTEMES · cea · wikitude

Overview of the lectures

3D Modeling

3D modeling methods
- Explicit approaches (mesh,parametric,subdivision)
- Implicit approches (voxels, parametric, point sets)
- Fractals

Smooth surface & differential geometry

Meshes
- Structure
- Data structure and halfedge, CGAL
- Subdivision algorithm
- Mesh smoothing

Computational geometry
- Convex Hull
- Delaunay triangulation

Animation

Physically Based Animation
- Numerical solution to ODE (explicit, implicit approaches)
- Cloth simulation
- Collision detection structures

Geometric Surface Deformation
- Rigid transformations
- Skinning

Rendering

Ray tracing
- Principle
- Common objects (plane, sphere, implicit surfaces)
- Effects (shadowing, caustics, etc)

Introduction to OpenGL
- Drawing a triangle
- Passing uniform argument
- Interpolation
- Shading