3D Camera for a Cellular Phone

Deborah Cohen & Dani Voitsechov
Supervisor : Raja Giryes

2010/11

Contents

Why 3D ?

Nowadays, more and more applications need real-time, accurate, low cost 3D scanning :

- Entertainment (video games, movies)
- Objects recognition
- Medical imaging
- Robot navigation systems…

3D Techniques

Some techniques that exist today :

- Time-of-flight 3D laser scanner
- Triangulation
- Passive stereometric scanners
- Active stereometric scanners

Our technique :

Active stereometric → Structured Light → Time-multiplexed

Project definition and goals

- Main goal :

Miniaturization of the camera for a cell phone

- Steps :
  - implementation of a control system for the camera and the projector
  - synchronization (between the camera and the projector)
  - patterns projection and images capture
  - off-line calibration
  - reconstruction
  - optimizations

- Objectives : real-time, accuracy, low cost

Projective model of a structured light system

- Assuming a pin-hole model
  for the camera and the projector

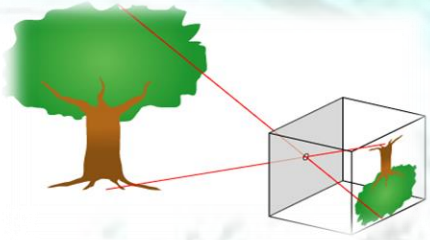


Figure 1 – Pin-hole model

- Camera : 2 coordinates - line
- Projector : 1 coordinate - plane

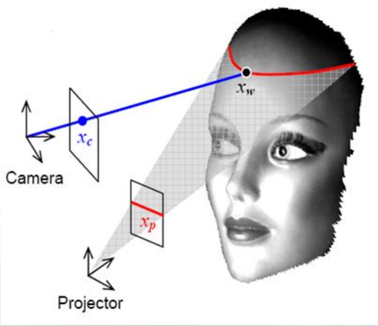


Figure 2 – Camera and projector planes vs. object plane

Projective model of a structured light system (cont.)

Forward projection : $T : X_w \rightarrow \left(X_c, X_p \right)$

Backward projection : $T^{-1} : \left(X_c, X_p \right) \rightarrow X_w$

X_w – homogenous world coordinate system

X_c – homogenous camera coordinate system

X_p – homogenous projector coordinate system

C_c – camera perspective projection matrix

C_p – projector perspective projection matrix

The matrices describe the intrinsic and extrinsic properties of the camera and the projector.

Projective model of a structured light system (cont.)

- Transformation of world coordinates to camera coordinates :

$$X_c = C_c X_w \quad \text{where} \quad C_c = \alpha \begin{bmatrix} f_x & kf_y & x_c^0 \\ 0 & f_y & y_c^0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_c & t_c \end{bmatrix}$$

- Transformation of world coordinates to projector coordinates :

$$X_p = C_p X_w \quad \text{where} \quad C_p = \alpha \begin{bmatrix} f_p & 0 & x_p^0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R_p & t_p \end{bmatrix}$$

Algorithm

Phase 1 – Calibration

- Determining the PPMs: C_c and C_p

Phase 2 – Decoding

- Computing x_p

Phase 3 – Reconstruction

- Finding the world coordinates x_w

Algorithm - Decoding

- Input : $I_H -$ fully illuminated image, $I_L -$ fully darkened image, $I_1, ..., I_N -$ N coded images
- Output : $x_p -$ projector coordinate for each pixel

- Stages :
  1. Normalization : $J_k(x,y) = \dfrac{I_k(x,y) - I_L(x,y)}{I_H(x,y) - I_L(x,y)}$
  2. Binarization : $B_k(x,y) = \begin{cases} 1 & J_k(x,y) > threshold \\ 0 & else \end{cases}$
  3. Decoding : $x_p = \sum_{k=1}^{N} 2^{-k} B_k(x,y)$

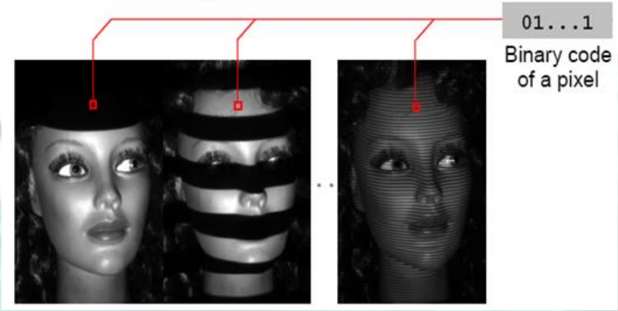


Figure 4 – Projected coded light patterns

- Fixed point algorithm :
  - look-up table for normalization
  - shift operation instead of division (power of 2)

Algorithm - Reconstruction

- Input : $x_p, x_c, y_c -$ projector and camera coordinates
- Output : $x_w -$ world coordinates

- Back projection : $\underline{x}_w = [x_w, y_w, z_w]^T -$ non-homogenous world coordinates

$$x_w = -R^{-1}s \qquad \text{where} \qquad Q = \begin{bmatrix} x_c c_3 - c_1 \\ y_c c_3 - c_2 \\ x_p p_2 - p_1 \end{bmatrix} = [R, s]$$

$c_k, p_k -$ k-th row of C_c and C_p respectively

- Fixed point algorithm :
  - using homogeneous coordinates (fractions, scaling)

Scheme of object reconstruction

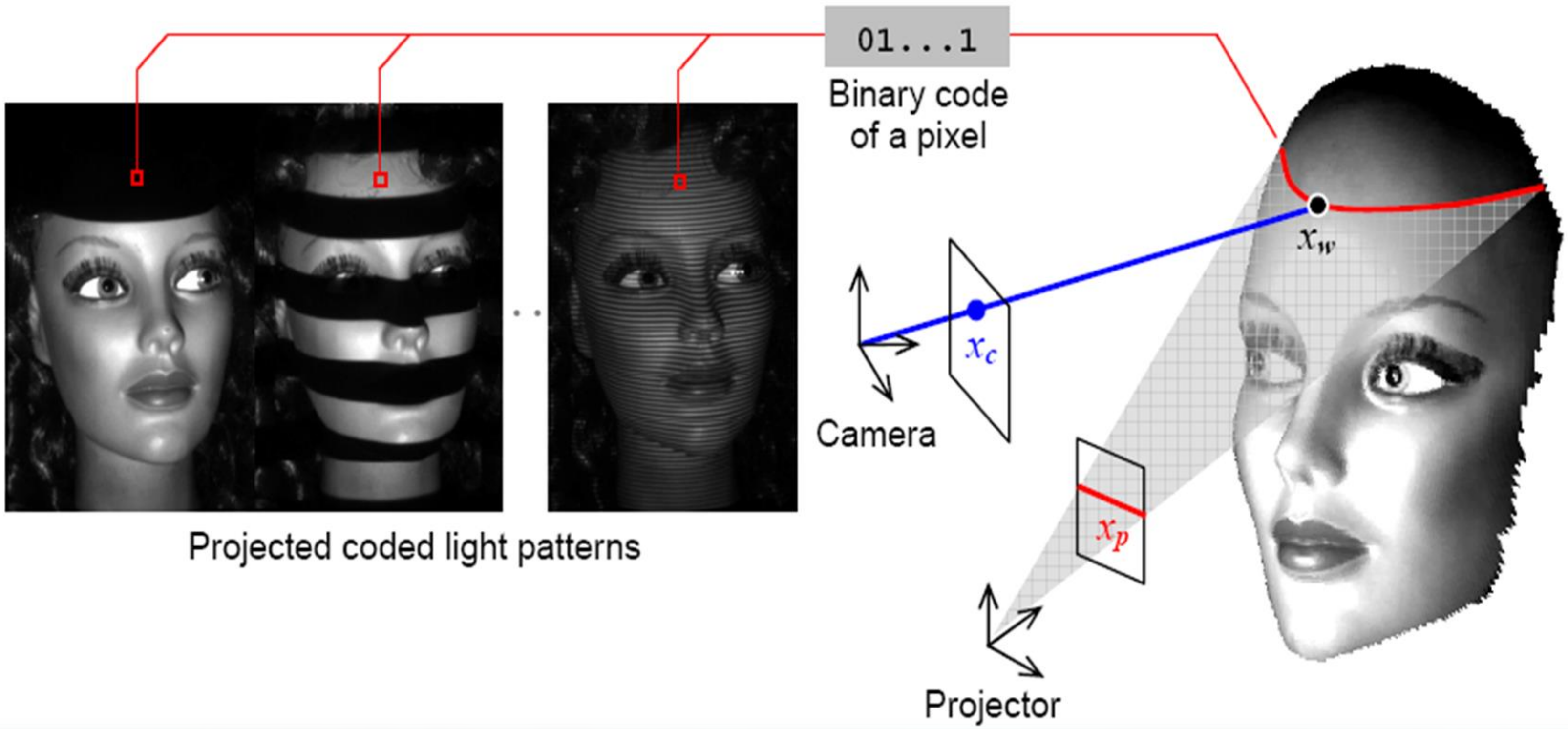


Figure 5 – Scheme of three-dimensional object reconstruction in a coded light scanner system

Algorithm - Calibration

- Input : set of measured $\left\{\left(x_c, x_p\right)\right\}_{n=1}^{N}$ and corresponding known $\left\{x_w\right\}_{n=1}^{N}$
- Output : T^{-1} – calibration data

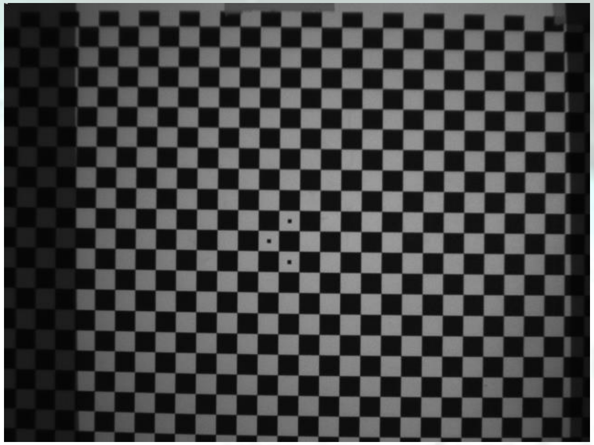


Figure 3 – Corner detection

- Minimize squared error between known fiducial points in World Coordinate System (WCS) and the backward projected measurements :

$$T = \arg\min \sum_{i=1}^{N} \left\| T^{-1}\left(x_c, x_p\right)_k - \left(x_w\right)_k \right\|_2^2$$

- Solving by numerical global optimization methods

BeagleBoard

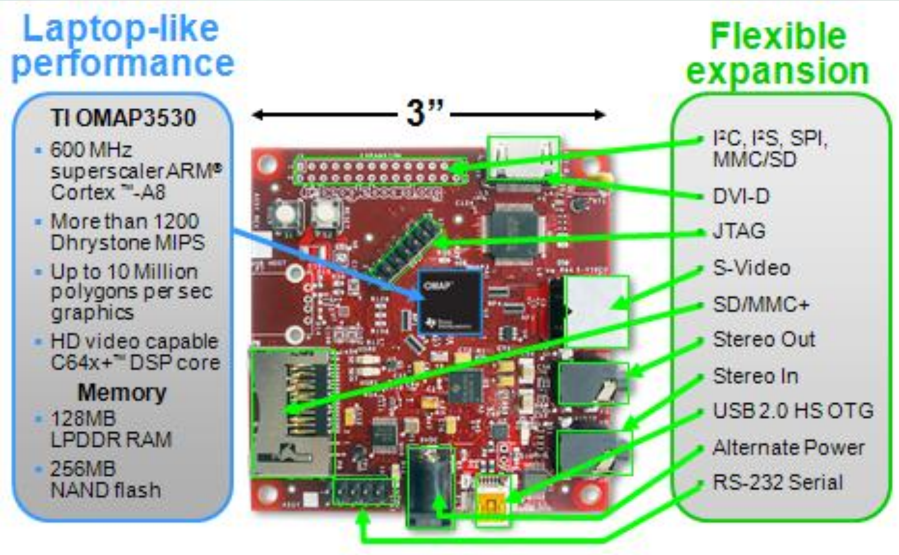


Figure 9 – BeagleBoard described

- 3'' x 3''
- POP (package on package)
  - CPU/Memory chip
- Processor TI OMAP3530 :
  - ARM cortex-A8 CPU (600 MHz)
  - TMS320C64x+ DSP (up to 430 MHz)
  - Imagination Technologies PowerVR SGX530 GPU
- Memory chip :
  - External : 128 MB LPDDR RAM memory & 256 MB NAND Flash memory
  - Internal : L1 - 112 KB (DSP),32 KB (ARM) , L2 - 96 KB (DSP),256 KB (ARM)
- Peripheral connections
  - DVI-D, USB, SD/MMC, S-Video, Stereo audio
- Development
  - Boot from NAND memory, SD/MMC, USB
  - Using Angstrom Linux

http://focus.ti.com/docs/prod/folders/print/omap3530.html

I/O devices

Camera

Logitech QuickCam Pro 9000

Figure 10 – Camera

Projector

TI DLP Pico projector

Figure 11 – Projector

System

Figure 6 – System size compared to iPhone

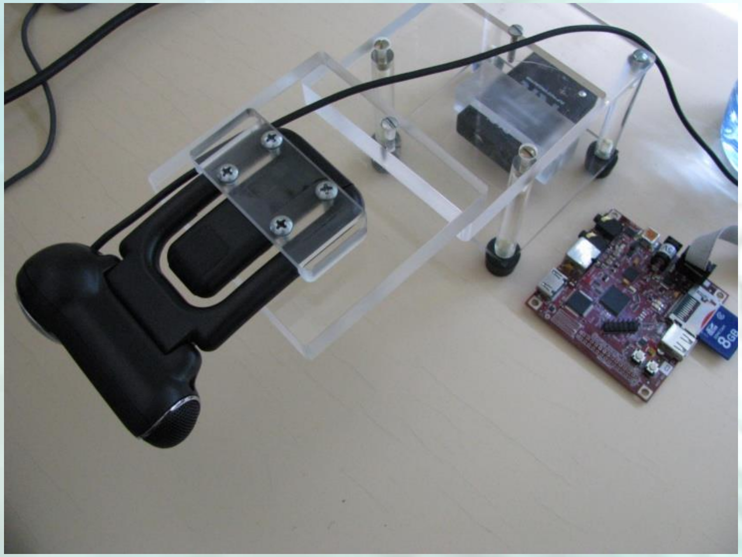


Figure 7 – Top view

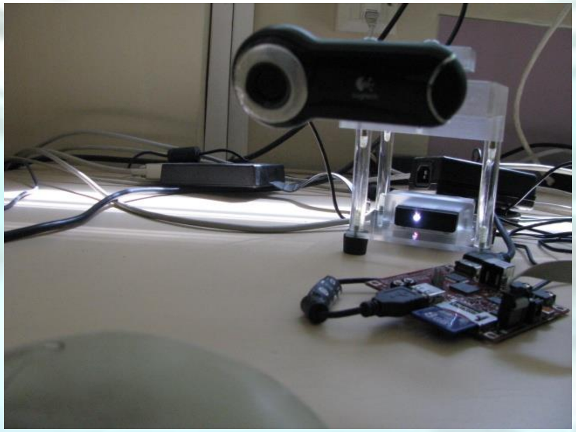
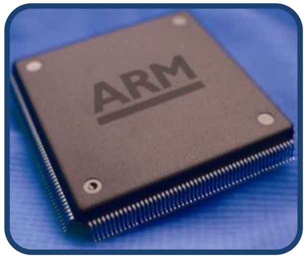


Figure 8 – Front view

Tasks division in the BeagleBoard



ARM

- controls DSP & GPU
- controls the communication with the camera and the projector

DSP

- runs fixed point reconstruction algorithm

GPU

- runs fixed point reconstruction algorithm

Tasks division in the BeagleBoard (cont.)

GPU uses the projector to illuminate the scene with 1-dimensional light patterns (9 patterns, Gray code)

↓

ARM uses the camera to capture the patterns

↓

Captured data is sent to the DSP

↓

DSP performs 3D reconstruction

↓

DSP returns the results to the ARM

Tasks division

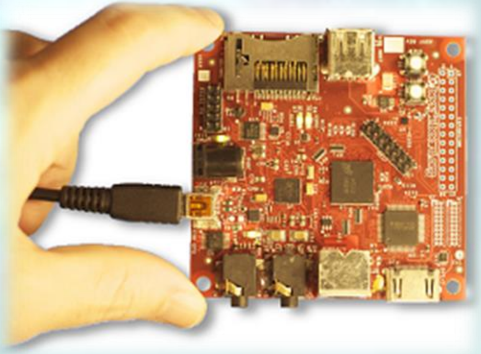


Figure 12 - BeagleBoard

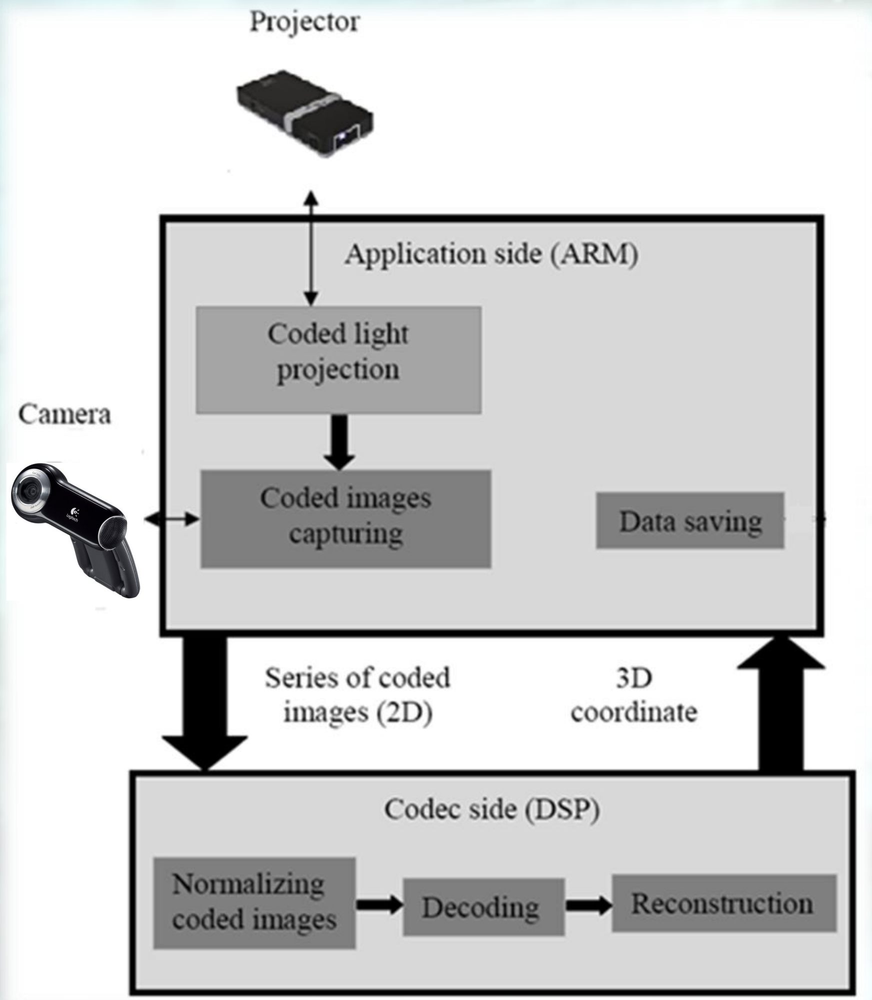


Figure 13 – Division of tasks between the ARM and the DSP

Step 1 - Patterns projection

OpenGL for Embedded Systems:

premier environment for developing portable, interactive 2D and 3D graphics applications

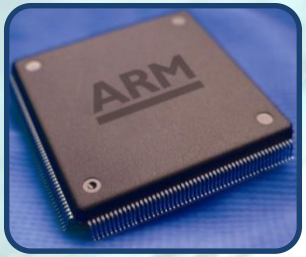
- Draw triangle arrays (2 triangles per stripes):

  void glDrawArrays(GLenum mode, GLint first, GLsizei count)

Step 1 - Patterns projection (cont.)

Figure 14 – Coded light projected by the pico projector

Step 2 - Pictures capture



SDK for UVC based cameras :

video for Linux to UVC (v4l2uvc)

- Grab frame in yuyv format through the ARM :

int uvcGrab (struct vdIn *vd)

- Store the y coordinate (grayscale picture)

Checking - Image projection

ES

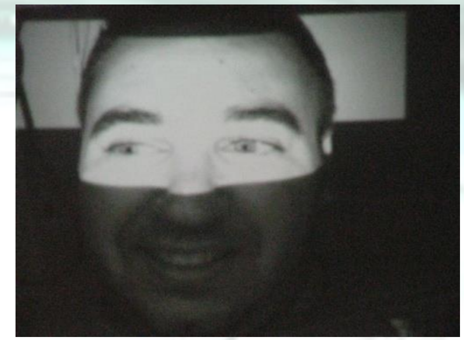


Figure 16 – Projection of a captured picture

- Using interleaved vertex data to match the texture coordinates with the triangle coordinates
- Bind texture (picture) to 2 triangles which cover the whole display (projected picture)

void glVertexAttribPointer (Gluint index, GLint size, GLenum type, GLboolean normalized, GLsizei stride, const GLvoid * pointer)

Step 3 - ARM to DSP / DSP to ARM

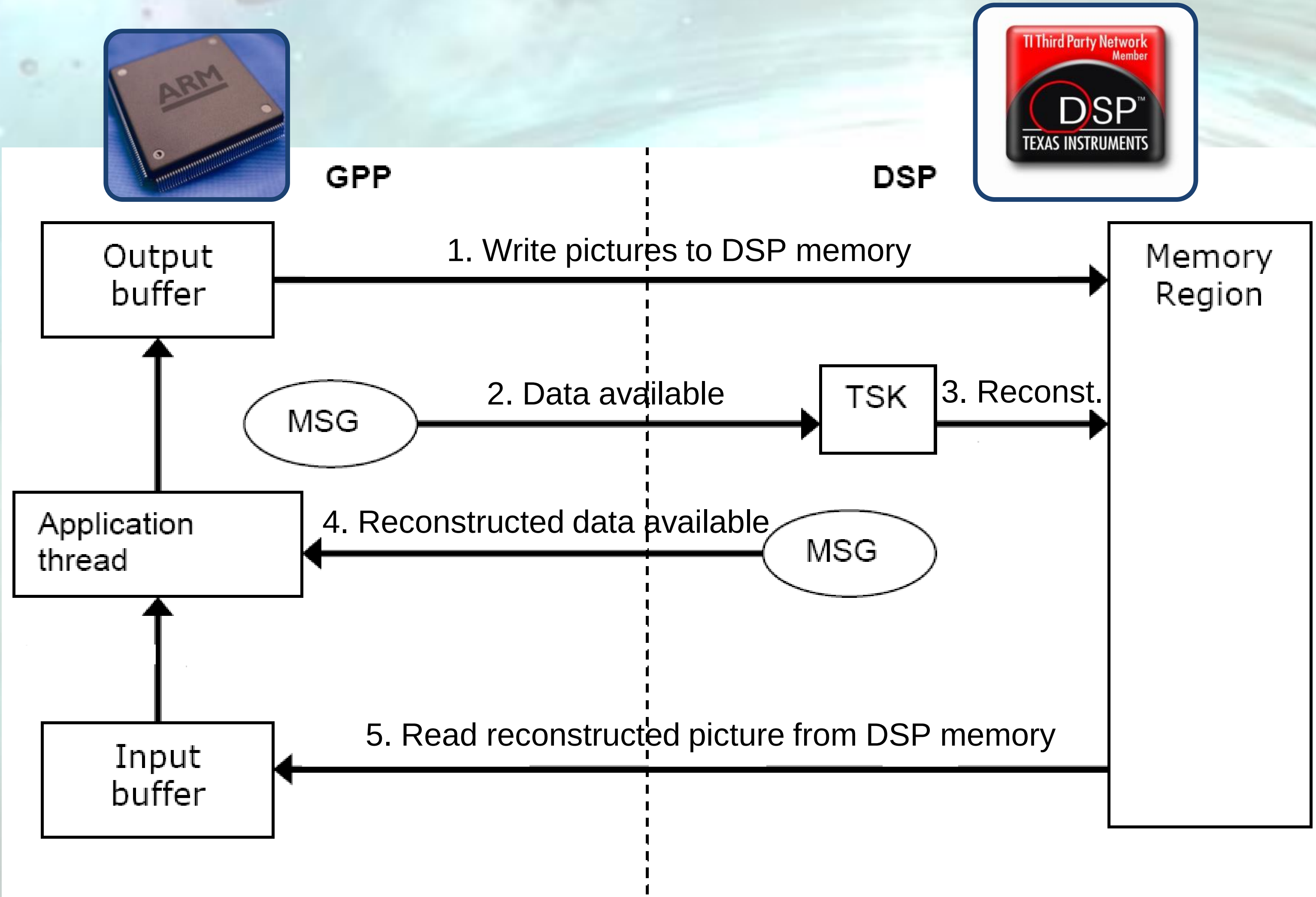


Figure 15 – Communication flow between ARM and DSP

Step 3 (2) - ARM tasks

- Inputs : calibration matrices, 9 pictures

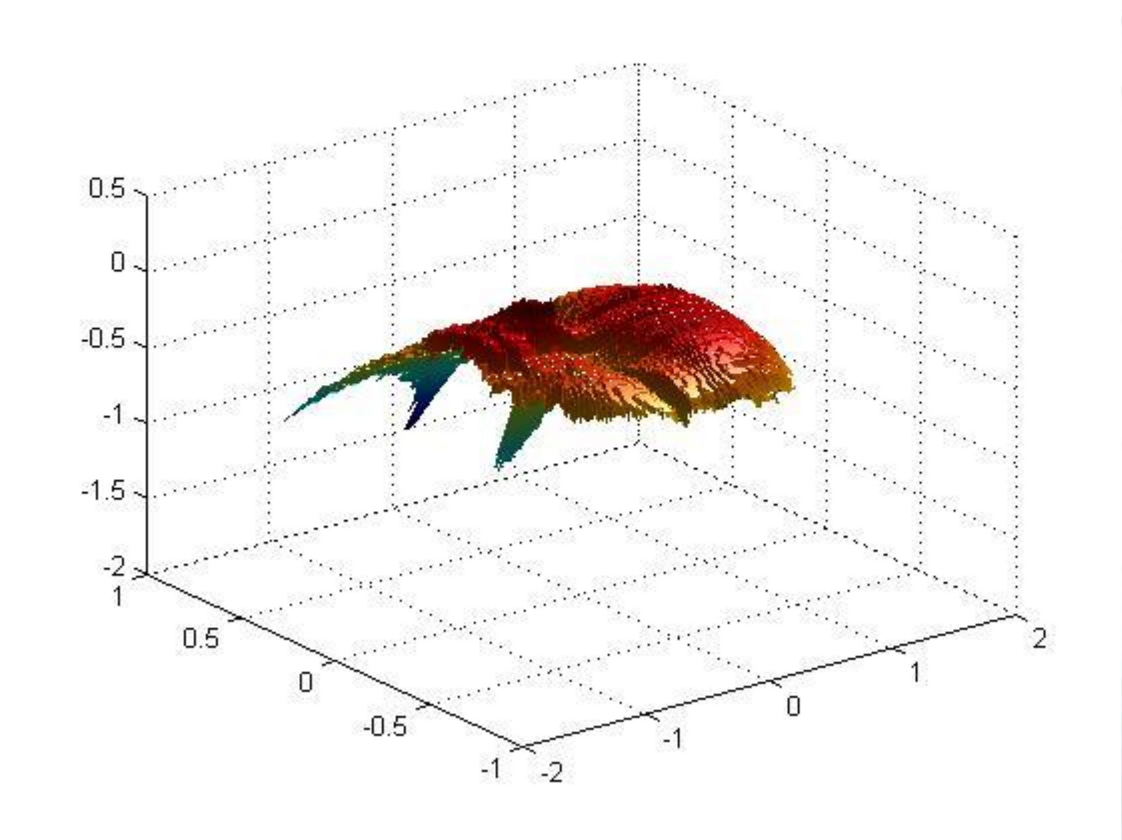
1. Write pictures to read buffer
2. Transform calibration matrices to calibration parameters (written in message)
3. Send message

Step 3 (2) - DSP tasks

- Inputs : calibration parameters, 9 pictures

1. Perform decoding & reconstruction
2. Write 3D homogeneous coordinates to write buffer (X,Y,Z,W)
3. Send message

Result



Using surf function from Matlab

Time analysis

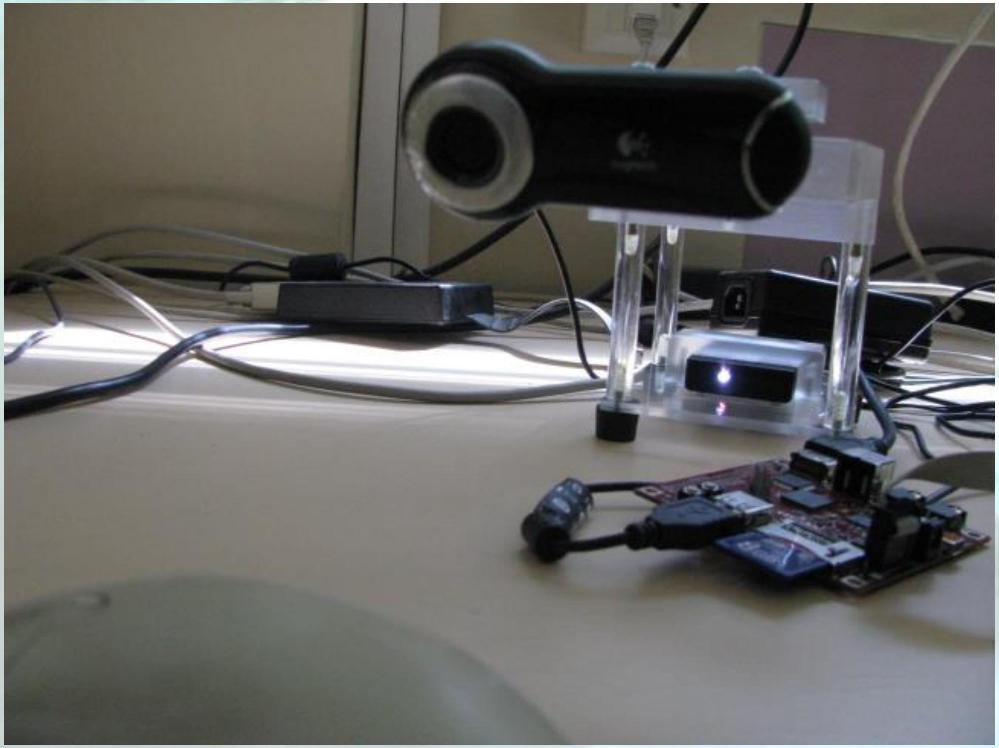
TASK	TOTAL TIME (sec)	EFFECTIVE TIME (sec)
Patterns projection + pictures capture	23.78 (*)	5.78 (**)

(*) Sync in software (sleep)

(**) Without sleep time

TASK	TOTAL TIME (sec)	RECONSTRUCTION TIME (sec)
ARM reconstruction	1.31	1.19
DSP reconstruction	0.50	0.24

Demo



Advantages and drawbacks

Advantages

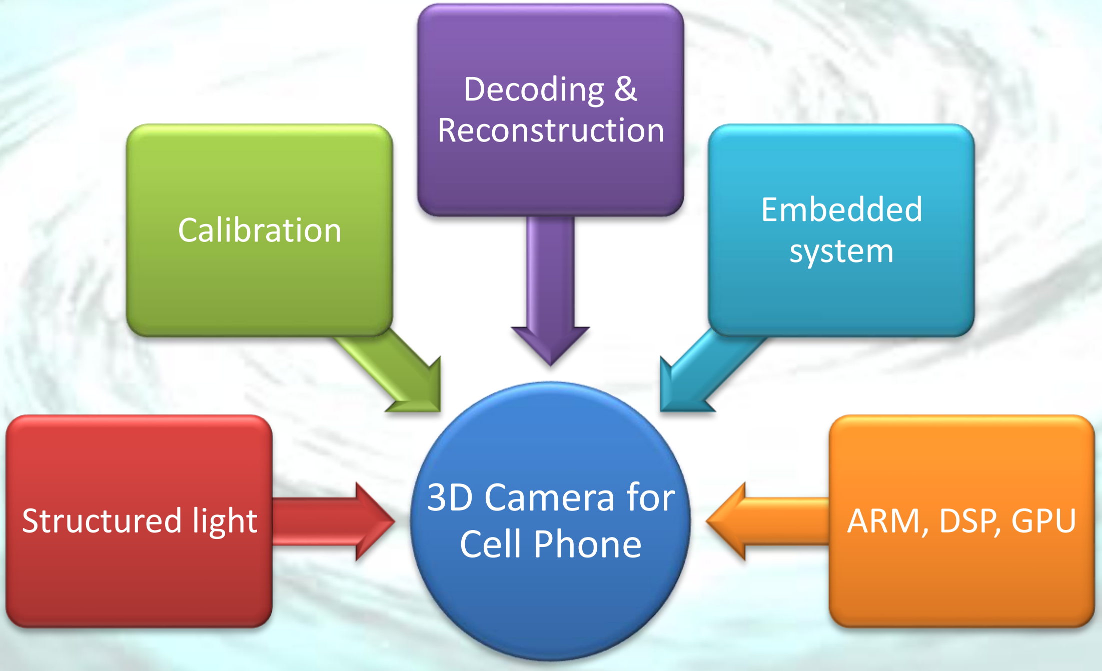
- Accuracy
- Real time (tasks division between the processors)
- Compatibility (Linux)
- Low cost
- Low power

Drawbacks

- Suitable only for scenes with low to medium motion
- Calibration relatively long (but performed once)
- Reflective or transparent surfaces raise difficulties

Summary

- 3D
- Structured light
- Calibration
- Decoding
- Reconstruction
- Embedded system
- BeagleBoard
- Camera and projector
- Patterns projection
- Picture capture
- DSP ⟺ ARM ⟺ GPU
- Image projection



Future work

- Accelerate the projections and captures using devices with hardware synchronization
- Consider other real-time platforms
- Use cellular phone camera
- Optimizations

Thanks

We want to thank the SIPL and GIP staffs, and particularly our supervisor Raja, and also Pavel and Alon who were always ready to help us even after dark.

QUESTIONS ?