Introducción a la Fotografia 3D

UBA/FCEN Marzo 27 – Abril 12 2013

Clase 6 : Martes Abril 9

Gabriel Taubin

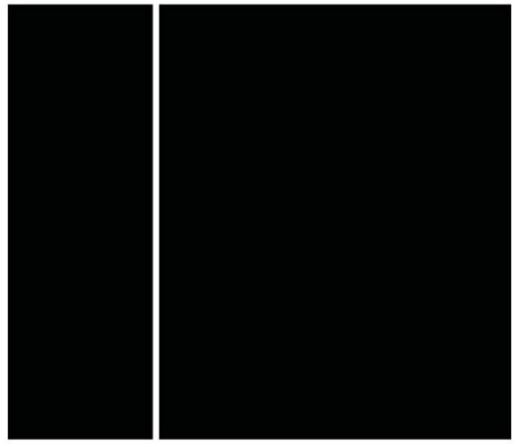
Brown University

BROWN

Course Schedule

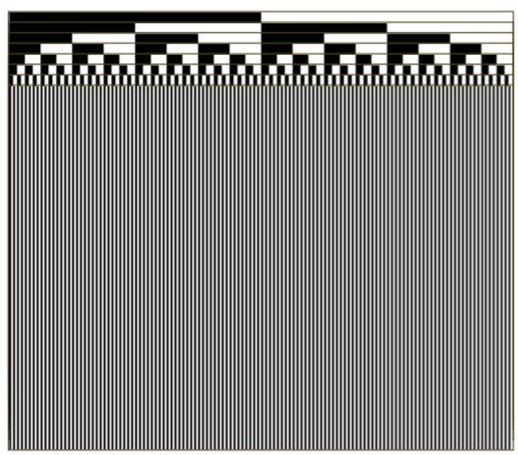
- ***Structured Lighting***
- Projector Calibration and Structured Light Reconstruction

Structured Lighting: Swept-Planes Revisited



- Swept-plane scanning recovers 3D depth using ray-plane intersection
- Use a data projector to replace manually-swept laser/shadow planes
- How to assign correspondence from projector planes to camera pixels?
- Solution: Project a spatially- and temporally-encoded image sequence
- **What is the optimal image sequence to project?**

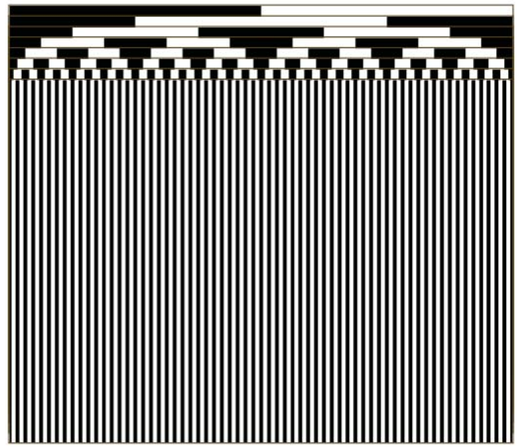
Structured Lighting: Binary Codes



Binary Image Sequence [Posdamer and Altschuler 1982]

- Each image is a bit-plane of the binary code for projector row/column
- Minimum of 10 images to encode 1024 columns or 768 rows
- In practice, 20 images are used to encode 1024 columns or 768 rows
- Projector and camera(s) must be synchronized

Structured Lighting: Gray Codes

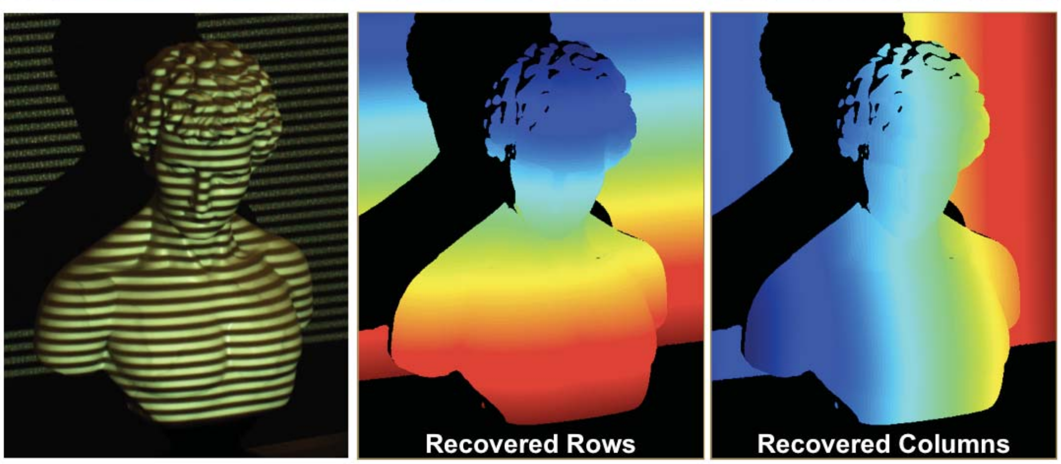


Gray Code Image Sequence [Inokuchi 1984]

- Each image is a bit-plane of the Gray code for each projector row/column
- Requires same number of images as a binary image sequence, but has better performance in practice

```
Bin2Gray(B,G)
G ← B
for i ← n-1 downto 0
    G[i] ← B[i+1] xor B[i]
```

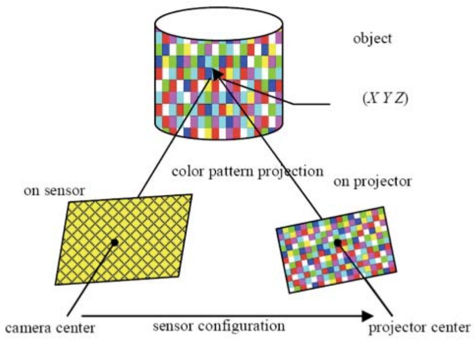
Gray Codes: Decoding Performance



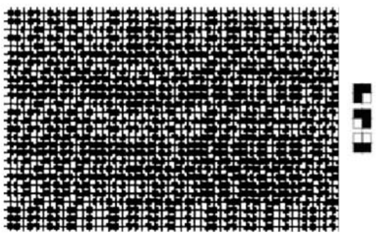
3D Reconstruction using Structured Light [Inokuchi 1984]

- Implemented using a total of 42 images
  (2 to measure dynamic range, 20 to encode rows, 20 to encode columns)
- Individual bits assigned by detecting if bit-plane (or its inverse) is brighter
- Decoding algorithm: Gray code → binary code → integer row/column index

Additional Structured Lighting Patterns

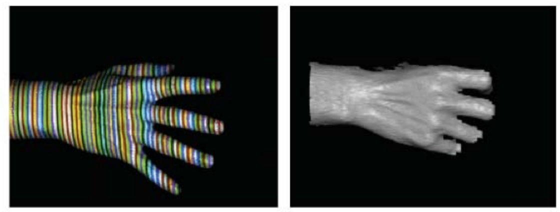


Spatial encoding strategies [Chen et al. 2007]

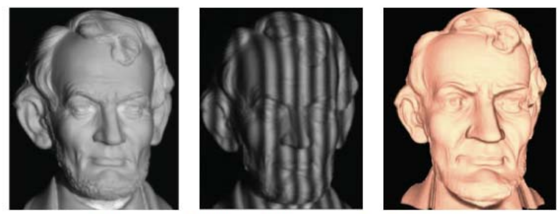


Pseudorandom and M-arrays [Griffin 1992]

"Single-shot" patterns (N-arrays, grids, random, etc.)



De Bruijn sequences [Zhang et al. 2002]



Phase-shifting [Zhang et al. 2004]

J. Salvi, J. Pagès, and J. Batlle. Pattern Codification Strategies in Structured Light Systems. *Pattern Recognition*, 2004

Assembling Your Own Scanner

Parts List [$600-$800, but only $300 without mounting hardware]

- [1] **Camera (Logitech QuickCam Pro 9000) [$70]**
- [1] **Projector (InFocus LP 70+) [$200-$400 from eBay]**
- [1] **Manfrotto 190XB 3 Section Aluminum Tripod [$130]**
- [1] **Manfrotto 486RC2 Compact Ball Head [$75]**
- [1] **Custom Aluminum Rail (with 3/8 inch holes for mounting ball heads) [NA]**
- [1] **Custom Aluminum Tripod Adaptor Plate (for mounting projector to ball head) [NA]**
- [2] **Manfrotto 484 Mini Ball Heads [$45 each]**
- [1] **Sunpak 620-500C Versipod (with adjustable, locking ball head) [$15]**
- [1] **~11x17 inch foamcore board (with an affixed calibration chessboard) [$10]**

Assembling Your Own Scanner

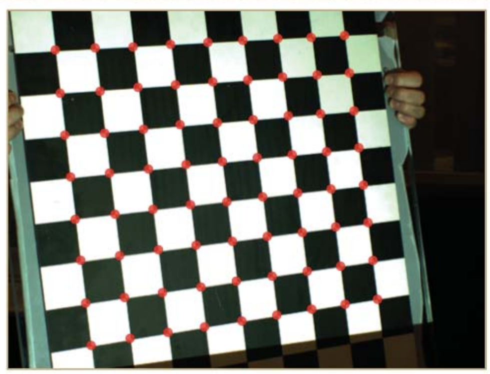
Parts List [$600-$800, but only $300 without mounting hardware]

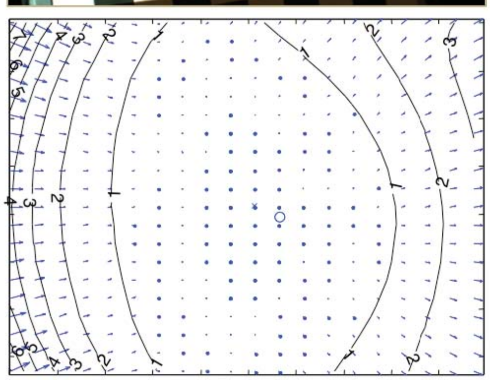
- [1] **Camera (Logitech QuickCam Pro 9000) [$70]**
- [1] **Projector (InFocus LP 70+) [$200-$400 from eBay]**
- [1] **Manfrotto 190XB 3 Section Aluminum Tripod [$130]**
- [1] **Manfrotto 486RC2 Compact Ball Head [$75]**
- [1] **Custom Aluminum Rail (with 3/8 inch holes for mounting ball heads) [NA]**
- [1] **Custom Aluminum Tripod Adaptor Plate (for mounting projector to ball head) [NA]**
- [2] **Manfrotto 484 Mini Ball Heads [$45 each]**
- [1] **Sunpak 620-500C Versipod (with adjustable, locking ball head) [$15]**
- [1] **~11x17 inch foamcore board (with an affixed calibration chessboard) [$10]**

Developing Your Own Software

Key Functions

- Camera and projector calibration





Developing Your Own Software

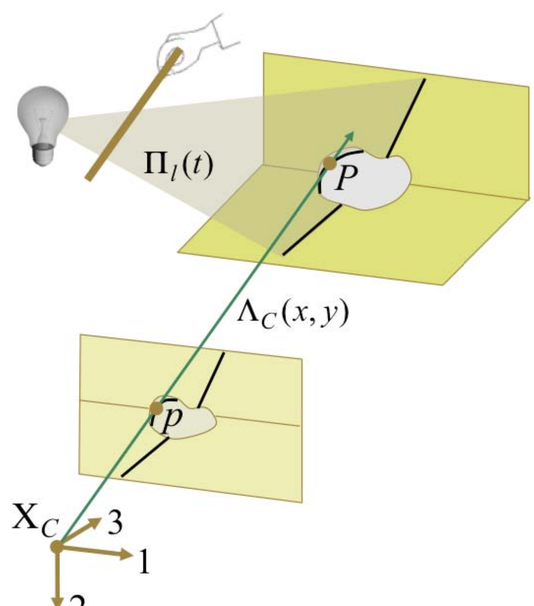
Key Functions

- Camera and projector calibration
- Camera and projector interfaces

Developing Your Own Software

Key Functions

- Camera and projector calibration
- Camera and projector interfaces
- Geometric operations



$$P = \Lambda_C(x,y) \cap \Pi_l(t)$$

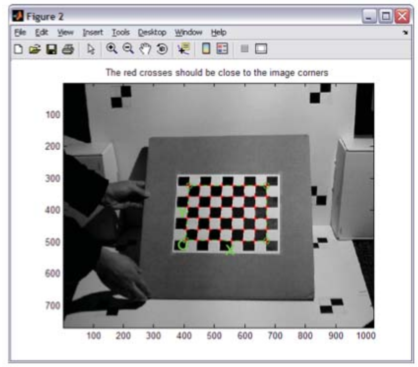
Developing Your Own Software

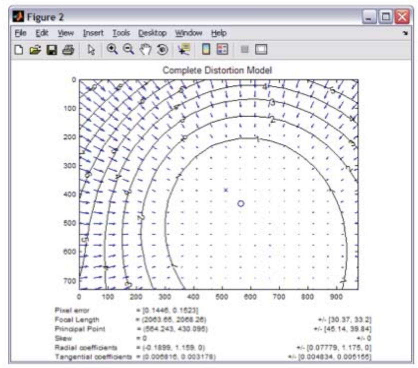
Key Functions

- Camera and projector calibration
- Camera and projector interfaces
- Geometric operations

Matlab

- Camera Calibration Toolbox
  (updated to support projector calibration)





Developing Your Own Software

Key Functions

- Camera and projector calibration
- Camera and projector interfaces
- Geometric operations

Matlab

- Camera Calibration Toolbox
  (updated to support projector calibration)
- Projector-Camera Calibration Toolbox
  (just released, supports full calibration)

http://code.google.com/p/procamcalib/

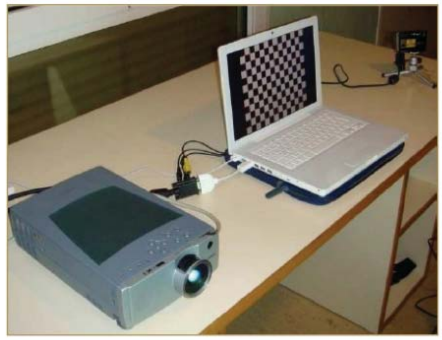
Developing Your Own Software

Key Functions

- Camera and projector calibration
- Camera and projector interfaces
- Geometric operations

Matlab

- Camera Calibration Toolbox (updated to support projector calibration)
- Projector-Camera Calibration Toolbox (just released, supports full calibration)
- Image Acquisition Toolbox, etc.
- Geometric operations (course source)



http://code.google.com/p/procamcalib/

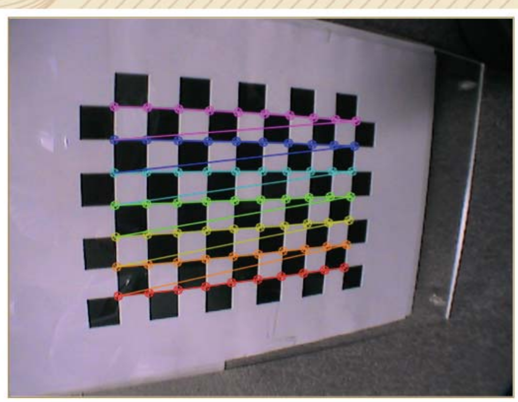
Developing Your Own Software

Key Functions

- Camera and projector calibration
- Camera and projector interfaces
- Geometric operations

Matlab

- Camera Calibration Toolbox (updated to support projector calibration)
- Projector-Camera Calibration Toolbox (just released, supports full calibration)
- Image Acquisition Toolbox, etc.
- Geometric operations (course source)



OpenCV

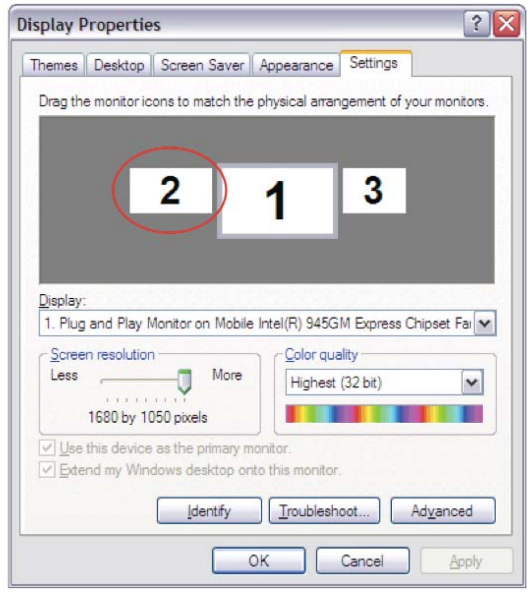
- Supports camera calibration
- Updated to support projector calibration (course source)

http://opencv.willowgarage.com/wiki/

Controlling a Projector for Structured Lighting

Operating System

- A projector is just another display…



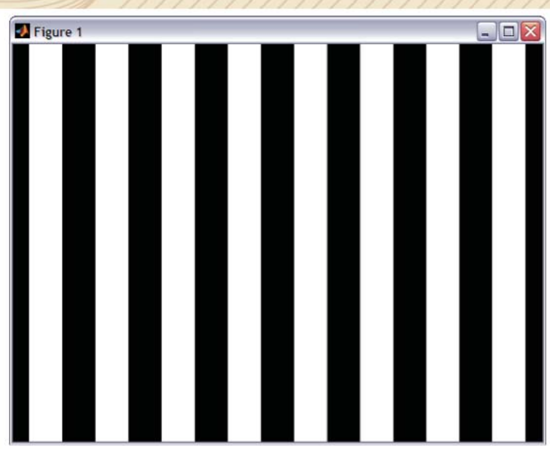
Controlling a Projector for Structured Lighting

Operating System

- A projector is just another display…

Matlab

- Fullscreen display not fully supported (cannot remove window decorations)




```
imagesc(P{1}(:,:,5));
axis image off;
set(gca,'Pos',[0 0 1 1]);
set(gcf,'MenuBar','none');
set(gcf,'Pos',[-1023 283 1024 768]);
```

Controlling a Projector for Structured Lighting

Operating System

- A projector is just another display…

Matlab

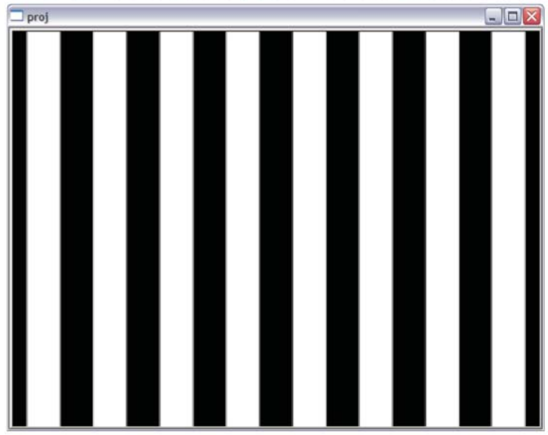
- Fullscreen display not fully supported (cannot remove window decorations)
- Use Java to control displays (slow, multiple display problems)
- Use the *Psychophysics Toolbox* (Screen.m wraps OpenGL functions)

OpenCV

- Cannot remove window decorations
- cvNamedWindow/cvMoveWindow

General

- Use OpenGL wrappers




```
cvNamedWindow("proj",CV_WINDOW_AUTOSIZE);
IplImage* proj_frame =
   cvCreateImage(cvSize(w,h),8,3);
cvSet(proj_frame,cvScalar(0,0,0));
cvShowImage("proj",proj_frame);
cvMoveWindow("proj",-w-7,-33);
cvWaitKey(1);
```

http://opencv.willowgarage.com/wiki

Course Schedule

- Structured Lighting
- ➢ ***Robust Pixel Classification***
- Projector Calibration / Structured Light Reconstruction
- Surface Reconstruction from Point Clouds
- Elementary Mesh Processing

Robust Pixel Classification

- **Robust pixel classification for 3d modeling with structured light**, by Yi Xu and Daniel G. Aliaga, in Proceedings of Graphics Interface 2007, GI ’07, pages 233–240, New York, NY, USA, 2007. ACM.
- **Fast separation of direct and global components of a scene using high frequency illumination**, by Shree K. Nayar, Gurunandan Krishnan, Michael D. Grossberg, and Ramesh Raskar, in ACM Trans. Graph., 25(3):935–944, July 2006.

Robust Pixel Classification

- The intensity of a pixel can be decomposed into the direct component and indirect (or global) component.
- The direct component is due to light bouncing off the surface in a single reflection.
- The indirect component is due to multiple reflections (e.g. inter-reflections, subsurface, scattering etc.).

$$L(p) = \begin{cases} L_d(p) + \alpha L_g(p) & \text{if p is on} \\ (1-\alpha) L_g(p) & \text{if p is off,} \end{cases}$$

- where $0 \leq \alpha \leq 1$ is a fraction of activated source pixels. For example, for a random binary pattern $\alpha = 1$, and for a white pattern $\alpha = 1$.

Robust Pixel Classification

- Estimate direct and global components for each pixel using the method by Nayar
- Use two images of the scene, one with the scene lit with high-frequency illumination, and the other lit with the complementary illumination.

$$L_d(p) = L_{max}(p) - \frac{\alpha}{1-\alpha} L_{min}(p)$$
$$L_g(p) = \frac{1}{1-\alpha} L_{min}(p),$$

Robust Pixel Classification

- Projecting the code pattern i and its inverse yields two values for each pixel.
- Classify p according to the rules

$$p \text{ is } \begin{cases} \text{uncertain} & \text{if } L_d(p) < m \\ \text{on} & \text{if } L_d(p) > L_g(p) \text{ and } L_i^+(p) > L_i^-(p) \\ \text{off} & \text{if } L_d(p) > L_g(p) \text{ and } L_i^+(p) < L_i^-(p) \\ \text{off} & \text{if } L_i^+(p) < L_d(p) \text{ and } L_i^-(p) > L_g(p) \\ \text{on} & \text{if } L_i^+(p) > L_g(p) \text{ and } L_i^-(p) < L_d(p) \\ \text{uncertain} & \text{otherwise,} \end{cases}$$