Structured-Light Based Acquisition (Part 1)

CS635 Spring 2017

Daniel G. Aliaga
Department of Computer Science
Purdue University
Passive vs. Active Acquisition

- Passive
  + Just take pictures
  + Does not intrude in the environment (=passive)
    - Some surfaces cannot be acquired
    - Robustness is problematic
- Active
  + Emit “light” into the scene so as to force the generation of robust correspondence
    - Environment is intruded (=active)
Active Acquisition

• Some options:
  – Laser scanning
  – “Structured Light”
Laser Scanning
Light Stripe Scanning (Single Stripe)

- Optical triangulation
  - Project a single stripe of laser light
  - Scan it across the surface of the object
  - This is a very precise version of structured light scanning
  - Good for high resolution 3D, but needs many images and takes time
• Project laser stripe onto object
• Depth from ray-plane triangulation:
  – Intersect camera ray with light plane

\[
\begin{align*}
x &= x' \frac{z}{f} \\
y &= y' \frac{z}{f} \\
z &= \frac{-Df}{Ax' + By' + Cf}
\end{align*}
\]
Example: Laser scanner

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

Cyberware® face and head scanner
Example: Laser scanner

Digital Michelangelo Project
http://graphics.stanford.edu/projects/mich/
Example: Laser scanner

Portable scanner by Minolta
Digital Projector Structured Light

• Goal: generate correspondences so as to enable a robust 3D reconstruction
Digital Projector Structured Light

• Method:
  – Use the projector as a “pattern” generator
  – Have the camera see the “pattern” and generate 1 or more corresponded points
What are possible patterns?
- Spatial patterns
- Temporal patterns
- Color patterns
- And combinations of the above
Digital Projector Structured Light

| Time-multiplexing | Binary codes | Poedderman et al. | ✓ |  ✓ | ✓ |
|                  |             | Inokuchi et al. | ✓ | ✓ | ✓ |
|                  |             | Minou et al. | ✓ | ✓ | ✓ |
|                  |             | Trobina | ✓ | ✓ | ✓ |
|                  |             | Valkenburg and Mevius | ✓ | ✓ | ✓ |
|                  |             | Skocaj and Leonardis | ✓ | ✓ | ✓ |
|                  |             | Rocchini et al. | ✓ | ✓ | ✓ |
| n-ary codes      |             | Caspi et al. | ✓ | ✓ | ✓ |
|                  |             | Horn and Kiryati | ✓ | ✓ | ✓ |
|                  | Gray code + Phase shifting | Bergmann | ✓ | ✓ | ✓ |
|                  |             | Sansoni et al. | ✓ | ✓ | ✓ |
|                  |             | Wisor | ✓ | ✓ | ✓ |
|                  |             | Gühring | ✓ | ✓ | ✓ |
| Hybrid methods   |             | Kosuke Sato | ✓ | ✓ | ✓ |
|                  |             | Hall-Holt and Rusinkiewicz | ✓ | ✓ | ✓ |

| Spatial Neighborhood | Non-formal codification | Martuyama and Abe | ✓ | ✓ | ✓ |
|                      |                         | Durdile et al. | ✓ | ✓ | ✓ |
|                      |                         | Ito and Ishii | ✓ | ✓ | ✓ |
|                      |                         | Boyer and Kak | ✓ | ✓ | ✓ |
|                      |                         | Chen et al. | ✓ | ✓ | ✓ |
| De Bruijn sequences |                         | Hilgii and Mltre | ✓ | ✓ | ✓ |
|                      |                         | Monics et al. | ✓ | ✓ | ✓ |
|                      |                         | Vuytsako and Oosterlinck | ✓ | ✓ | ✓ |
|                      |                         | Salvi et al. | ✓ | ✓ | ✓ |
|                      |                         | Lavoie et al. | ✓ | ✓ | ✓ |
|                      |                         | Zhang et al. | ✓ | ✓ | ✓ |
| M-arrays             | Morita et al. | ✓ | ✓ | ✓ |
|                      | Petriu et al. | ✓ | ✓ | ✓ |
|                      | Kiyasu et al. | ✓ | ✓ | ✓ |
|                      | Spoelder et al. | ✓ | ✓ | ✓ |
|                      | Griffin and Yee | ✓ | ✓ | ✓ |
|                      | Davies and Nixon | ✓ | ✓ | ✓ |
|                      | Morano et al. | ✓ | ✓ | ✓ |
| Grey levels          | Carrhill and Hummel | ✓ | ✓ | ✓ |
|                      | Chazan and Kiryati | ✓ | ✓ | ✓ |
|                      | Hung | ✓ | ✓ | ✓ |
| Colour               | Tajima and Iwakawa | ✓ | ✓ | ✓ |
|                      | Smutny and Pajdla | ✓ | ✓ | ✓ |
|                      | Geng | ✓ | ✓ | ✓ |
|                      | Wust and Capson | ✓ | ✓ | ✓ |
|                      | Tatsuo Sato | ✓ | ✓ | ✓ |

Scene applicability: Static, Moving

Pixel depth: Grey levels, Colour

Coding strategy: Periodical, Absolute

Let's focus on binary striped patterns...
Binary Pattern Structured Light
Binary Coding

• Assign each pixel a unique illumination code over time [Posdamer 82]
Binary Coding

- Assign each pixel a unique illumination code over time [Posdamer 82]
$2^n - 1$ stripes in $n$ images

Example:

3 binary-encoded patterns which allows the measuring surface to be divided in 8 sub-regions
## Binary vs Gray Codes

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Binary</th>
<th>Gray Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000</td>
<td>0000</td>
</tr>
<tr>
<td>1</td>
<td>0001</td>
<td>0001</td>
</tr>
<tr>
<td>2</td>
<td>0010</td>
<td>0011</td>
</tr>
<tr>
<td>3</td>
<td>0011</td>
<td>0010</td>
</tr>
<tr>
<td>4</td>
<td>0100</td>
<td>0110</td>
</tr>
<tr>
<td>5</td>
<td>0101</td>
<td>0111</td>
</tr>
<tr>
<td>6</td>
<td>0110</td>
<td>0101</td>
</tr>
<tr>
<td>7</td>
<td>0111</td>
<td>0100</td>
</tr>
<tr>
<td>8</td>
<td>1000</td>
<td>1100</td>
</tr>
<tr>
<td>9</td>
<td>1001</td>
<td>1101</td>
</tr>
<tr>
<td>10</td>
<td>1010</td>
<td>1111</td>
</tr>
</tbody>
</table>
Binary vs Gray Codes

Pattern 1

Pattern 2

Pattern 3

Binary code

Gray code
Standard Pixel Classification

ON(1)   OFF(0)   Uncertain
Pixel Classification Challenges

Illuminated (ON)
Non-illuminated (OFF)

?
Standard Pixel Classification

• Interval

• Common methods
  – Simple threshold
  – Albedo threshold
  – Dual pattern
1. Simple Threshold

User specifies one threshold $t$ for all pixels

$p$ is ON

$q$ is OFF

$p_{\text{off}}$, $t$, $p_{\text{on}}$
1. Simple Threshold

User specifies two thresholds $t_1$, $t_2$

$p$ is ON

$q$ is OFF

$p_{\text{on}}$ $t_1$ $t_2$ $p_{\text{off}}$
2. Albedo Threshold

Compute the albedo $t_p$ for each pixel

$P_{off}$ $t_p$ $P_{on}$

Albedo image
2. Albedo Threshold

Compute the albedo $t_p$ for each pixel

$p$ is OFF

Albedo image
3. Dual Pattern: Pattern and Inverse

Without explicitly computing $t_p$

$p$ is ON

$p$ is OFF

Inverse pattern
Limitations of Standard Methods

- All three methods assume the two intervals do not overlap
  - This is incorrect when there is strong indirect (global) light

- Haven’t actually established the correct bounds
Example Reconstruction

(lost samples are due to missing and incorrect classifications)
Key Observations

• We can estimate tight intensity value bounds for when the pixel is ON and for when it is OFF.

• A pixel is classifiable when its intensity value falls into one interval but not in the other.

[Xu07, Xu09]
Example Comparison

Using standard pixel classification

Point splatting of the reconstructed living room scene
Pixel Intensity \((p \text{ is ON})\)

\[ p = d + i_{on} \]

\(p\) is directly illuminated

(direct) \quad (indirect)
Pixel Intensity ($p$ is OFF)

$p$ is not directly illuminated

$$p = i_{\text{off}}$$

(indirect)
Pixel Intensity

\[ p = \begin{cases} 
  d + i_{on} & \text{If } p \text{ is ON} \\
  i_{off} & \text{If } p \text{ is OFF} 
\end{cases} \]

• Chicken and egg problem is
  – Need to know \( d, i_{on}, i_{off} \) to classify a pixel.
  – Need to classify a pixel to know \( d, i_{on}, i_{off} \)

 ALL white pattern

 (all projector pixels on)
Direct and Indirect Separation

• Direct and indirect (global) components of each pixel under ALL white pattern can be separated easily (Nayar et al. SIGGRAPH’06).

\[ p = d_{\text{total}} + i_{\text{total}} \]
Direct and Indirect Separation

• Project high frequency binary pattern and its inverse to separate light components.

• Structured light patterns include the separation patterns.
  – Thus, separation can be applied to previously captured data to obtain per pixel

\[ p = d_{\text{total}} + i_{\text{total}} \]
Pixel Classification Scenarios

\[ d_{total} > i_{total} \]

0 \[ P_{off} \] \[ i_{total} \] \[ d_{total} \] \[ P_{on} \] \[ d_{total} + i_{total} \] 

0 OFF ON 255
Pixel Classification Scenarios

\[ d_{\text{total}} \approx 0 \]

0 \quad P_{\text{off}} \quad i_{\text{total}} \quad d_{\text{total}} \quad P_{\text{on}} \quad d_{\text{total}} + i_{\text{total}}

0 \quad \text{Uncertain} \quad 255
Pixel Classification Scenarios

\[ d_{\text{total}} \leq i_{\text{total}} \]

0 \hspace{1cm} P_{\text{off}} \hspace{1cm} i_{\text{total}} \hspace{1cm} P_{\text{on}} \hspace{1cm} d_{\text{total}+i_{\text{total}}}

0 \hspace{1cm} \text{OFF} \hspace{1cm} \text{Uncertain} \hspace{1cm} \text{ON} \hspace{1cm} 255
Single Pattern Classification Rules

- \( d_{total} < m \rightarrow \text{pixel is uncertain} \)
- \( p < \min(d_{total}, i_{total}) \rightarrow \text{pixel is off} \)
- \( p > \max(d_{total}, i_{total}) \rightarrow \text{pixel is on} \)
- otherwise \( \rightarrow \text{pixel is uncertain} \)
Classification Results

<table>
<thead>
<tr>
<th>ON(1)</th>
<th>OFF(0)</th>
<th>Uncertain</th>
</tr>
</thead>
</table>

Hand-painted ground truth
Classification Results

ON(1)  OFF(0)  Uncertain

Hand painted ground truth
Increased Reconstructed Points

Using standard pixel classification

Using our pixel classification

Point splatting of the reconstructed living room scene
Increased Reconstructed Points

Using standard pixel classification

Using our pixel classification
Increased Reconstructed Points
Increased Reconstructed Points

Point splatting of the reconstructed wooden objects

Using standard pixel classification
Using our pixel classification
Increased Reconstructed Points

Using standard pixel classification

Using our pixel classification
Increased Reconstructed Points

Point splatting of the reconstructed shiny objects

Bigger splat size                       Smaller splat size

Using standard pixel classification         Using our pixel classification
Another issue...
Another issue...

- Classify...
  - pixels at stripe boundaries?
  - pixels at strip middle?
  - all pixels?