

# 3D Computer Vision

## Project Presentation BLOCK MATCHING

Diesner,	Erik
Enderling,	Heiko
Gumz,	Mathias
Hoerning,	Ronny
Krueger,	Franziska
Schumann,	Christian

# Talk outline

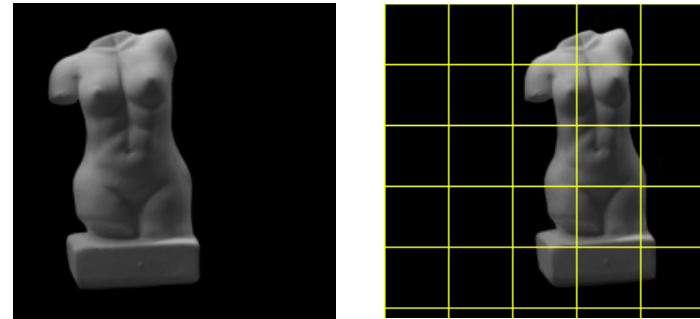
- Block Matching – basic idea
- First approach using IDL
- Second approach using C++ / FOX
- Results
- Summary

# Block Matching – Basic idea

- Two images, taken from a scene from different camera positions
- Solving the correspondence problem ( finding conjugate pairs )
- Disparity map generation
- Depth map generation
- Mapping the 3D surface on a scene

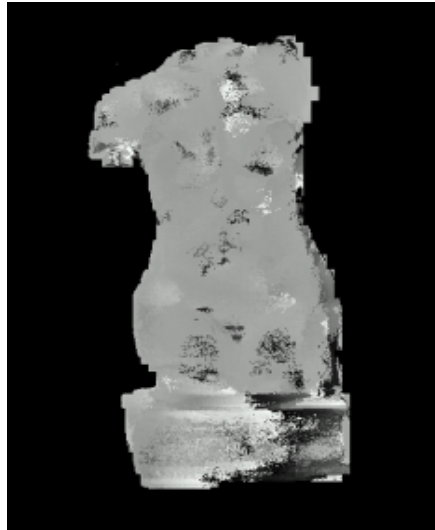
# Block Matching – Basic idea

- divide right image into blocks
- for each block try to find block with lowest difference in the left image

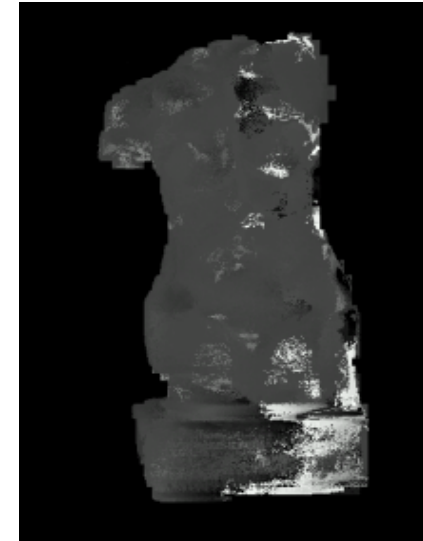


- save average difference as disparity in a block disparity map
- for each pixel in right image try to find the corresponding pixel in the left image
- use block disparities to select the area where to search

=> disparity map



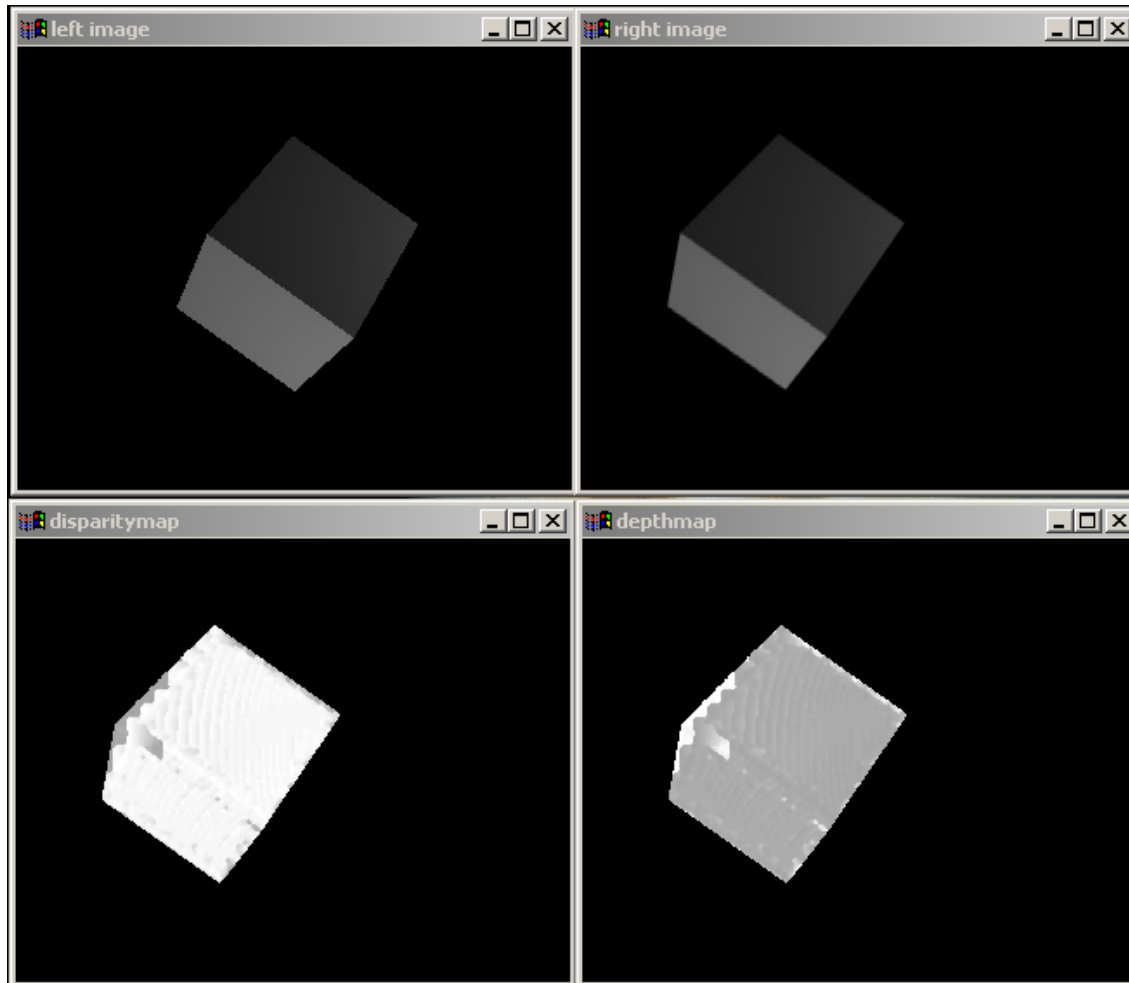
=> depth map



# First approach in IDL

- Why IDL ?
  - very fast array computations
  - many algorithms already implemented
  - just to get started
- Results ?
  - not as satisfying as expected
  - complicate to handle GUI builder
  - problems with OpenGL support

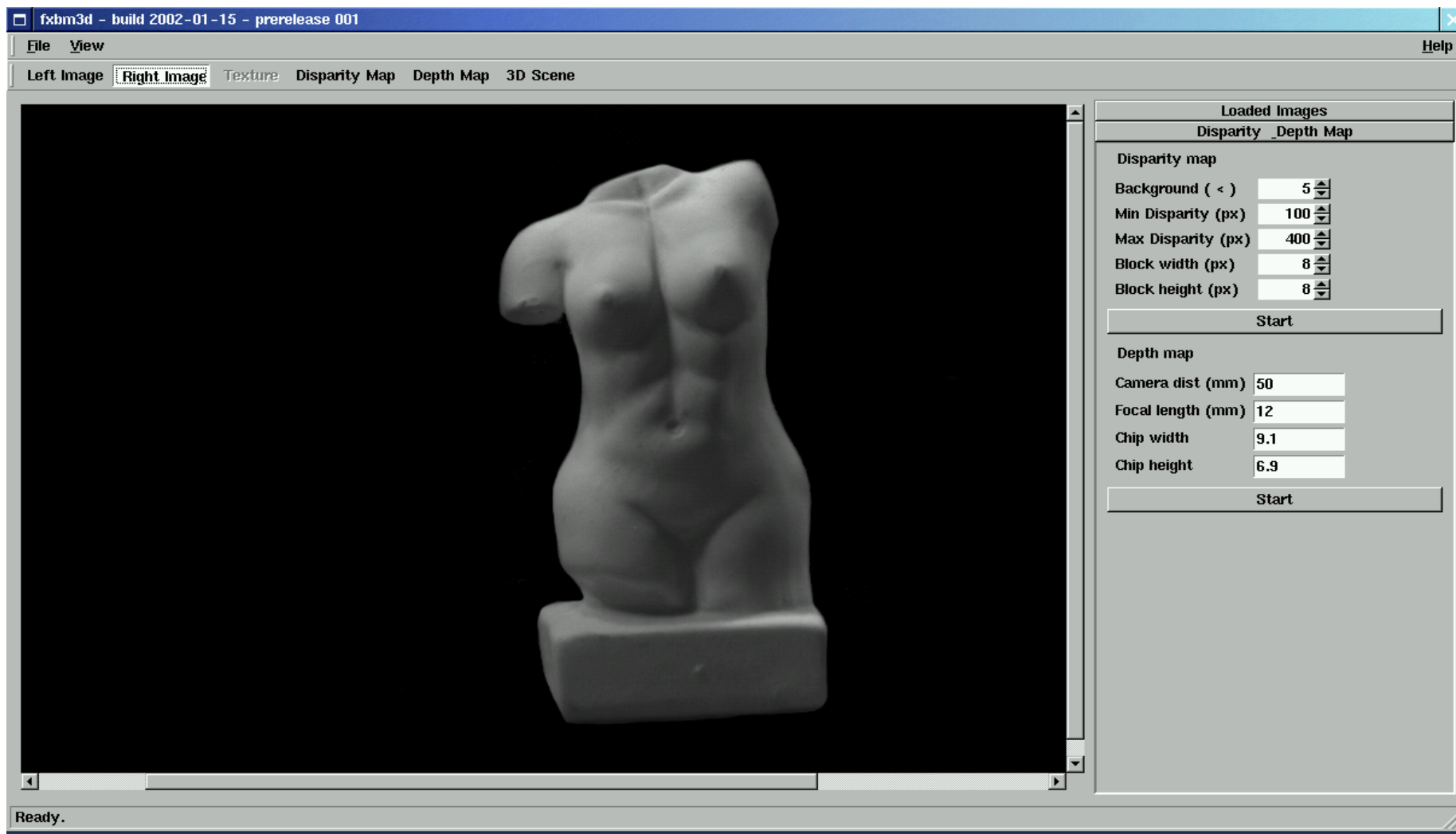
# Results using IDL



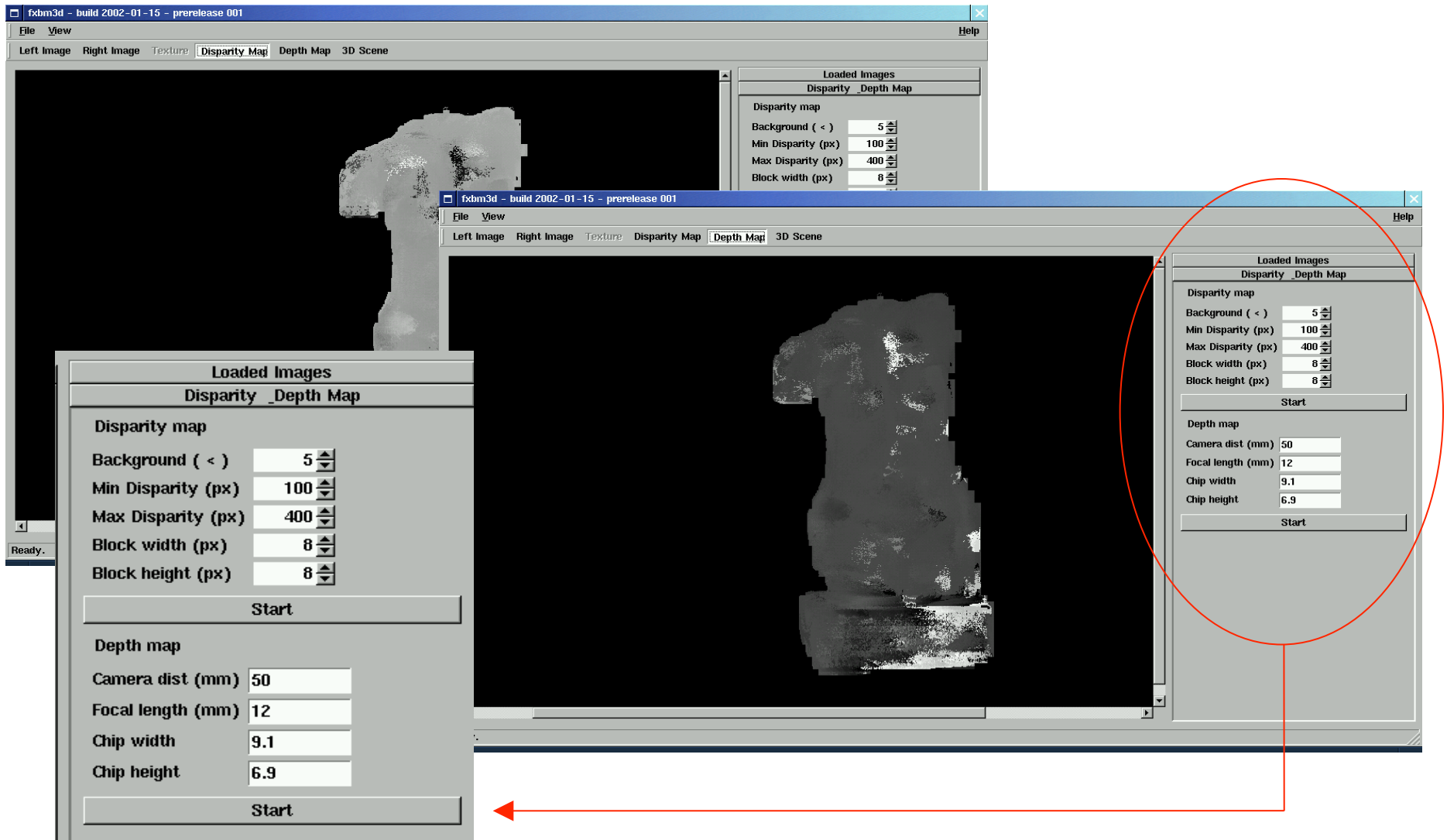
# Second approach using C++/FOX

- Why convert ?
  - problems with own implemented functions
  - general problems using IDL
- Why C++ / FOX ?
  - fundamental knowledge of the APIs
  - OpenGL support
  - easy to use

# Results

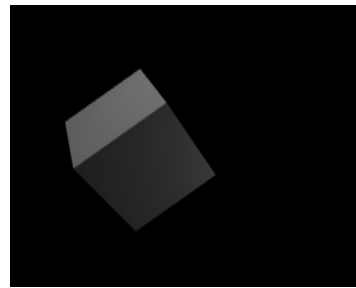
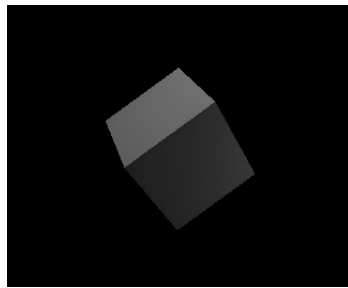


# Disparity and depth map

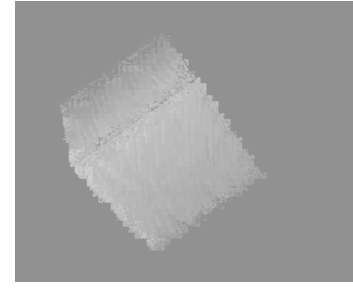
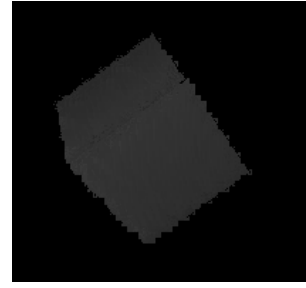


# Problems with taken images

- noise and distortions lead to errors
- generated images to check the algorithms

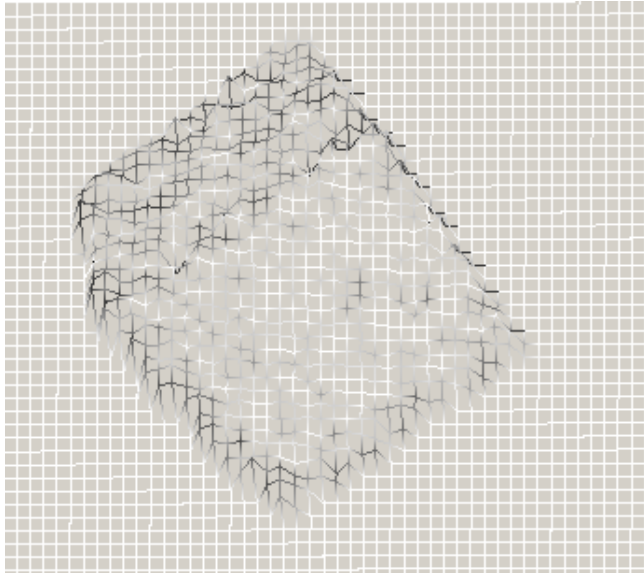


input images, left and right

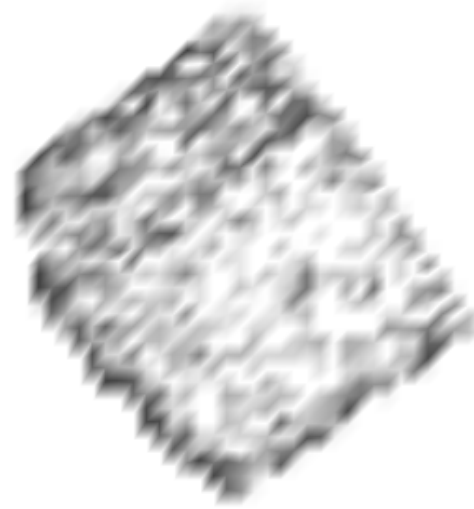


disparity map, depth map

# 3D shape

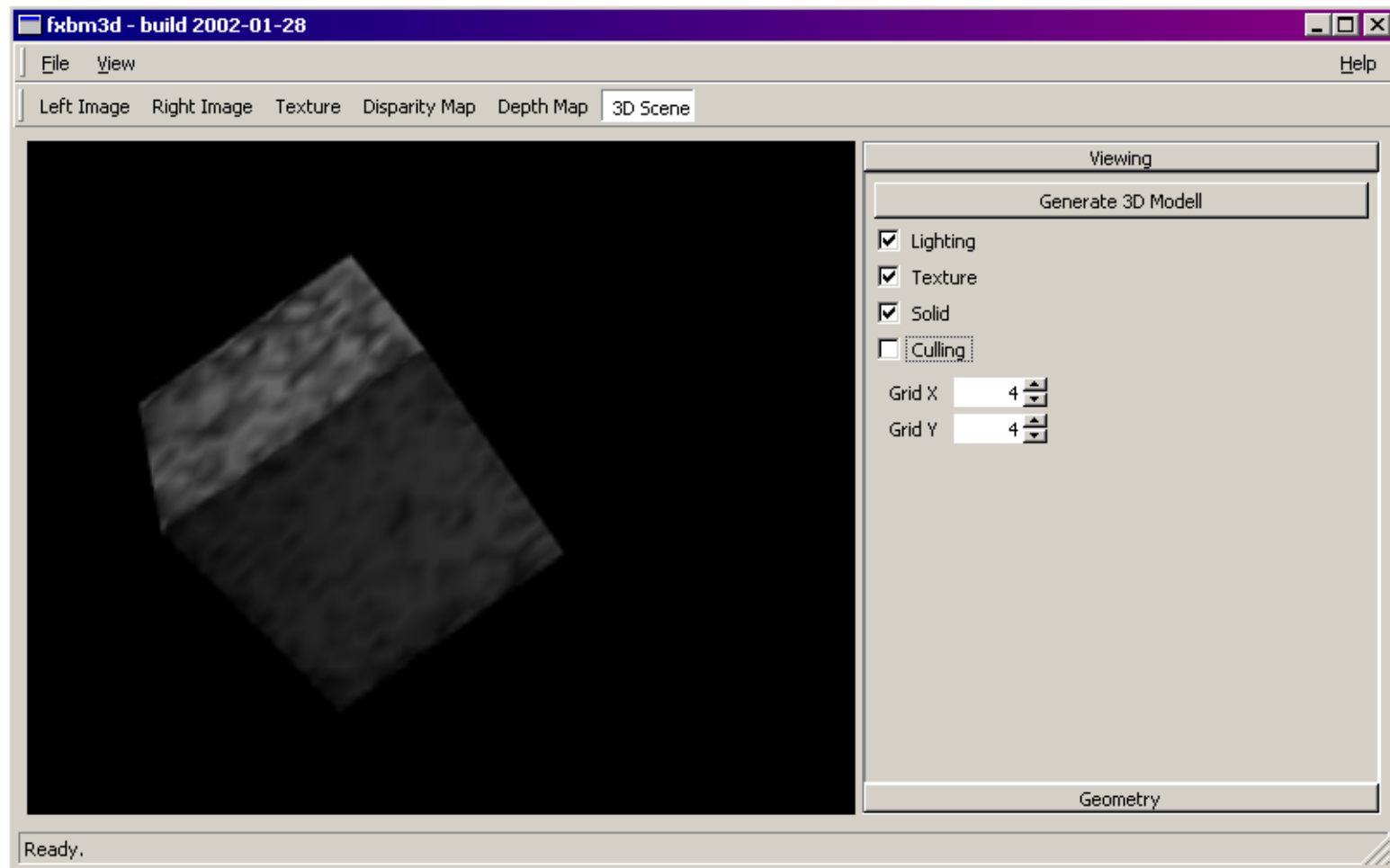


3D polygon mesh

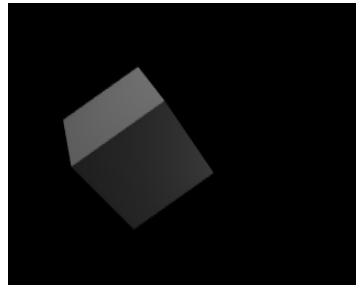
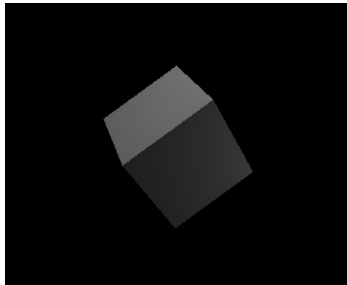


3D solid object

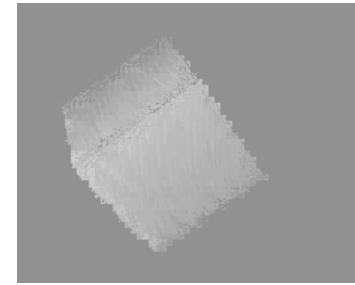
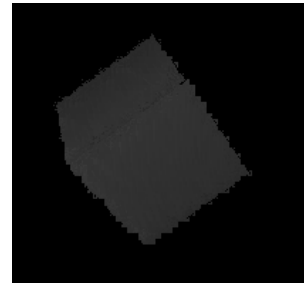
# Final solution



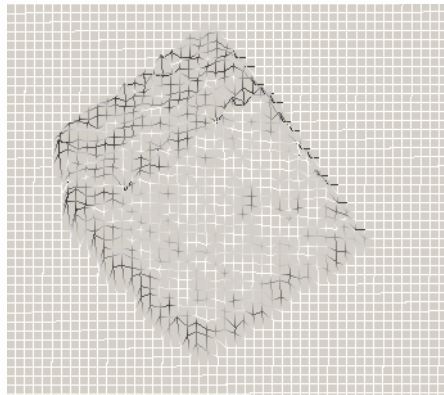
# Step by step



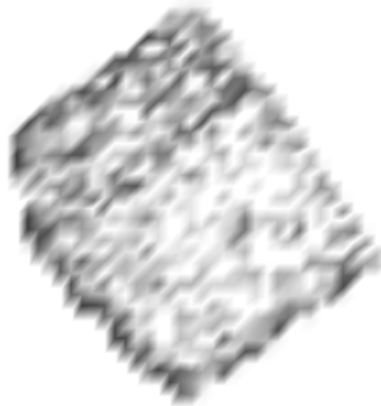
input images, left and right



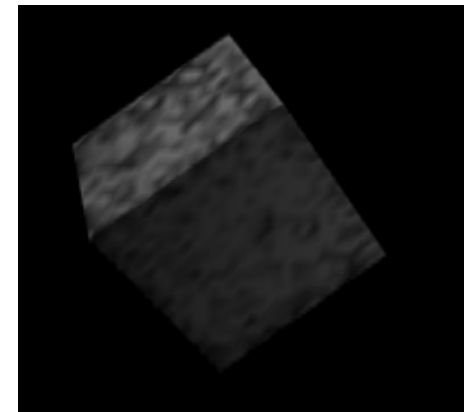
disparity map, depth map



3d mesh,

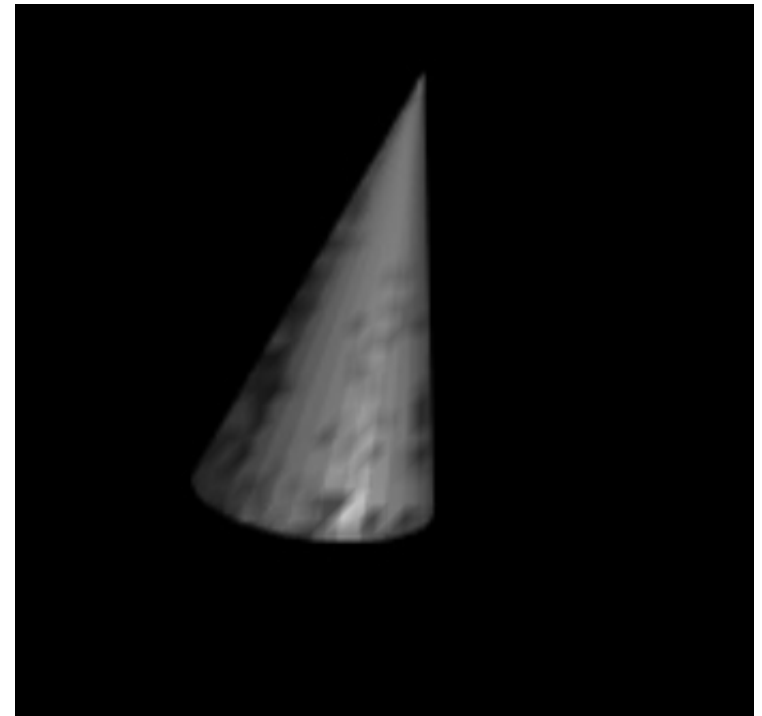
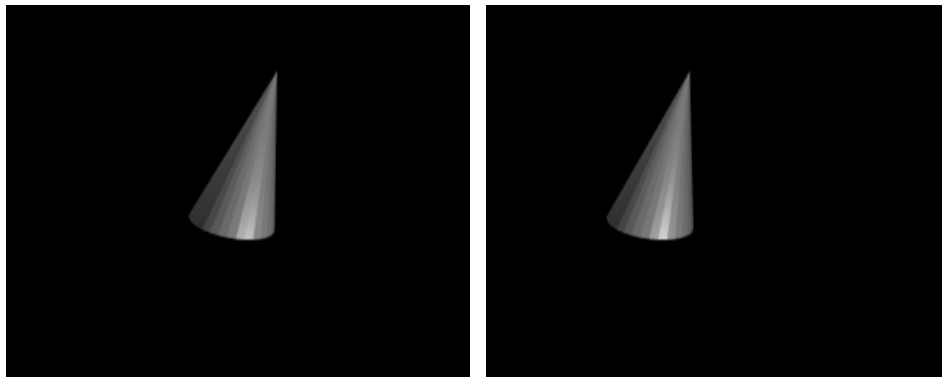


solid object,



textured

# Different input data



# Summary

- taken images not the best choice
- manual segmentation to improve the results
- rendered images to check algorithms
- converting to C++ helpful in order to find errors in implemented functions
- quite a lot of errors even after filtering