3D Reconstruction & Scanning

http://graphics.stanford.edu/projects/mich/

The Digital Michelangelo Project 1997–9

lbg@dongseo.ac.kr
Agenda

- Multiple View Geometry
- Depth Camera
- Structured Light
- Camera Calibration
- Projector Calibration

http://www.ten24.info/
http://www.youtube.com/watch?v=j4waCVRgBWI
http://www.youtube.com/watch?v=Wgp0Cg3UFA4
Figure 1.3: Reconstruction of three-dimensional point through triangulation.
- Projective Transformations
- Camera Calibration
- Epipolar Geometry
- Feature Points
- Correspondence Search
- RANSAC Algorithm
- 3D Reconstruction
- SIFT & ASIFT
Scale Invariant Feature Transform

Scale-invariant feature transform (or SIFT) is an algorithm in computer vision to detect and describe local features in images. The algorithm was published by David Lowe in 1999.

Applications include object recognition, robotic mapping and navigation, image stitching, 3D modeling, gesture recognition, video tracking, and match moving.

The algorithm is patented in the US; the owner is the University of British Columbia.

David Lowe
Computer Science Department
University of British Columbia
Depth Camera

lbg@dongseo.ac.kr
Microsoft Kinect

- Motion sensing input device by Microsoft
- Depth camera tech. developed by PrimeSense Invented in 2005
- Software tech. developed by Rare
- First announced at E3 2009 as “Project Natal”
- Windows SDK Releases


lbg@dongseo.ac.kr 4/28/2015
ToF 3D Camera

Time of flight cameras

3DV ZSense
- Infrared camera + GaAs solid state shutter
- Pulsed infrared lasers
- RGB camera

3DV, Canesta (no-longer public)
PMD Technologies [http://www.PMDTec.com](http://www.PMDTec.com)
Mesa Technologies [http://www.mesa-imaging.ch](http://www.mesa-imaging.ch)

The Mesa Imaging SwissRanger 4000 (SR4000) is probably the most well-known ToF depth camera. It has a range of 5-8 meters, 176 x 144 pixel resolution over 43.6° x 34.6° field of view. It operates at up to 54 fps, and costs about $9,000. I've seen these used in a number of academic laboratories.

The PMD Technologies CamCube 2.0 is a lesser-known, but equally impressive ToF depth camera. It has a range of 7 meters, 204 x 204 pixel resolution with 40.0° x 40.0° field of view. It operates at 25 fps, and last time I checked, it costs around $12,000.
Computer Vision

Spring 2006 15-385,-685

Instructor: S. Narasimhan

Wean 5403
T-R 3:00pm – 4:20pm

Lecture #17

Structured Light + Range Imaging

Guido Gerig
CS 6320, Spring 2012
(thanks: slides Prof. S. Narasimhan, CMU, Marc Pollefeys, UNC)
http://www.cs.cmu.edu/afs/cs/academic/class/15385-s06/lectures/ppts/lec-17.ppt

lbv@dongseoo.ac.kr
Correspondence is hard!
Structured Light Triangulation

Correspondence becomes easier!
Example: Laser scanner

- very accurate < 0.01 mm
- more than 10 sec per scan

Cyberware® face and head scanner

Digital Michelangelo Project
http://graphics.stanford.edu/projects/mich/
Portable 3D laser scanner

http://www.youtube.com/watch?v=IqOg77liryg
Leica ScanStation C10

Distance Measuring System
Receiver
Transmitter

Angle Measuring System (Hz)
Emitter
Receiver
Glass Circle
Motorization

Laser Plummet
Emitter

X-Mirror
Camera Aperture
Laser Aperture

Angle Measuring System (V)
Motorization
Glass Circle

Camera
Dual Axis Compensator
Oil Surface
Emitter

Crude Storage Facility
Corpus Christi, Texas

270°
360°
Leica ScanStation C10
# Leica ScanStation C10

## System Performance

<table>
<thead>
<tr>
<th>Accuracy of single measurement</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Position*</td>
<td>6 mm</td>
<td>* At 1 m - 50 m range, one sigma</td>
</tr>
<tr>
<td>Distance*</td>
<td>4 mm</td>
<td></td>
</tr>
<tr>
<td>Angle (horizontal/vertical)</td>
<td>60 μrad / 60 μrad (12” / 12”)</td>
<td></td>
</tr>
</tbody>
</table>

| Modeled surface precision**/noise | 2 mm |
| Target acquisition***            | 2 mm std. deviation |
| Dual-axis compensator            | Selectable on/off, resolution 1”, dynamic range +/- 5’, accuracy 1.5” |

## Laser Scanning System

<table>
<thead>
<tr>
<th>Type</th>
<th>Pulsed; proprietary microchip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color</td>
<td>Green, wavelength = 532 nm visible</td>
</tr>
<tr>
<td>Laser Class</td>
<td>3R (IEC 60825-1)</td>
</tr>
<tr>
<td>Range</td>
<td>300 m @ 90%; 134 m @ 18% albedo (minimum range 0.1 m)</td>
</tr>
<tr>
<td>Scan rate</td>
<td>Up to 50,000 points/sec, maximum instantaneous rate</td>
</tr>
</tbody>
</table>

### Scan resolution

| Spot size                  | From 0 – 50 m: 4.5 mm (FWHM-based); 7 mm (Gaussian-based) |
| Point spacing              | Fully selectable horizontal and vertical; < 1 mm minimum spacing, through full range; single point dwell capacity |

### Field-of-View

| Horizontal                 | 360° (maximum) |
| Vertical                   | 270° (maximum) |
| Aiming/Sighting            | Parallax-free, integrated zoom video |

### Scanning Optics

Vertically rotating mirror on horizontally rotating base; Smart X-Mirror™ automatically spins or oscillates for minimum scan time
LMS LIDAR scanner from SICK

3D laser scanners

3D laser scanners allow almost continuous scanning of their environment—regardless of whether objects are moving or not. As a result, 3D laser scanners are ideally suited to tasks such as collision prevention on automated vehicles or the scanning of objects.

**JEF3xx**
- 2D and 3D full field control
- Simple download of new configurations
- Multiple I/O and Interfacing
- Clamping functions for parameters are available

Show 3D laser scanner JEF3xx in catalog

**JEF5xx**
- 2D and 3D laser measurement sensor
- Multiple I/O and Interfacing
- Large working range without any additional focal adjustments
- Output of reflectivity data

Show 3D laser scanner JEF5xx in catalog

Detection and Ranging Solutions Catalog

Download Detection and Ranging Solutions Catalog
Structured Light 3D Surface Imaging

lbg@dongseo.ac.kr
Continuum of Triangulation Methods

- Single-stripe: Slow, robust
- Multi-stripe Multi-frame
- Single-frame: Fast, fragile
Faster Acquisition?

- Project multiple stripes simultaneously
- Correspondence problem: which stripe is which?

- Common types of patterns:
  - Binary coded light striping
  - Gray/color coded light striping
Structured Light Projection Classification

**Structured Light 3D Surface Imaging Techniques**

- **Sequential Projections (Multi-Shots)**
  - Binary Code
  - Gray Code
  - Phase Shift
  - Hybrid: Gray code + Phase Shift

- **Continuous Varying Pattern (Single Shot)**
  - Rainbow 3D Camera
  - Continuously Varying Color Code

- **Stripe Indexing (Single Shot)**
  - Color Coded Stripes
  - Segmented Stripes
  - Gray Scale Coded Stripes
  - De Bruijn Sequence

- **Grid Indexing (Single Shot)**
  - Pseudo Random Binary-Dots
  - Mini-Patterns as Codewords
  - Color Coded Grid
  - 2D Color Coded Dot Array

- **Hybrid Methods**
Binary Coding

Example: 7 binary patterns proposed by Posdamer & Altschuler

Codeword of this pixel: 1010010 → identifies the corresponding pattern stripe

Example: A Tale of Two Pixels
Example: A Tale of Two Pixels
Example: A Tale of Two Pixels
Example: A Tale of Two Pixels
Example: A Tale of Two Pixels
Example: A Tale of Two Pixels
Example: A Tale of Two Pixels
Example: A Tale of Two Pixels
Example: A Tale of Two Pixels
Example: A Tale of Two Pixels
## Decoding the Example

### Yellow Pixel Decoding
- **Gray Code**: 110100111
- **Binary**: 100111010
- **Projector column**: 314
- **Camera column**: 335
- **Disparity**: $335 - 314 = 21$
- ** Unscaled results:**
  - $x = \text{Camera column} = 335$
  - $y = \text{Camera row}$
  - $z = (\text{Disparity})^{-1} = 0.0476$

### Red Pixel Decoding
- **Gray Code**: 010000111
- **Binary**: 011111010
- **Projector column**: 250
- **Camera column**: 392
- **Disparity**: $392 - 250 = 142$
- ** Unscaled results:**
  - $x = \text{Camera column} = 392$
  - $y = \text{Camera row}$
  - $z = (\text{Disparity})^{-1} = 0.007042$
Results from Meshlab
Overview

Geometric calibration

- Camera intrinsics: $K_{\text{cam}}$
- Projector intrinsics: $K_{\text{proj}}$
- Projector-Camera extrinsics: Rotation and translation: $R, T$

The simplest structured-light system consists of a camera and a data projector.
Application: 3D scanning

1. Data acquisition
2. Decode
3. Triangulation
4. Mesh

Correspondences + Calibration = Pointcloud

Pointclouds from several viewpoints can be merged into a single one and used to build a 3D model
Camera calibration: well-known problem

Pinhole model + radial distortion

\[
K = \begin{bmatrix}
    f_x & s & c_x \\
    0 & f_y & c_y \\
    0 & 0 & 1
\end{bmatrix}
\]

\[x = K \cdot L(X; k_1, k_2, k_3, k_4)\]

X: 3D point
k_1, ..., k_4: distortion coefficients
K: camera intrinsics
x: projection of X into the image plane

If we have enough X↔x point correspondences we can solve for all the unknowns

Images from different viewpoints

\[x_1 = K \cdot L(R_1X + T_1; k_1, k_2, k_3, k_4)\]
\[x_2 = K \cdot L(R_2X + T_2; k_1, k_2, k_3, k_4)\]
\[x_3 = K \cdot L(R_3X + T_3; k_1, k_2, k_3, k_4)\]
...
Fringe Projection 3D Scanning

lb@dongseo.ac.kr
Agenda

- Fringe Projection Techniques
- Three step phase shifting algorithm
Three-step Phase-shifting algorithm

\[
\begin{align*}
I_1 &= I' + I'' \cos(\phi - 2\pi/3) \\
I_2 &= I' + I'' \cos(\phi) \\
I_3 &= I' + I'' \cos(\phi + 2\pi/3)
\end{align*}
\]
Wrapped Phase Image

\[ \phi = \tan^{-1} \left[ \frac{\sqrt{3}(I_1 - I_3)}{2I_2 - I_1 - I_3} \right] \]

Wrapped Phase Image
Unwrapped Phase Image

\[ \Phi = 2\pi \times m + \phi, \ m \text{ a series of constants} \]
Calibration Plane

\[ z = z_0 + c_0(\Phi - \Phi^r) \]
3D Result
Phase shifting patterns

For details on three-step phase-shifting patterns see Malacara, Optical Shop Testing, in the references.
Sinusoid patterns

For details on generating dithered sinusoid patterns, see Wang and Zhang, Applied Optics 51(27) 2012
System Setup
Large angle setting
Small angle setting

Weak Fringe Bending

Noisy Results

Small Shadows

Small Holes
Calibration I
Calibration II
http://www.jove.com/video/50421/high-resolution-high-speed-three-dimensional-video-imaging-with
Thanks you!!
lbg@dongseo.ac.kr
http://kowon.dongseo.ac.kr/~lbgs/
RECENT PROGRESS IN CODED STRUCTURED LIGHT AS A TECHNIQUE TO SOLVE THE CORRESPONDENCE PROBLEM: A SURVEY

J. BATLLE,*† E. MOUADDIB‡ and J. SALVI†

† Computer Vision and Robotics Group Universitat de Girona, Avda. Lluís Santalo, s/n. 17071 Girona, Spain
‡ Laboratoire des Systèmes Automatiques, Université de Picardie Jules Verne, 7, Rue du Moulin Neuf, 80000 Amiens, France

(Received 5 December 1996)

Abstract—We present a survey of the most significant techniques, used in the last few years, concerning the coded structured light methods employed to get 3D information. In fact, depth perception is one of the most important subjects in computer vision. Stereovision is an attractive and widely used method, but, it is rather limited to make 3D surface maps, due to the correspondence problem. The correspondence problem can be improved using a method based on structured light concept, projecting a given pattern on the measuring surfaces. However, some relations between the projected pattern and the reflected one must be solved. This relationship can be directly found codifying the projected light, so that, each imaged region of the projected pattern carries the needed information to solve the correspondence problem. © 1998 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Pattern projection Correspondence problem Active stereo Depth perception Range data
Computer vision.

General Stereoscopic System

lbg@dongseo.ac.kr
General Stereoscopic System

\[
\begin{bmatrix}
  x_2 \\
  y_2 \\
  z_2
\end{bmatrix} = R \begin{bmatrix}
  x_1 \\
  y_1 \\
  z_1
\end{bmatrix} + T.
\]

\[
P_1 = F_1 + \alpha(P_0 - F_1)
\]

\[
\begin{bmatrix}
  x_{p1} \\
  y_{p1} \\
  0
\end{bmatrix} = \begin{bmatrix}
  0 \\
  0 \\
  f_1
\end{bmatrix} + \alpha \begin{bmatrix}
  x_{p0} \\
  y_{p0} \\
  z_{p0} - f_1
\end{bmatrix}
\]

\[
\begin{align*}
x_{p0} &= \frac{y_{p0}}{f_1} = \frac{f_1 - z_{p0}}{f_1} \\
x_{p1} &= \frac{y_{p1}}{f_1} \\
x'_{p0} &= \frac{y'_{p0}}{f_2} = \frac{f_2 - z'_{p0}}{f_2}
\end{align*}
\]

Fig. 1. A general stereoscopic system made by the relationship between two optical sensors.
Simple Stereoscopic System

\[ P_1 = F_1 + \alpha(P_0 - F_1) \]

\[
\begin{bmatrix}
    x_{p1} \\
    y_{p1} \\
    0
\end{bmatrix} = 
\begin{bmatrix}
    0 \\
    0 \\
    f_1
\end{bmatrix} + \alpha 
\begin{bmatrix}
    x_{p0} \\
    y_{p0} \\
    z_{p0} - f_1
\end{bmatrix}.
\]

\[
O_2 = \begin{bmatrix}
    x_2 \\
    y_2 \\
    z_2
\end{bmatrix}, \quad F_2 = \begin{bmatrix}
    x_2 \\
    y_2 \\
    z_2 + f_2
\end{bmatrix}.
\]

\[
P_2 = \begin{bmatrix}
    x_2 \\
    y_2 \\
    z_2
\end{bmatrix} + \begin{bmatrix}
    x_{p2} \\
    y_{p2} \\
    0
\end{bmatrix} = \begin{bmatrix}
    x_2 + x_{p2} \\
    y_2 + y_{p2} \\
    z_2
\end{bmatrix}.
\]

Fig. 2. Measurement system used in structured light
Simple Stereoscopic System

\[ P_2 = F_2 + \beta(P_0 - F_2) \]
\[ \begin{bmatrix} x_2 + x_{p2} \\ y_2 + y_{p2} \\ z_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ y_2 \\ z_2 + f_2 \end{bmatrix} + \beta \begin{bmatrix} x_{p0} - x_2 \\ y_{p0} - y_2 \\ z_{p0} - z_2 - f_2 \end{bmatrix} \]

\[ x_{p0} = \frac{f_1 - z_{p0}}{f_1} x_{p1}, \]
\[ y_{p0} = \frac{f_1 - z_{p0}}{f_1} y_{p1}, \]
\[ x_{p0} = x_2 + \frac{f_2 + z_2 - z_{p0}}{f_2} x_{p2}, \]
\[ y_{p0} = y_2 + \frac{f_2 + z_2 - z_{p0}}{f_2} y_{p2}. \]

Fig. 2. Measurement system used in structured light
Simple Stereoscopic System

\[
\frac{f_1 - z_{p0}}{f_1} x_{p1} = x_2 + \frac{f_2 + z_2 - z_{p0}}{f_2} x_{p2},
\]

\[
\frac{f_1 - z_{p0}}{f_1} y_{p1} = y_2 + \frac{f_2 + z_2 - z_{p0}}{f_2} y_{p2}.
\]

\[
z_{p0} = \frac{f_1 f_2}{f_1 x_{p2} - f_2 x_{p1}} \left( x_2 + x_{p2} - x_{p1} + \frac{z_2 x_{p2}}{f_2} \right).
\]

\[
z_{p0} = \frac{f_1 f_2}{f_1 y_{p2} - f_2 y_{p1}} \left( y_2 + y_{p2} - y_{p1} + \frac{z_2 y_{p2}}{f_2} \right).
\]

Fig. 2. Measurement system used in structured light
Camera Calibration
Pinhole Camera Model

\[(X, Y, Z)^T \mapsto (fX / Z, fY / Z)^T\]

\[
\begin{bmatrix}
X \\
Y \\
Z \\
1
\end{bmatrix}
\mapsto
\begin{bmatrix}
fX \\
fY \\
Z
\end{bmatrix}
= \begin{bmatrix}
f & 0 \\
f & 0 \\
1 & 0
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Z \\
1
\end{bmatrix}
\]

\[
\begin{bmatrix}
fX \\
fY \\
Z
\end{bmatrix}
= \begin{bmatrix}
f & 1 & 0 \\
1 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
X \\
Y \\
Z \\
1
\end{bmatrix}
\]
Internal Camera Parameters

\[
\begin{bmatrix}
u' \\
v' \\
w'
\end{bmatrix} = \begin{bmatrix}
\alpha_x & s & x_0 & 0 \\
0 & \alpha_y & y_0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix} \begin{bmatrix}
x_s \\
y_s \\
z_s \\
1
\end{bmatrix}
\]

with

\[\alpha_x = f \ k_x\]
\[\alpha_y = -f \ k_y\]

\[x_{\text{pix}} = u' / w'\]
\[y_{\text{pix}} = v' / w'\]

\[
\begin{bmatrix}
\alpha_x & s & x_0 & 0 \\
0 & \alpha_y & y_0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix} \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0
\end{bmatrix} = K \begin{bmatrix} I_3 & 0_3 \end{bmatrix}
\]

• \(\alpha_x\) and \(\alpha_y\) “focal lengths” in pixels

• \(x_0\) and \(y_0\) coordinates of image center in pixels

• Added parameter \(S\) is skew parameter

• \(K\) is called calibration matrix. **Five degrees of freedom.**

  • \(K\) is a 3x3 upper triangular matrix
Camera rotation and translation

\[ \tilde{X}_{\text{cam}} = R(\tilde{X} - \tilde{C}) \]

\[ X_{\text{cam}} = \begin{bmatrix} R & -R\tilde{C} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} R & -RC \\ 0 & 1 \end{bmatrix} X \]

\[ x = K[I | 0]X_{\text{cam}} \]

\[ x = KR[I | -\tilde{C}]X \]

\[ x = PX \]

\[ P = K[R | t] \quad t = -R\tilde{C} \]
Camera Distortion

Pincushion Distortion

No Distortion

Barrel Distortion

Element CCD
Correcting Radial Distortion of Cameras

\[ x_u = c_x + (x_d - c_x) f_2(r_d^2) \]
\[ = c_x + (x_d - c_x)(1 + k_1 r_d^2 + k_2 r_d^4 + k_3 r_d^6) \]
\[ y_u = c_y + (y_d - c_y) f_2(r_d^2) \]
\[ = c_y + (y_d - c_y)(1 + k_1 r_d^2 + k_2 r_d^4 + k_3 r_d^6) \]
\[ r_d^2 = (x_d - c_x)^2 + (y_d - c_y)^2 \]

\[ x_p = \frac{m_0 x_u + m_1 y_u + m_2}{m_6 x_u + m_7 y_u + 1} \]
\[ y_p = \frac{m_3 x_u + m_4 y_u + m_5}{m_6 x_u + m_7 y_u + 1} \]
Structured-light systems are simple and effective tools to acquire 3D models. Built with off-the-shelf components, a data projector and a camera, they are easy to deploy and compare in precision with expensive laser scanners. But such a high precision is only possible if camera and projector are both accurately calibrated. Robust calibration methods are well established for cameras but, while cameras and projectors can both be described with the same mathematical model, it is not clear how to adapt these methods to projectors. In consequence, many of the proposed projector calibration techniques make use of a simplified model, neglecting lens distortion, resulting in loss of precision. In this paper, we present a novel method to estimate the image coordinates of 3D points in the projector image plane. The method relies on an uncalibrated camera and makes use of local homographies to reach sub-pixel precision. As a result, any camera model can be used to describe the projector, including the extended pinhole model with radial and tangential distortion coefficients, or even those with more complex lens distortion models.
Projector calibration: ?

Use the pinhole model to describe the projector:
- Projectors work as an inverse camera

\[
K_{proj} = \begin{bmatrix}
fx & s & cx \\
0 & fy & cy \\
0 & 0 & 1
\end{bmatrix}
\]

\[x = K_{proj} \cdot L(X; k_1, k_2, k_3, k_4)\]

If we model the projector the same as our camera, we would like to calibrate the projector just as we do for the camera:
- We need correspondences between 3D world points and projector image plane points: \(X \leftrightarrow x\)
- The projector cannot capture images

Challenge: How do we find point correspondences?
Related works

There have been proposed several projector calibration methods*, they can be divided in three groups:

1. **Rely on camera calibration**
   - First the camera is calibrated, then, camera calibration is used to find the 3D world coordinates of the projected pattern
   - *Inaccuracies in the camera calibration* translates into errors in the projector calibration

2. **Find projector correspondences using homographies between planes**
   - Cannot model projector lens distortion because of the linearity of the transformation

3. **Too difficult to perform**
   - Required special equipment or calibration artifacts
   - Required color calibration
   - ...

(*) See the paper for references

Existing methods were *not accurate enough* or *not practical*.
Proposed method: overview

Features:

Simple to perform:
- no special equipment required
- reuse existing components

Accurate:
- there are no constrains for the mathematical model used to describe the projector
- we use the full pinhole model with radial distortion (as for cameras)

Robust:
- can handle small decoding errors
Proposed method: acquisition

Traditional camera calibration
- requires a planar checkerboard (easy to make with a printer)
- capture pictures of the checkerboard from several viewpoints

Structured-light system calibration
- use a planar checkerboard
- capture structured-light sequences of the checkerboard from several viewpoints
Proposed method: decoding

Decoding depends on the projected pattern
- The method does not rely on any specific pattern

Our implementation uses complementary gray code patterns
- Robust to light conditions and different object colors (notice that we used the standard B&W checkerboard)
- Does not required photometric calibration (as phase-shifting does)
- We prioritize calibration accuracy over acquisition speed
- Reasonable fast to project and capture: if the system is synchronized at 30fps, the 42 images used for each pose are acquired in 1.4 seconds

Our implementation decodes the pattern using “robust pixel classification”(*)
- High-frequency patterns are used to separate direct and global light components for each pixel
- Once direct and global components are known each pixel is classified as ON, OFF, or UNCERTAIN using a simple set of rules

(*) Y. Xu and D. G. Aliaga, “Robust pixel classification for 3D modeling with structured light”
Conclusions

- It works 😊
- No special setup or materials required
- Very similar to standard stereo camera calibration
- Reuse existing software components
  - Camera calibration software
  - Structured-light projection, capture, and decoding software
- Local homographies effectively handle projector lens distortion
- Adding projector distortion model improves calibration accuracy
- Well-calibrated structured-light systems have a precision comparable to some laser scanners
Thanks you!!

lbgl@dongseo.ac.kr
http://kowon.dongseo.ac.kr/~lbg/
Proposed method: projector calibration

Once the structured-light pattern is decoded we have a mapping between projector and camera pixels:

1) Each camera pixel is associated to a projector row and column, or set to **UNCERTAIN**

   ![Image](image1.png)

   For each \((x, y)\): \(\text{Map}(x, y) = (\text{row}, \text{col})\) or **UNCERTAIN**

2) The map is **not bijective**: many camera pixels corresponds to the same projector pixel

3) Checkerboard corners are **not located at integer pixel locations**
Proposed method: projector calibration

Solution: local homographies

1. Surface is locally planar: actually the complete checkerboard is a plane
2. Radial distortion is negligible in a small neighborhood
3. Radial distortion is significant in the complete image:
   - a single global homography is not enough

For each checkerboard corner solve:

$$\hat{H} = \arg\min_{H} \sum_{\forall p} \|q - Hp\|^2, \quad \bar{q} = \hat{H} \cdot p$$

$$\hat{H} \in \mathbb{R}^{3 \times 3}, \quad p = [x, y, 1]^T, \quad q = [col, row, 1]^T$$
Proposed method: projector calibration

Summary:

1. **Decode** the structured-light pattern: camera $\leftrightarrow$ projector map
2. Find checkerboard corner locations in camera image coordinates
3. **Compute** a local homography $H$ for each corner
4. **Translate** each corner from image coordinates $x$ to projector coordinates $x'$ applying the corresponding local homography $H$

\[
x' = H \cdot x
\]

5. Using the correspondences between the projector corner coordinates and 3D world corner locations, $X \leftrightarrow x'$, find projector intrinsic parameters
Camera calibration and system extrinsics

Camera intrinsics

Using the corner locations in image coordinates and their 3D world coordinates, we calibrate the camera as usual

- Note that no extra images are required

System extrinsics

Once projector and camera intrinsics are known we calibrate the extrinsics (R and T) parameters as is done for camera-camera systems

Using the previous correspondences, $x \leftrightarrow x'$, we fix the coordinate system at the camera and we solve for R and T:

$$
\tilde{x}_1 = L^{-1}(K_{\text{cam}}^{-1} \cdot x_1; k_1, k_2, k_3, k_4)
$$

$$
\tilde{x}_2 = L^{-1}(K_{\text{cam}}^{-1} \cdot x_2; k_1, k_2, k_3, k_4)
$$

$$
\tilde{x}_3 = L^{-1}(K_{\text{cam}}^{-1} \cdot x_3; k_1, k_2, k_3, k_4)
$$

$$
x'_1 = K_{\text{proj}} \cdot L(R \cdot \tilde{x}_1 + T; k'_1, k'_2, k'_3, k'_4)
$$

$$
x'_2 = K_{\text{proj}} \cdot L(R \cdot \tilde{x}_2 + T; k'_1, k'_2, k'_3, k'_4)
$$

$$
x'_3 = K_{\text{proj}} \cdot L(R \cdot \tilde{x}_3 + T; k'_1, k'_2, k'_3, k'_4)
$$

...
The proposed calibration method can be implemented fully automatic:
- The user provides a folder with all the images
- Press “calibrate” and the software automatically extracts the checkerboard corners, decode the structured-light pattern, and calibrates the system

Algorithm
1. Detect checkerboard corner locations for each plane orientation
2. Estimate global and direct light components
3. Decode structured-light patterns
4. Compute a local homography for each checkerboard corner
5. Translate corner locations into projector coordinates using local homographies
6. Calibrate camera intrinsics using image corner locations
7. Calibrate projector intrinsics using projector corner locations
8. Fix projector and camera intrinsics and calibrate system extrinsic parameters
9. Optionally, all the parameters, intrinsic and extrinsic, can be optimized together
Results

Comparison with existing software:

**procamcalib**
- Projector-Camera Calibration Toolbox
- http://code.google.com/p/procamcalib/

Reprojection error comparison

<table>
<thead>
<tr>
<th>Method</th>
<th>Camera</th>
<th>Projector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td></td>
<td>0.1447</td>
</tr>
<tr>
<td>With global homography</td>
<td>0.3288</td>
<td>0.2176</td>
</tr>
<tr>
<td>Procamcalib</td>
<td></td>
<td>0.8671</td>
</tr>
</tbody>
</table>

- Only projector calibration is compared
- Same camera intrinsics is used for all methods
- Global homography means that a single homography is used to translate all corners
Results

Example of projector lens distortion

<table>
<thead>
<tr>
<th>k1</th>
<th>k2</th>
<th>k3</th>
<th>k4</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.0888</td>
<td>0.3365</td>
<td>-0.0126</td>
<td>-0.0023</td>
</tr>
</tbody>
</table>

Non trivial distortion!
Results

Error distribution on a scanned 3D plane model:

Laser scanner comparison

3D Model

Hausdorff distance

Model with small details reconstructed using SSD
Direct/Global light components

\[ L^+ = L_d + \alpha L_g + b(1-\alpha)L_g \]
\[ L^- = bL_d + (1-\alpha)L_g + \alpha bL_g \]

\[ L_d = \frac{L^+ - L^-}{1-b} \]
\[ L_g = 2 \frac{L^- - bL^+}{1-b^2} \]
\[ \hat{L}^+ = \max_{0<i<K} I_i \]
\[ \hat{L}^- = \min_{0<i<K} I_i \]

Robust pixel classification

\[
\begin{align*}
L_d < m & \rightarrow \text{UNCERTAIN} \\
L_d > L_g \land p > \overline{p} & \rightarrow \text{ON} \\
L_d > L_g \land p < \overline{p} & \rightarrow \text{OFF} \\
p < L_d \land \overline{p} > L_g & \rightarrow \text{OFF} \\
p > L_g \land \overline{p} < L_d & \rightarrow \text{ON} \\
\text{otherwise} & \rightarrow \text{UNCERTAIN}
\end{align*}
\]
Triangulation

\[ \lambda_1 u_1 = R_1 X + T_1 \]
\[ \lambda_2 u_2 = R_2 X + T_2 \]

\[ \hat{u}_1 \lambda_1 u_1 = \hat{u}_1 R_1 X + \hat{u}_1 T_1 = 0 \]
\[ \hat{u}_2 \lambda_2 u_2 = \hat{u}_2 R_2 X + \hat{u}_2 T_2 = 0 \]

In homogeneous coordinates:

\[
\begin{bmatrix}
\hat{u}_1 R_1 & \hat{u}_1 T_1 \\
\hat{u}_2 R_2 & \hat{u}_2 T_2
\end{bmatrix} X = 0
\]
Calibration – Perspective Transformation

\[
\begin{bmatrix}
w x_{p1} \\
w y_{p1} \\
w
\end{bmatrix} =
\begin{bmatrix}
A_{11} & A_{12} & A_{13} & A_{14} \\
A_{21} & A_{22} & A_{23} & A_{24} \\
A_{31} & A_{32} & A_{33} & A_{34}
\end{bmatrix}
\begin{bmatrix}
x_{p0} \\
y_{p0} \\
z_{p0} \\
1
\end{bmatrix}.
\]

\[
A_{11} x_{p0} - A_{31} x_{p1} x_{p0} + A_{12} y_{p0} - A_{32} x_{p1} y_{p0} + A_{13} z_{p0} - A_{33} x_{p1} z_{p0} + A_{14} - A_{34} x_{p1} = 0,
\]

\[
A_{21} x_{p0} - A_{31} y_{p1} x_{p0} + A_{22} y_{p0} - A_{32} y_{p1} y_{p0} + A_{23} z_{p0} - A_{33} y_{p1} z_{p0} + A_{24} - A_{34} y_{p1} = 0.
\]
\[ QA = B, \]
\[ A = [A_{11}A_{12}A_{13}A_{14}A_{21}A_{22}A_{23}A_{24}A_{31}A_{32}A_{33}]^t \]

\[ q_x = \begin{bmatrix} x_{p0} \\
 y_{p0} \\
 z_{p0} \\
 1 \\
 0 \\
 0 \\
 0 \\
 0 \\
 -x_{p1}x_{p0} \\
 -x_{p1}y_{p0} \\
 -x_{p1}z_{p0} \end{bmatrix}^t, \quad q_y = \begin{bmatrix} 0 \\
 0 \\
 0 \\
 0 \\
 x_{p0} \\
 y_{p0} \\
 z_{p0} \\
 1 \end{bmatrix}^t, \quad b_x = [x_{p1}] \\
 b_y = [y_{p1}]. \]

\[ A = \begin{bmatrix} q_{x1} \\
 q_{y1} \\
 q_{x2} \\
 q_{y2} \\
 \vdots \\
 q_{x6} \\
 q_{y6} \end{bmatrix}, \quad b = \begin{bmatrix} b_{x1} \\
 b_{y1} \\
 b_{x2} \\
 b_{y2} \\
 \vdots \\
 b_{x6} \\
 b_{y6} \end{bmatrix}. \]
\[
\begin{bmatrix}
  w_1 x_{p1} \\
  w_1 y_{p1} \\
  w_1
\end{bmatrix} =
\begin{bmatrix}
  A_{111} & A_{112} & A_{113} & A_{114} \\
  A_{121} & A_{122} & A_{123} & A_{124} \\
  A_{131} & A_{132} & A_{133} & A_{134}
\end{bmatrix}
\begin{bmatrix}
  x_{p0} \\
  y_{p0} \\
  z_{p0} \\
  1
\end{bmatrix}
\]

\[
(A_{111} - A_{131} x_{p1}) x_{p0} + (A_{112} - A_{132} x_{p1}) y_{p0} + (A_{113} - A_{133} x_{p1}) z_{p0} = A_{134} x_{p1} - A_{114};
\]

\[
(A_{121} - A_{131} y_{p1}) x_{p0} + (A_{122} - A_{132} y_{p1}) y_{p0} + (A_{123} - A_{133} y_{p1}) z_{p0} = A_{134} y_{p1} - A_{124};
\]

\[
\begin{bmatrix}
  w_2 x_{p2} \\
  w_2 y_{p2} \\
  w_2
\end{bmatrix} =
\begin{bmatrix}
  A_{211} & A_{212} & A_{213} & A_{214} \\
  A_{221} & A_{222} & A_{223} & A_{224} \\
  A_{231} & A_{232} & A_{233} & A_{234}
\end{bmatrix}
\begin{bmatrix}
  x_{p0} \\
  y_{p0} \\
  z_{p0} \\
  1
\end{bmatrix}
\]

\[
(A_{211} - A_{231} x_{p2}) x_{p0} + (A_{212} - A_{232} x_{p2}) y_{p0} + (A_{213} - A_{233} x_{p2}) z_{p0} = A_{234} x_{p2} - A_{214};
\]

\[
(A_{221} - A_{231} y_{p2}) x_{p0} + (A_{222} - A_{232} y_{p2}) y_{p0} + (A_{223} - A_{233} y_{p2}) z_{p0} = A_{234} y_{p2} - A_{224}.
\]
\[ PV = F, \quad V = (P^t P)^{-1} P^t F. \]

\[
P = \begin{bmatrix}
A_{111} - A_{131}x_{p1} & A_{112} - A_{132}x_{p1} & A_{113} - A_{133}x_{p1} \\
A_{121} - A_{131}y_{p1} & A_{122} - A_{132}y_{p1} & A_{123} - A_{133}y_{p1} \\
A_{211} - A_{231}x_{p2} & A_{212} - A_{232}x_{p2} & A_{213} - A_{233}x_{p2} \\
A_{221} - A_{231}y_{p2} & A_{222} - A_{232}y_{p2} & A_{223} - A_{233}y_{p2}
\end{bmatrix}, \quad V = \begin{bmatrix} x_{p0} \\ y_{p0} \\ z_{p0} \end{bmatrix}, \quad F = \begin{bmatrix} A_{134}x_{p1} - A_{114} \\
A_{134}y_{p1} - A_{124} \\
A_{234}x_{p2} - A_{214} \\
A_{234}y_{p2} - A_{224} \end{bmatrix}.
3D Acquisition from Shadows

The idea

Desks Lamp

Stick or pencil

Camera

Desk

Time t
3D Acquisition from Shadows

The geometry

$P = (O, p) \cap \Pi$
3D Acquisition from Shadows

The geometry

\[ \Lambda = (O, \lambda) \cap \Pi_d \]

\[ \Pi = (S, \Lambda) \]
3D Acquisition from Shadows

The geometry

\[ \Lambda_1 = (O, \lambda_1) \cap \Pi_d \]
\[ \Lambda_2 = (O, \lambda_2) \cap \Pi_v \]
\[ \Pi = (\Lambda_1, \Lambda_2) \]
3D Acquisition from Shadows

Angel experiment

Accuracy: 0.1mm over 10cm ~ 0.1% error
3D Acquisition from Shadows

Scanning with the sun

Accuracy: 1cm over 2m

~ 0.5% error
3D Model Acquisition Pipeline

3D Scanner
3D Model Acquisition Pipeline

View Planning

3D Scanner
3D Model Acquisition Pipeline

- View Planning
- 3D Scanner
- Alignment
3D Model Acquisition Pipeline

- View Planning
- 3D Scanner
- Alignment
- Merging
3D Model Acquisition Pipeline

- View Planning
- Alignment
- Merging

3D Scanner

Done?
3D Model Acquisition Pipeline

3D Scanner

View Planning

Done?

Alignment

Merging

Display
Stanford Multi-Camera Array

- 640 × 480 pixels × 30 fps × 128 cameras
- synchronized timing
- continuous streaming
- flexible arrangement

http://graphics.stanford.edu/projects/array/
Listed below are the light fields in our archive. For each light field, there is a link to the imagery - sometimes in several forms, and sometimes accompanied by calibration information. Following this is a link that allows you to view the light field in your browser using our Flash-based light field viewer.

**Notes about the light field viewer**

In most cases you shouldn't need to download any software to use our viewer; just click on the indicated links below. Be warned though, this involves loading the entire light field (usually at slightly reduced spatial resolution) into memory. For the largest light fields, this involves downloading about 30MB of data, and will cause your browser to use up to a gigabyte of RAM. Firefox 2 when viewing these light fields, as does Internet Explorer 7. Safari and Opera should also work, with the appropriate flash player plugin.

You're free to take the viewer and use it for your own light fields. The source code and instructions for the using or modifying viewer are available.

**Light Fields from the Lego Gantry**

The light fields in this section were acquired by Andrew Adams.

**Chess**

289 views on a 17x17 grid, image resolution 1400x800  
[Original camera images](#), [Rectified and cropped images](#)  
A chess board with pieces. Chess boards are great for demonstrating refocusing.  
[View light field online](#)

**Lego Bulldozer**

290 views on a 17x17 grid, image resolution 1536x1152  
[Original camera images](#), [Rectified and cropped images](#)  
A Lego Technic bulldozer. Very complex geometry.  
[View light field online](#)

**Lego Truck**

289 views on a 17x17 grid, image resolution 1280x960  
[Original camera images](#), [Rectified and cropped images](#)  
A Lego Technic truck. Very complex geometry.  
[View light field online](#)

**Eucalyptus Flowers**

289 views on a 17x17 grid, image resolution 1280x1536  
[Original camera images](#), [Rectified and cropped images](#)  
Some traditional Australian flowers. Lots of fine geometry.  
[View light field online](#)
Stanford Light Field Archive
Lytro Camera

The Lytro camera. Create pictures that are worth exploring.

Lytro

Free two-day shipping

Buy now

Light Field Engine 1.0

The Light Field Engine replaces the sensor in a camera and processes the light rays data captured by the sensor.

The Light Field Engine travels with every Lytro picture as it is stored, allowing you to adjust pictures right on the camera, on your desktop and online.

Light Field Sensor

From a single lens, Lytro captures a complete map of light in every direction.

Light Field Sensor captures 11 million light rays.

Red Hot

Red Hot

Electric Blue

Red Hot

100 Pictures
$499.00

Electric Blue

100 Pictures
$399.00

CNET First Look

Rex Ng, chief executive of Lytro, a start-up company in Silicon Valley.
Adaptive Optics Microlens Array

125μ square-sided microlenses

4000 × 4000 pixels ÷ 292 × 292 lenses = 14 × 14 pixels per lens
Adobe LightField Camera

Adobe LightField Camera Prototypes

Adobe Systems, the leading specialist of image manipulation, multimedia and creativity software products, has long been investigating the possibilities of LightField photography and computational imaging. Known prototypes date back to 2004, and have evolved from compound lenses with only a few sub-images to microlens arrays.

Here’s an overview of Adobe’s (publicly shown) LightField camera prototypes.

1. Generation Prototype: Magic Lens

Adobe's first LightField camera prototype was tested and developed in 2004-2006, and publicly demoed in 2006/2007. It consisted of a 100 megapixel camera with a special compound lens (dubbed “Magic Lens”) made of 19 sub-lenses in a hexagonal array. Each of these is facing an individually configured prism set at a unique angle, resulting in 19 different focal points.

This setup created 19 subpictures of the entire scene, with a resolution of 5.2 megapixels per subpicture.
Thanks you!!

lbg@dongseo.ac.kr
http://kowon.dongseo.ac.kr/~lbg/