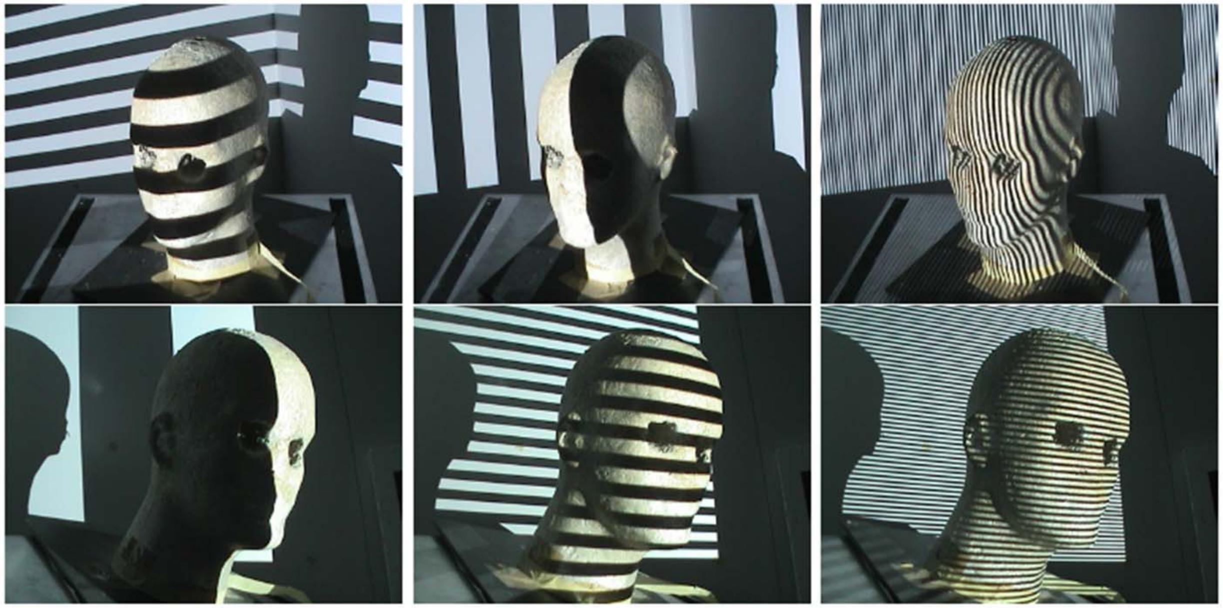


Structured Light II

Guido Gerig

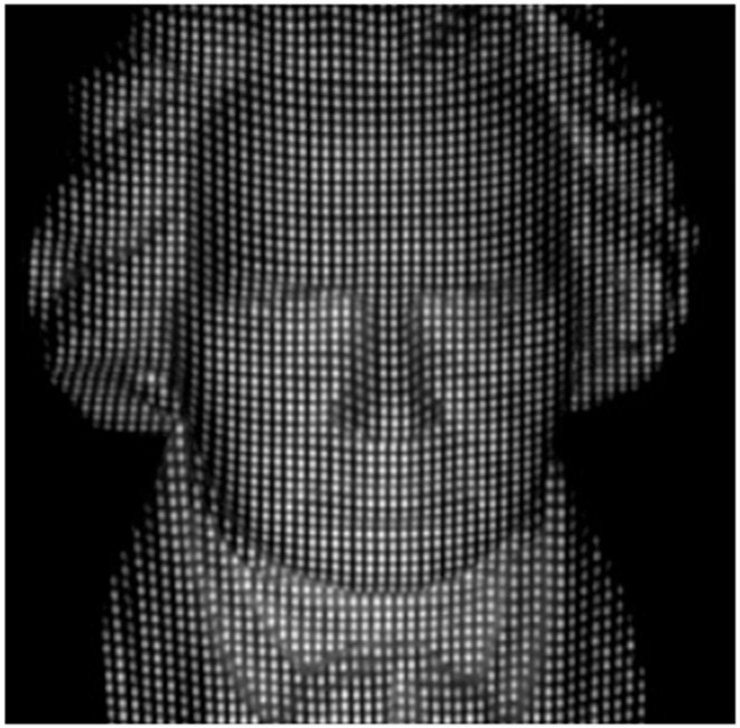
CS 6320, Spring 2013

(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

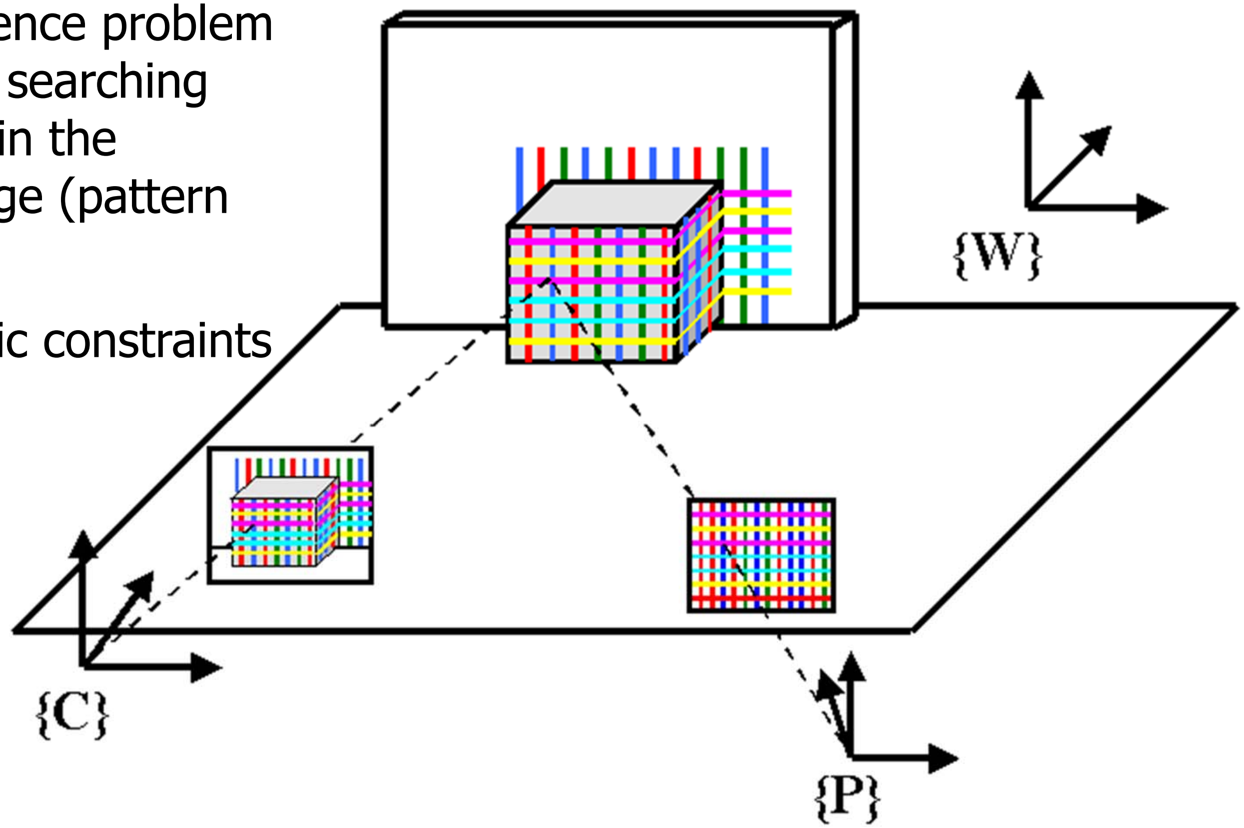
Variant

- Pattern projection
  - project a pattern instead of a single point
  - needs only a single image, one-shot recording
  - ...**but** matching is no longer unique (although still easier)
  - more on this later



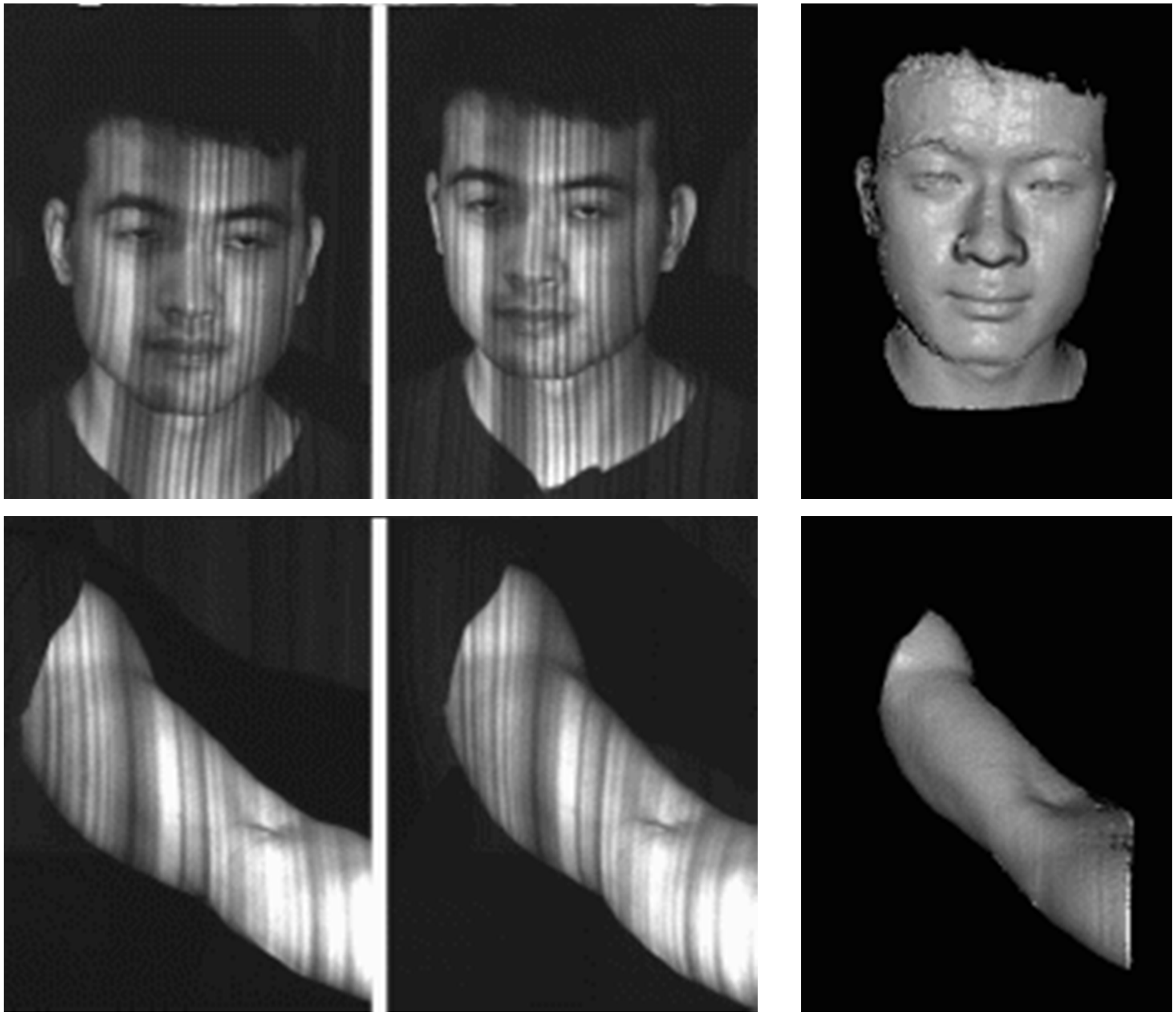
Active triangulation: Structured light

- One of the cameras is replaced by a light emitter
- Correspondence problem is solved by searching the pattern in the camera image (pattern decoding)
- No geometric constraints



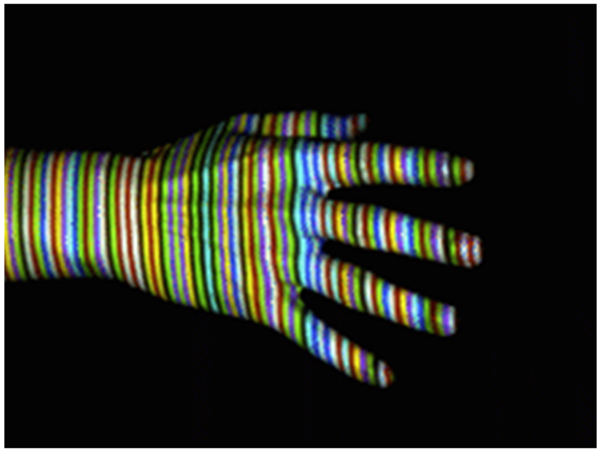
Space-time stereo

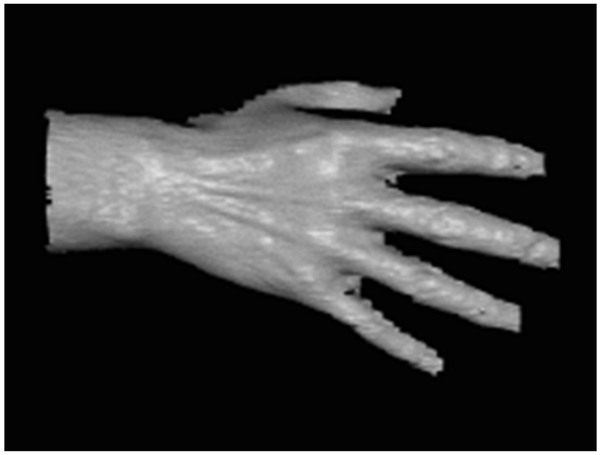
Zhang, Curless and Seitz, CVPR'03



Faster Acquisition?

- Project multiple stripes/patterns simultaneously.
- Correspondence problem: which stripe/pattern is which? How to uniquely identify patterns?





Zhang 2002: Works in real-time and on dynamic scenes

Coded structured light

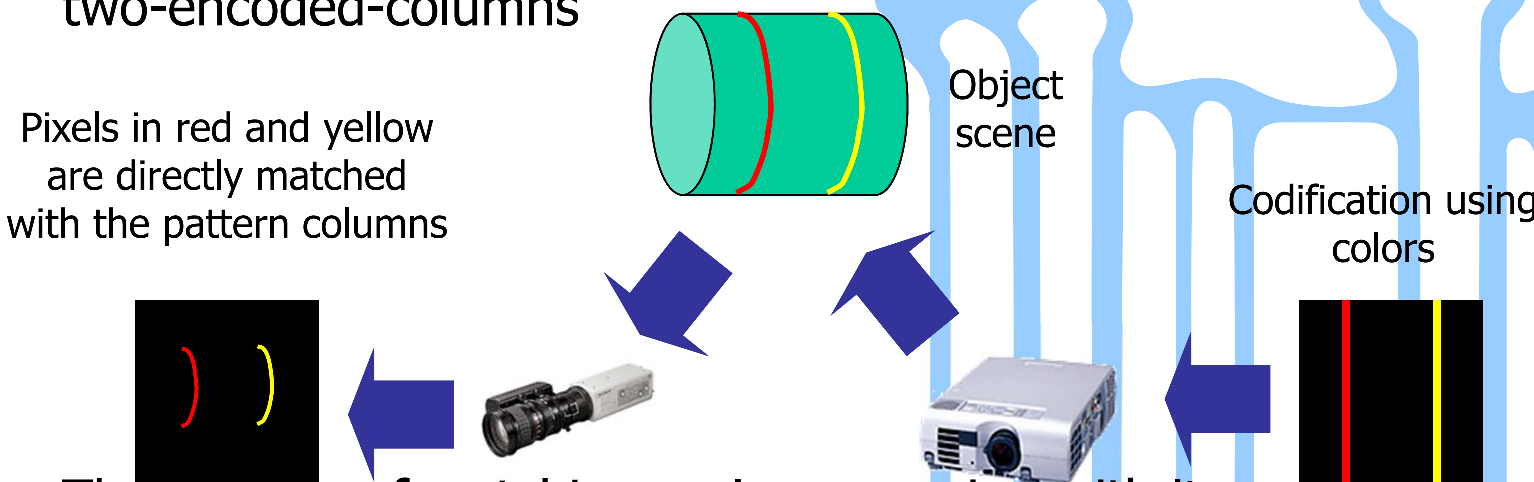
- Correspondence without need for geometrical constraints
- For dense correspondence, we need many light planes:
  - Move the projection device
  - Project many stripes at once: needs encoding
- Each pixel set is distinguishable by its encoding
- Codewords for pixels:
  - Grey levels
  - Color
  - Geometrical considerations

Codeword Classification

- Time-multiplexing:
  - Binary codes
  - N-ary codes
  - Gray code + phase shift
- Spatial Codification
  - De Bruijn sequences
  - M-arrays
- Direct encoding
  - Grey levels
  - Colour

Pattern encoding/decoding

- A pattern is encoded when after projecting it onto a surface, a set of regions of the observed projection can be easily matched with the original pattern. Example: pattern with two-encoded-columns

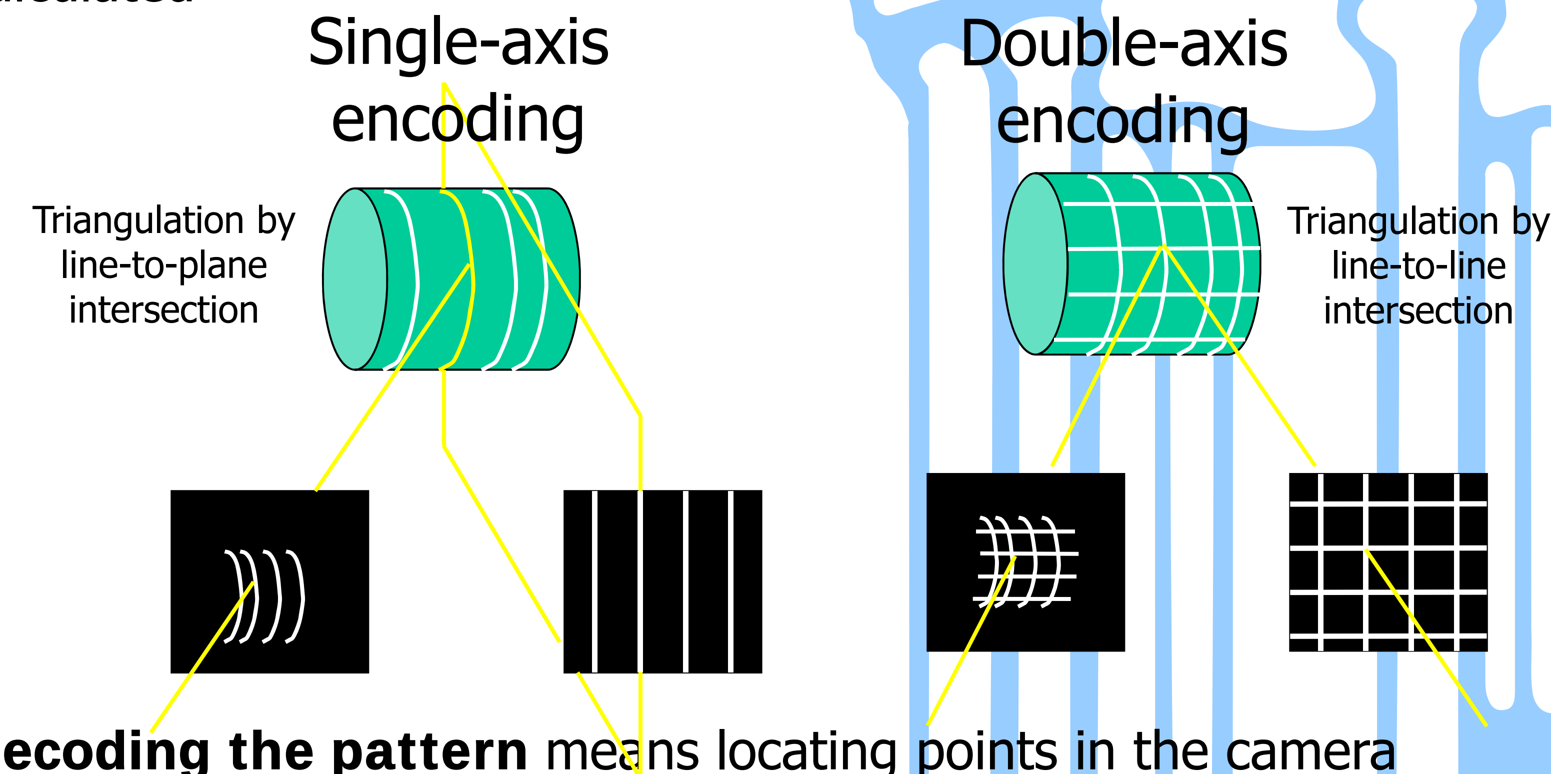


- The process of matching an image region with its corresponding pattern region is known as pattern decoding → similar to searching correspondences

- Decoding a projected pattern allows a large set of correspondences to be easily found thanks to the a priori knowledge of the light pattern

Pattern encoding/decoding

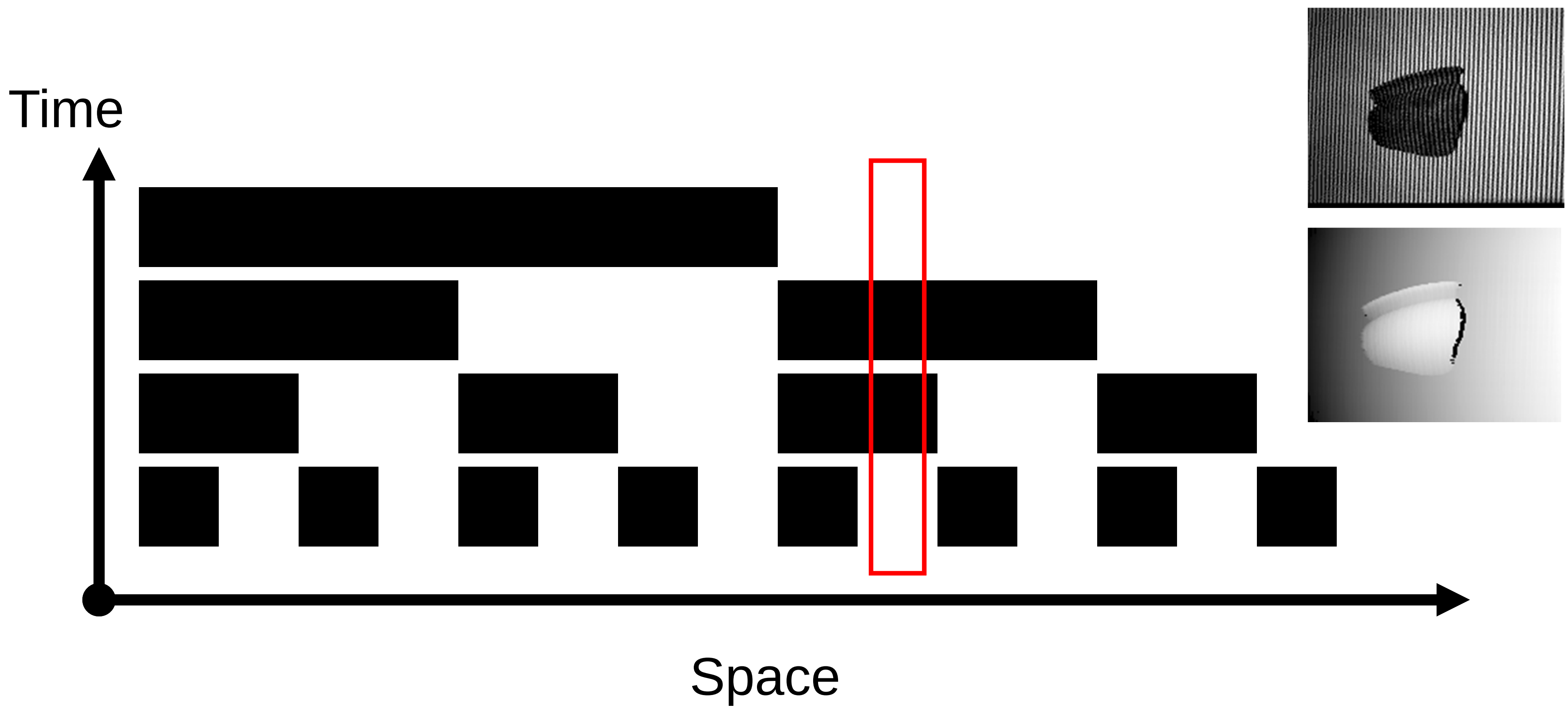
- Two ways of encoding the correspondences: single and double axis codification $\Rightarrow$ it determines how the triangulation is calculated



- **Decoding the pattern** means locating points in the camera image whose corresponding point in the projector pattern is a priori known

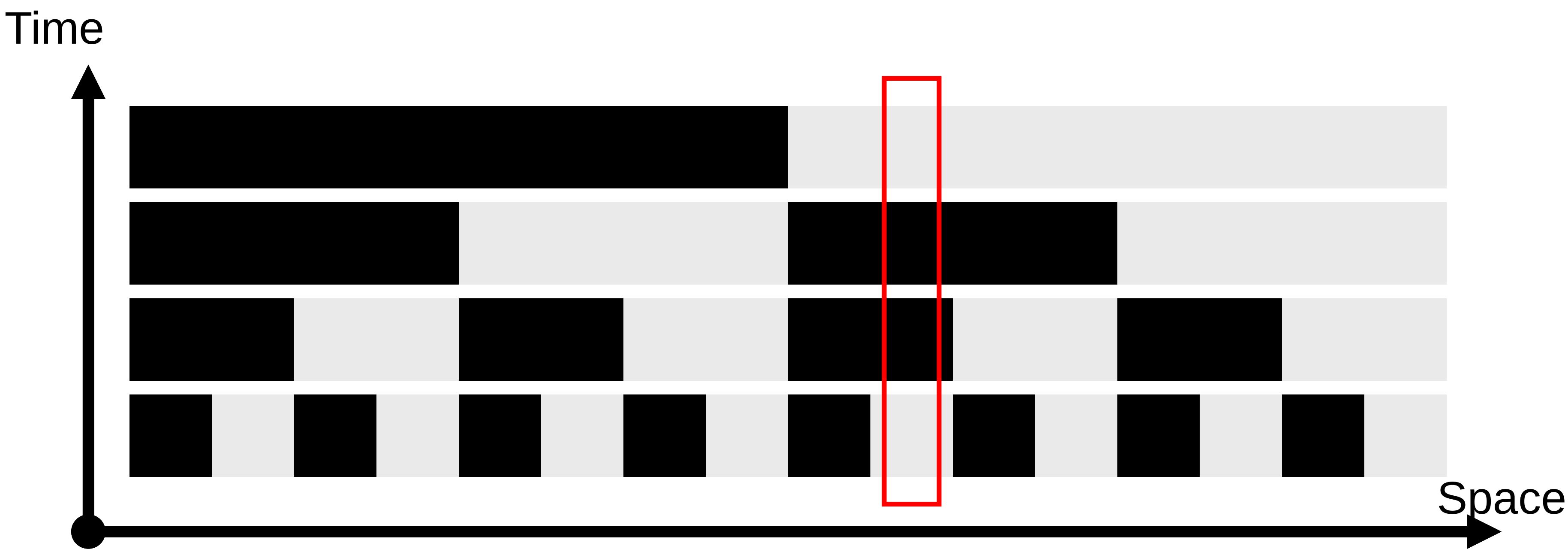
Time-Coded Light Patterns

- Assign each stripe a unique illumination code over time [Posdamer 82]



Binary Coding: Bit Plane Stack

- Assign each stripe a unique illumination code over time [Posdamer 82]



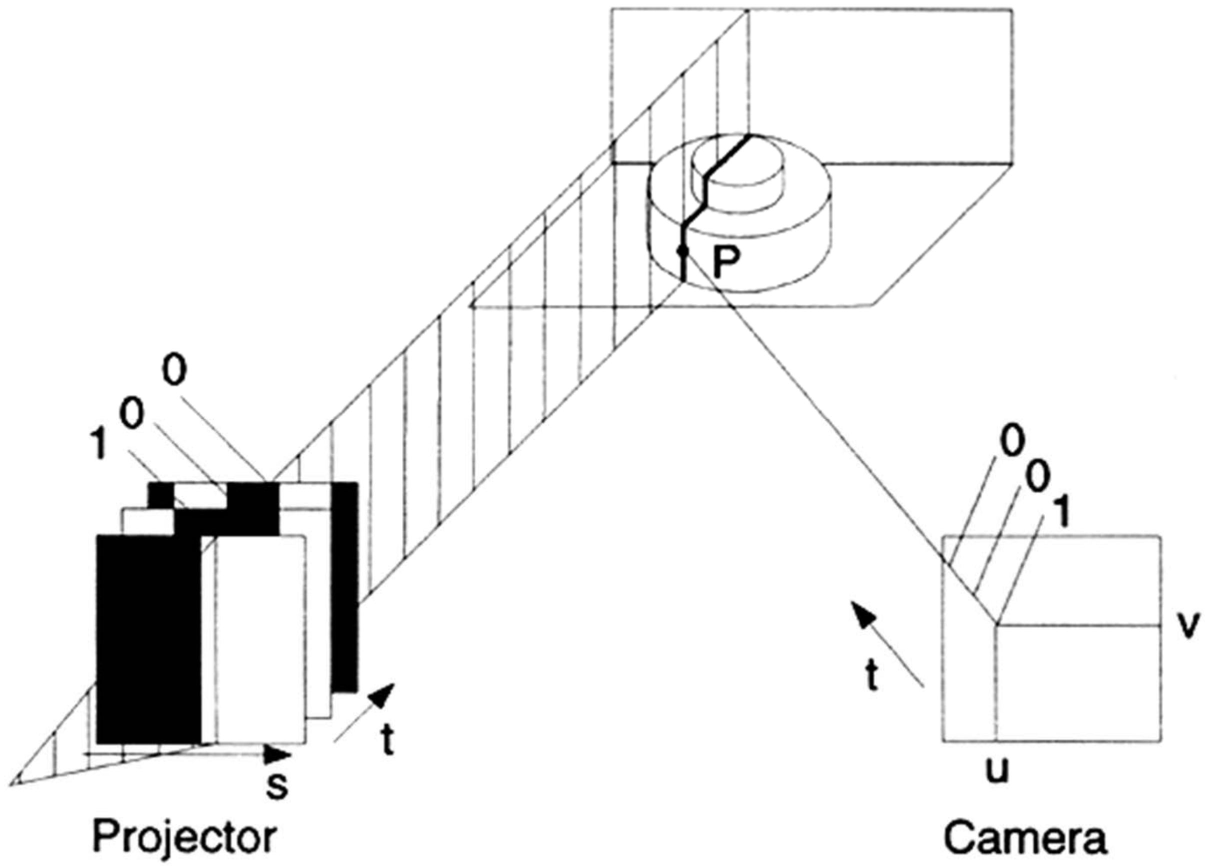
1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0

Binary Coding (II)

- **Binary coding**
  - Only two illumination levels are commonly used, which are coded as 0 and 1.
  - **Gray code** can be used for robustness (adjacent stripes must only differ in 1 bit)
  - Every encoded point is identified by the sequence of intensities that it receives
  - n patterns must be projected in order to encode 2^n stripes

  - Advantages
    - Easy to segment the image patterns
    - Need a large number of patterns to be projected

Binary Encoded Light Stripes



Binary Encoded Light Stripes

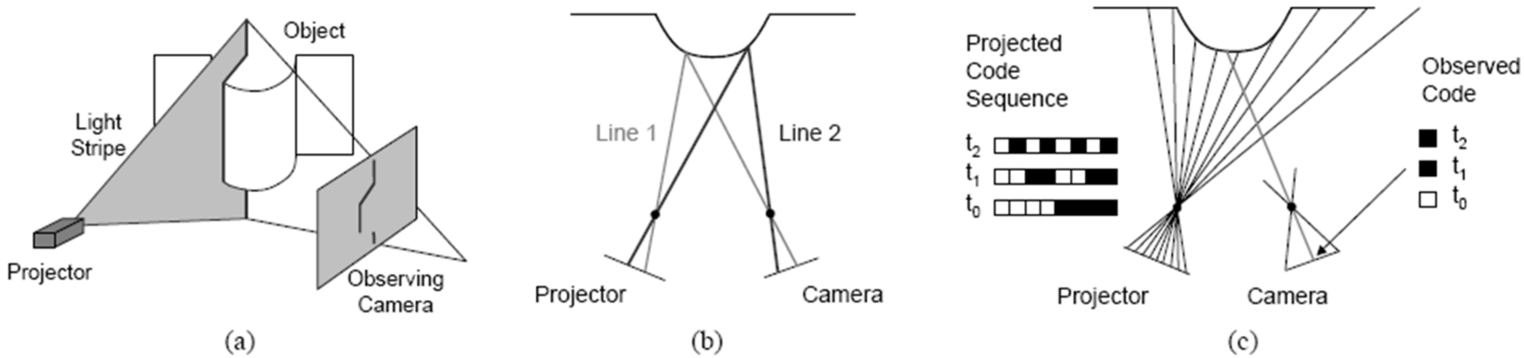
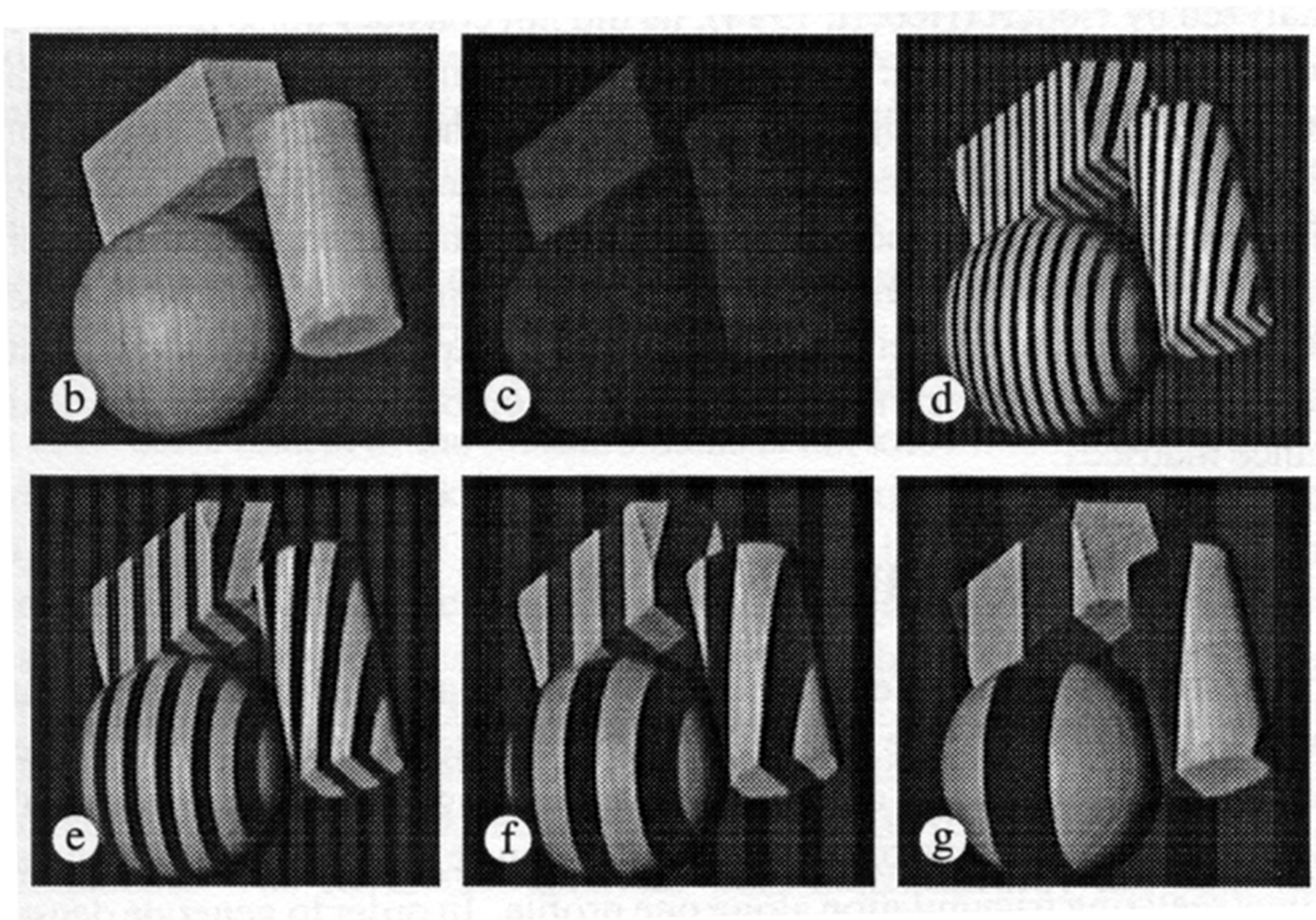


Figure 2. (a) Standard light sectioning with one light stripe. (b) Top view of light sectioning using more than one stripe. (c) Light sectioning using structured light.

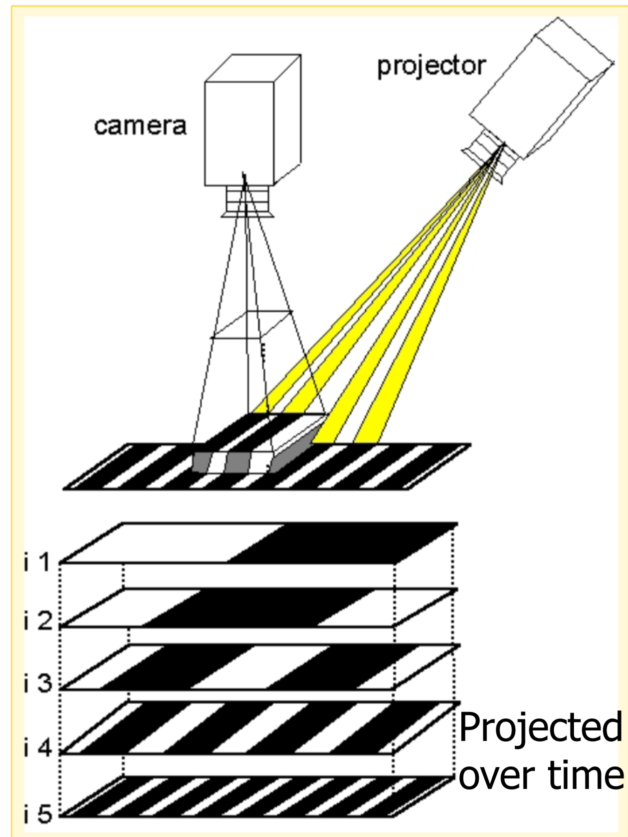
- Set of light planes are projected into the scene
- Individual light planes are indexed by an encoding scheme for the light patterns
  - Obtained images form a bit-plane stack
  - Bit-plane stack is used to uniquely address the light plane corresponding to every image point

Binary code projection



Time Multiplexing

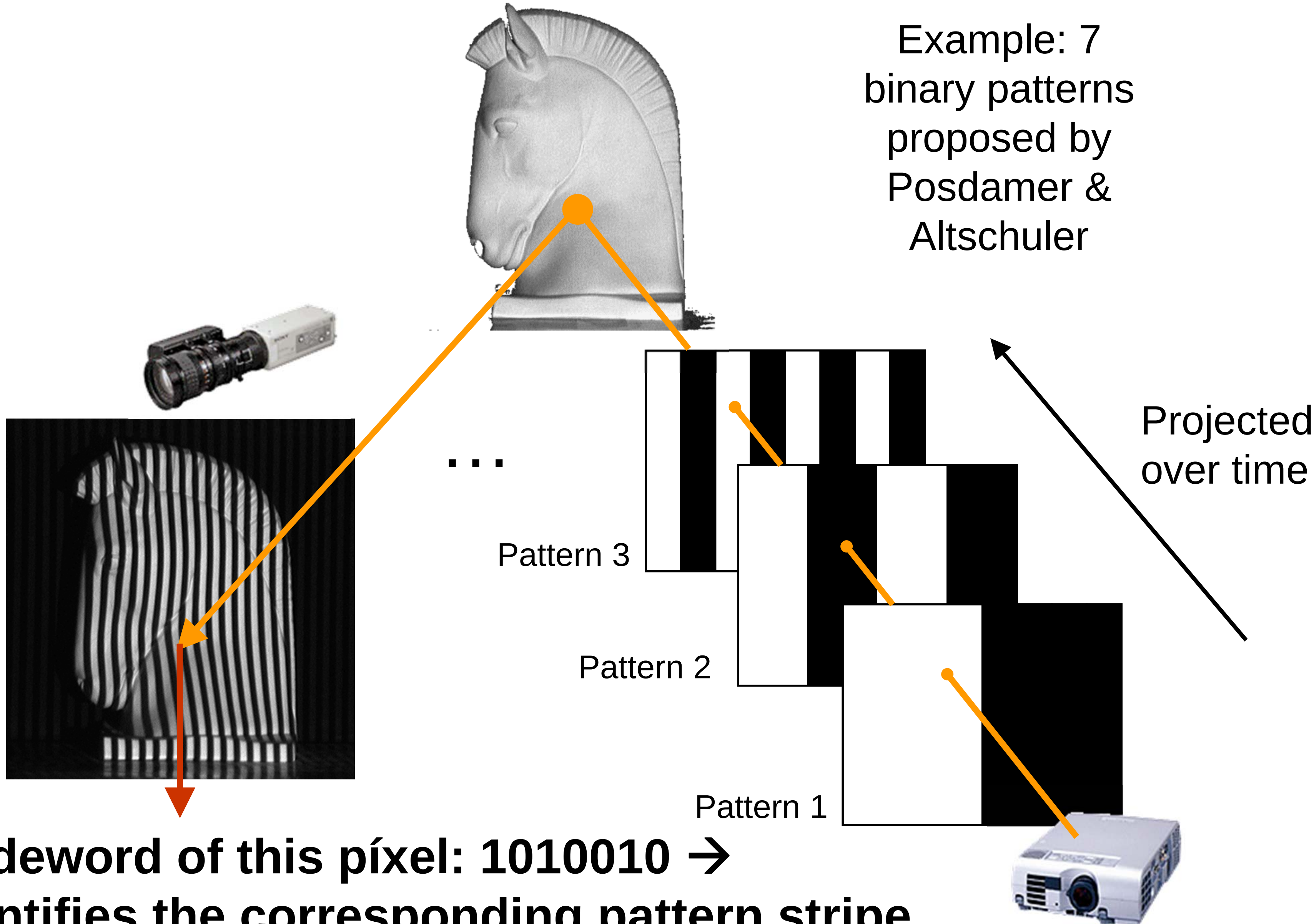
- A set of patterns are successively projected onto the measuring surface, codeword for a given pixel is formed by a sequence of patterns.
- The most common structure of the patterns is a sequence of stripes increasing its width by the time → single-axis encoding
- **Advantages:**
  - high resolution → a lot of 3D points
  - High accuracy (order of μm)
  - Robustness against colorful objects since binary patterns can be used
- **Drawbacks:**
  - Static objects only
  - Large number of patterns



Example: 5 binary-encoded patterns which allows the measuring surface to be divided in 32 sub-regions

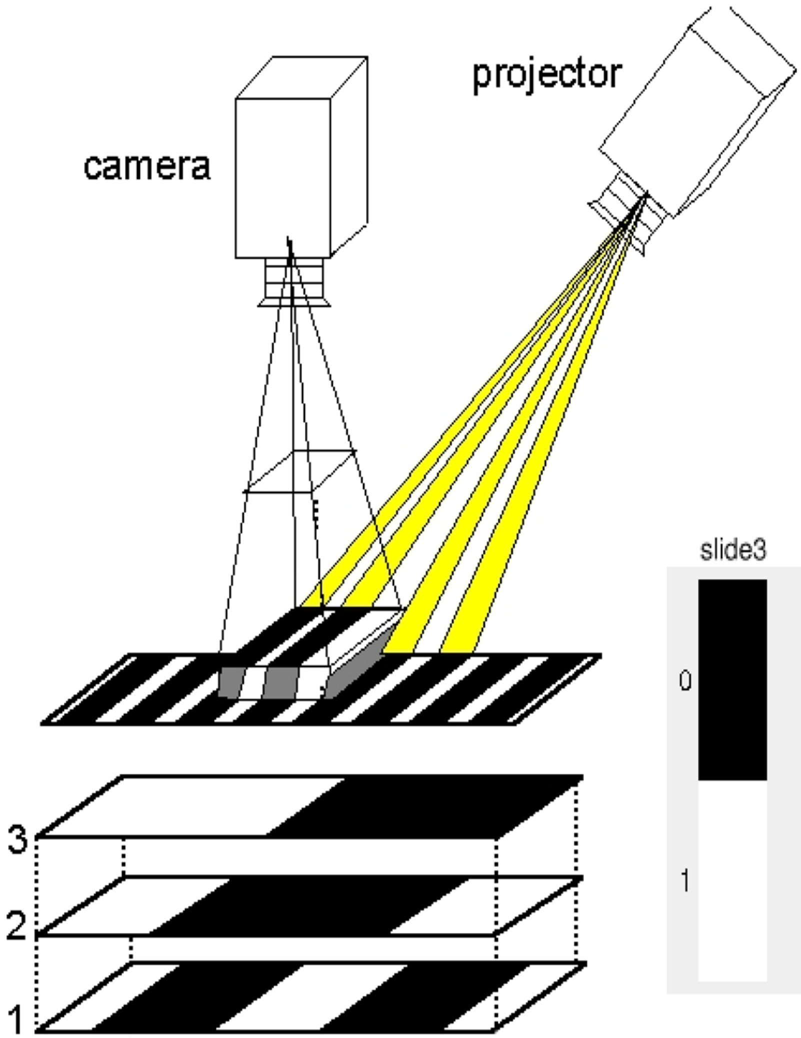
Binary Coding

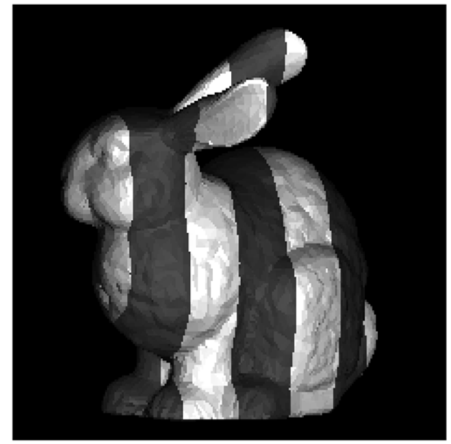
Example: 7 binary patterns proposed by Posdamer & Altschuler

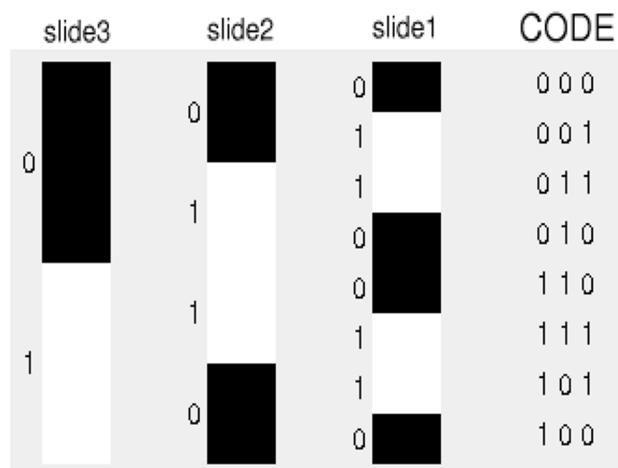


Codeword of this píxel: 1010010 → identifies the corresponding pattern stripe

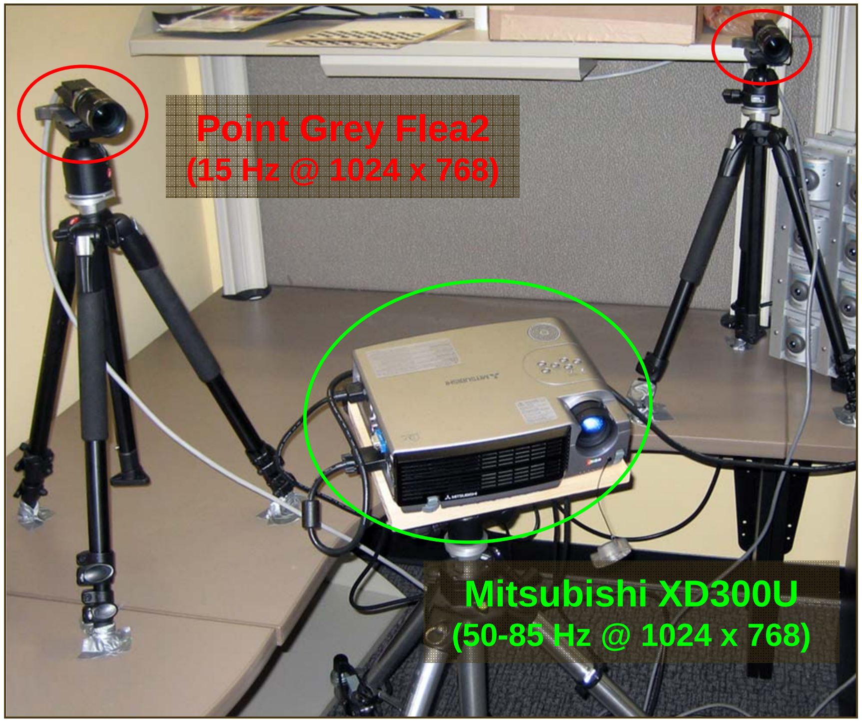
Concept Gray Code

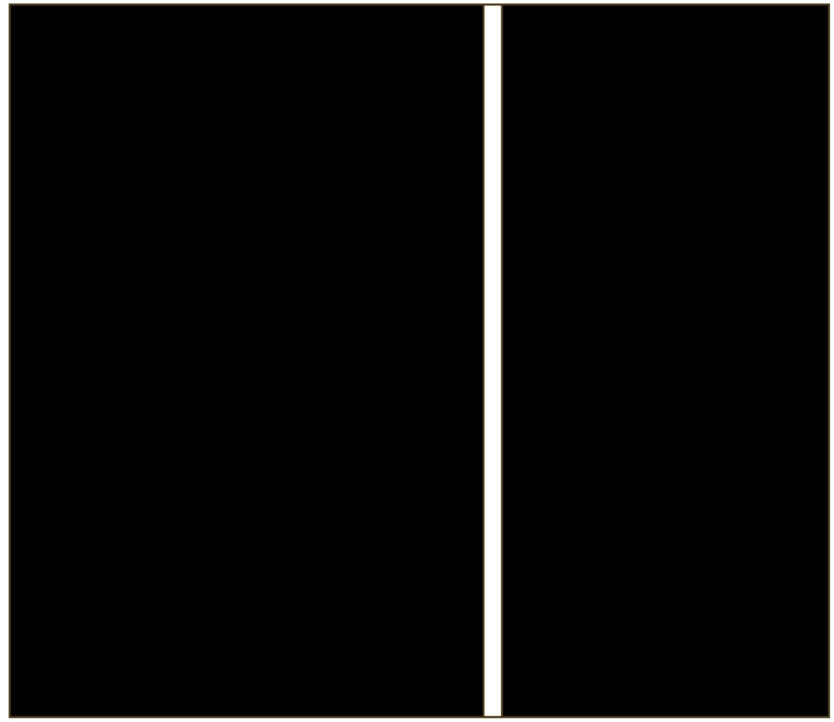






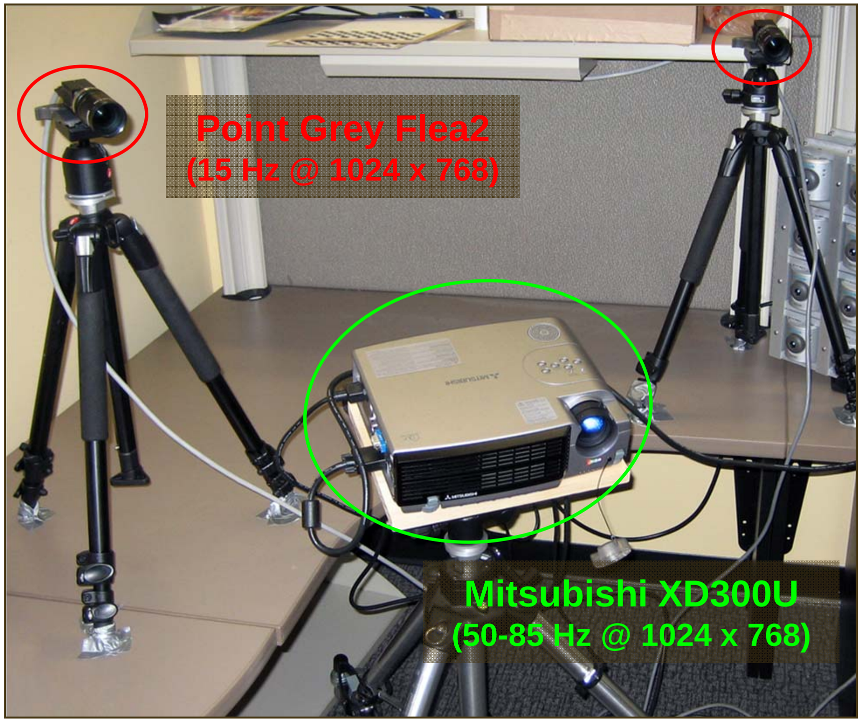
Structured Lighting: Swept-Planes Revisited

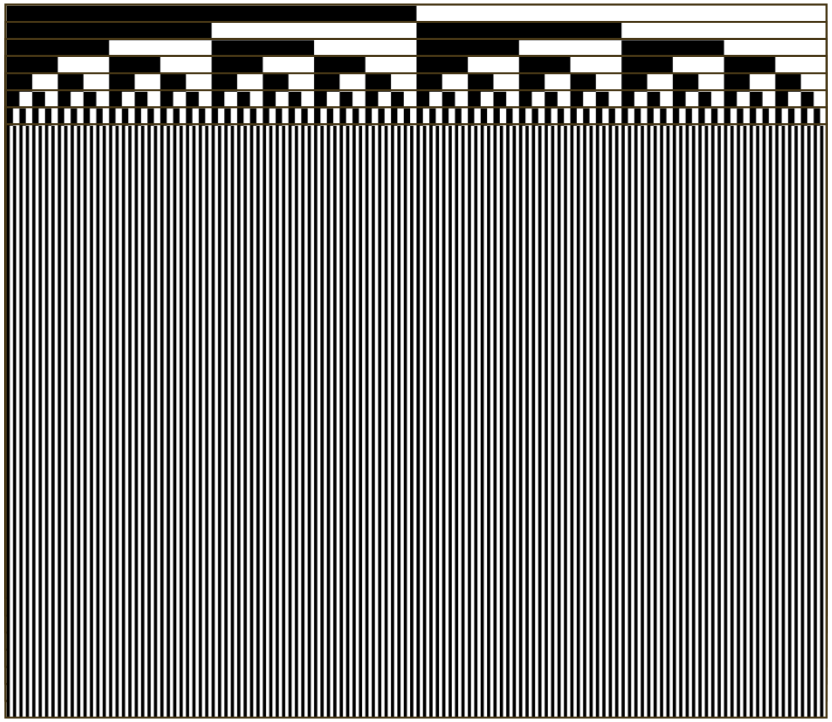




- Swept-plane scanning recovers 3D depth using ray-plane intersection
- Use a data projector to replace manually-swept laser/shadow planes
- How to assign correspondence from projector planes to camera pixels?
- Solution: Project a spatially- and temporally-encoded image sequence
- **What is the optimal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

Structured Lighting: Binary Codes

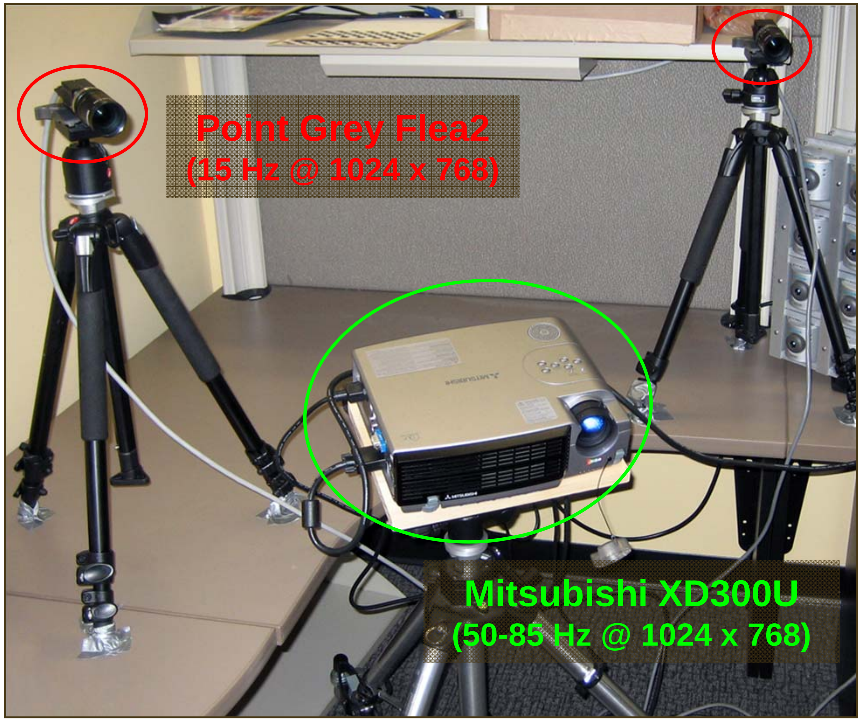


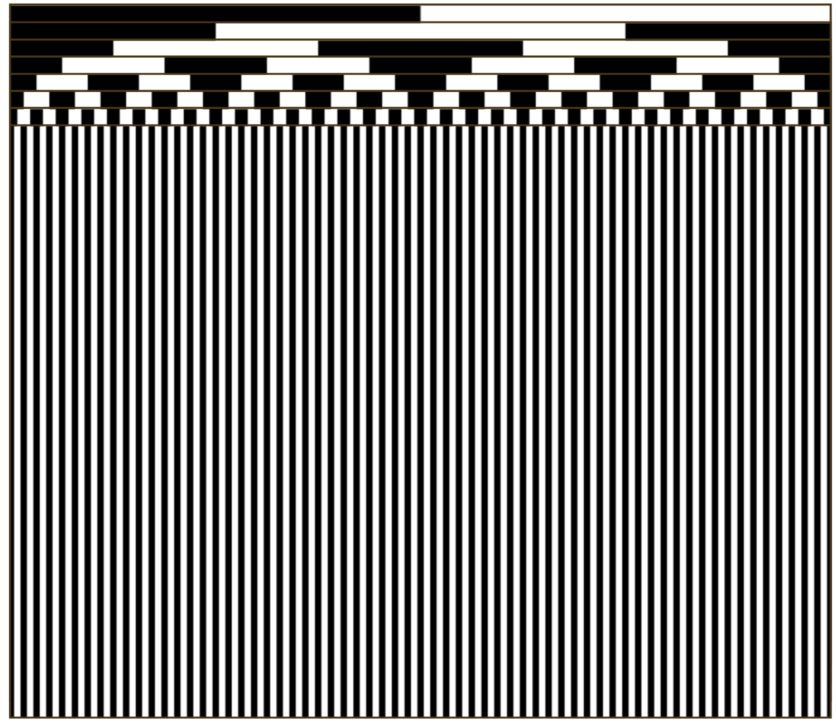


Binary Image Sequence [Posdamer and Altschuler 1982]

- 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



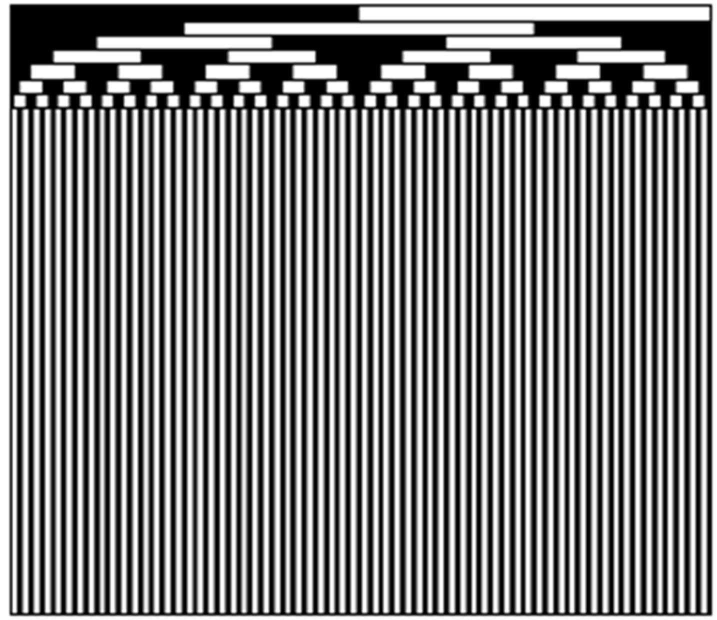


Gray Code Image Sequence [Inokuchi 1984]

- 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 Code



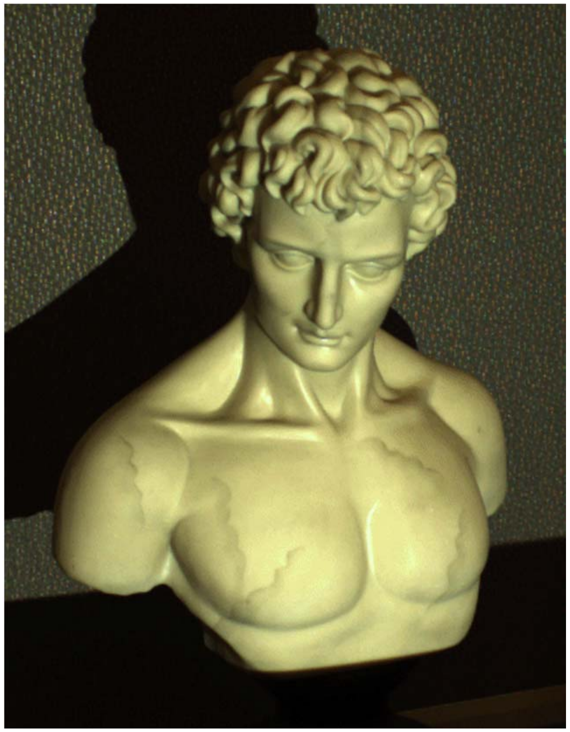
Gray Code Image Sequence

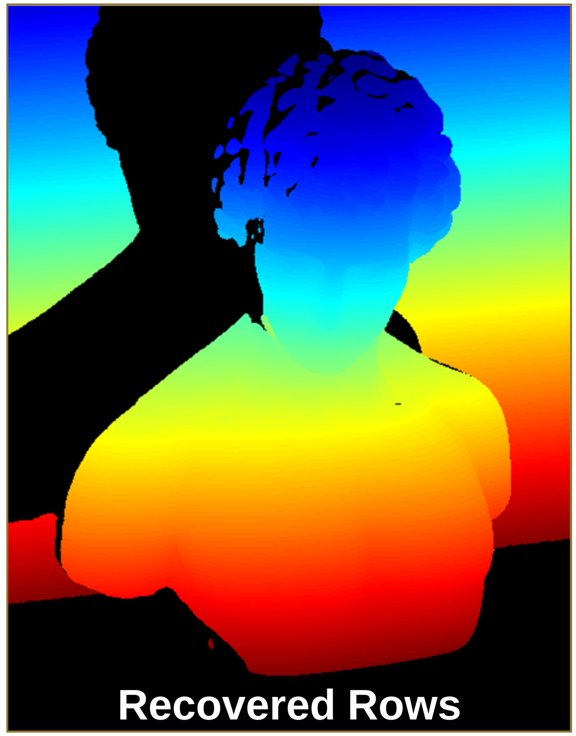
- Each image is a bit-plane of the Gray code for each projector row/column
- ~~Requires same number of images~~ as a binary image sequence, but has better performance in practice

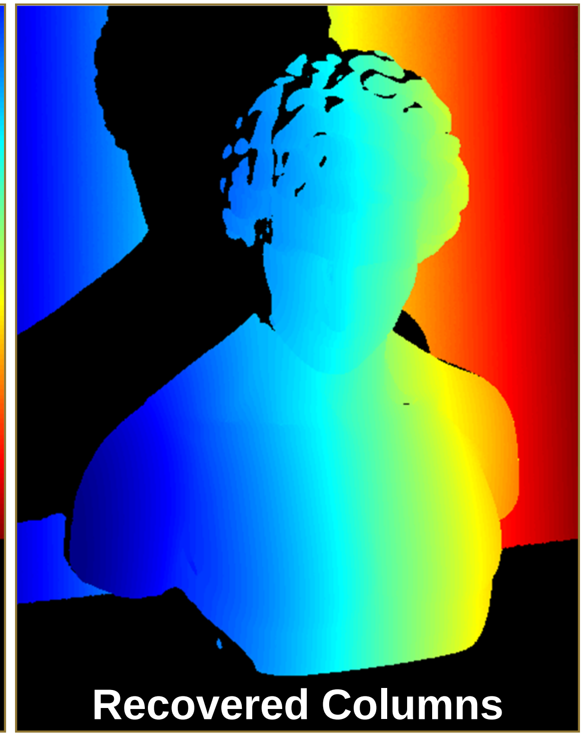
```
Bin2Gray(B,G)
G ← B
for i ← n-1 downto 0
    G[i] ← B[i+1] xor B[i]
```

•Frank Gray (-> name of coding sequence)
•Code of neighboring projector pixels only differ by 1bit, possibility of error correction!

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

Examples

http://www.youtube.com/watch?v=wryJeq3kdSg

Alternatives

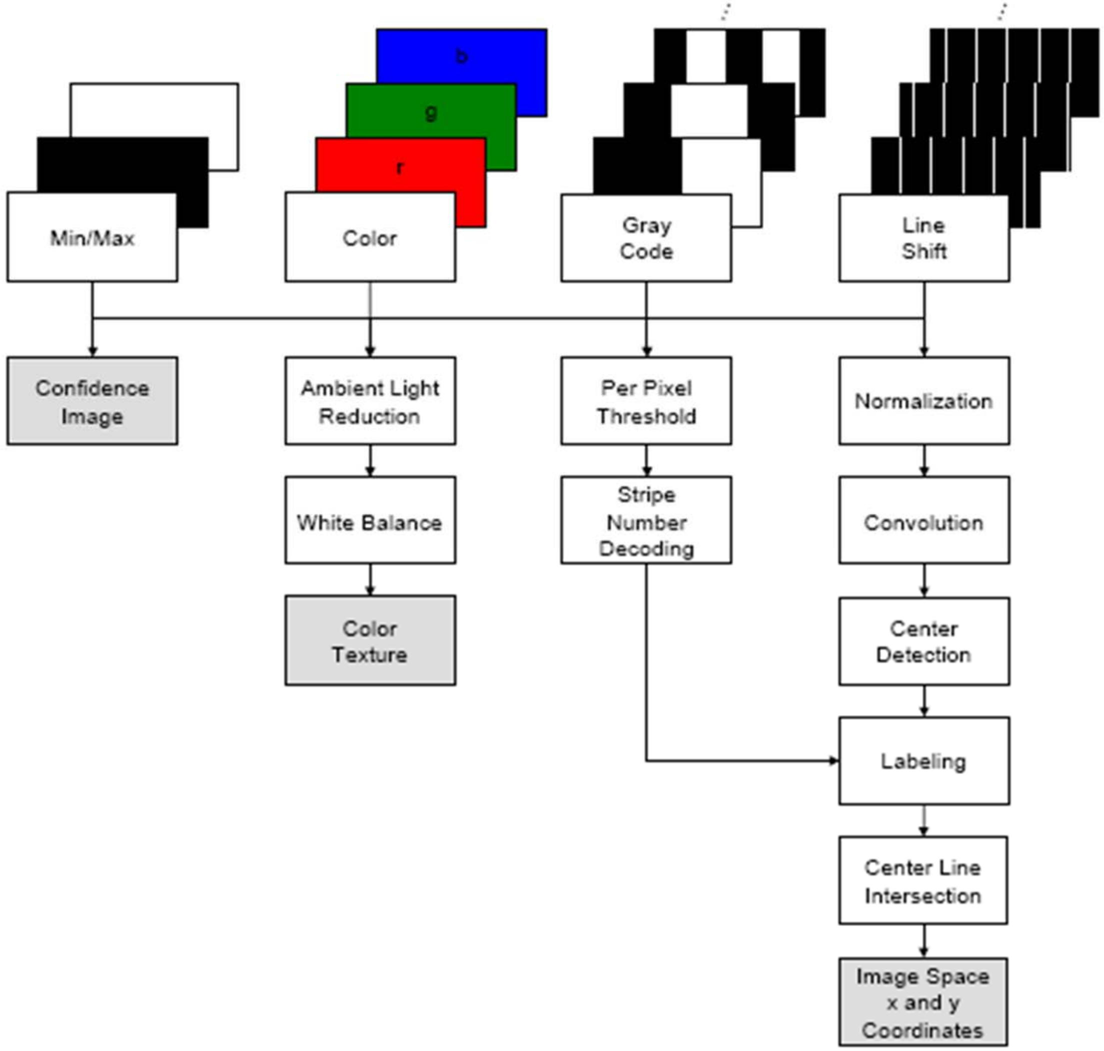


Figure 8. Flowchart of the Line Shift algorithm.

Towards higher precision and real time scanning

Line Shift Processing (Guehring et al.)

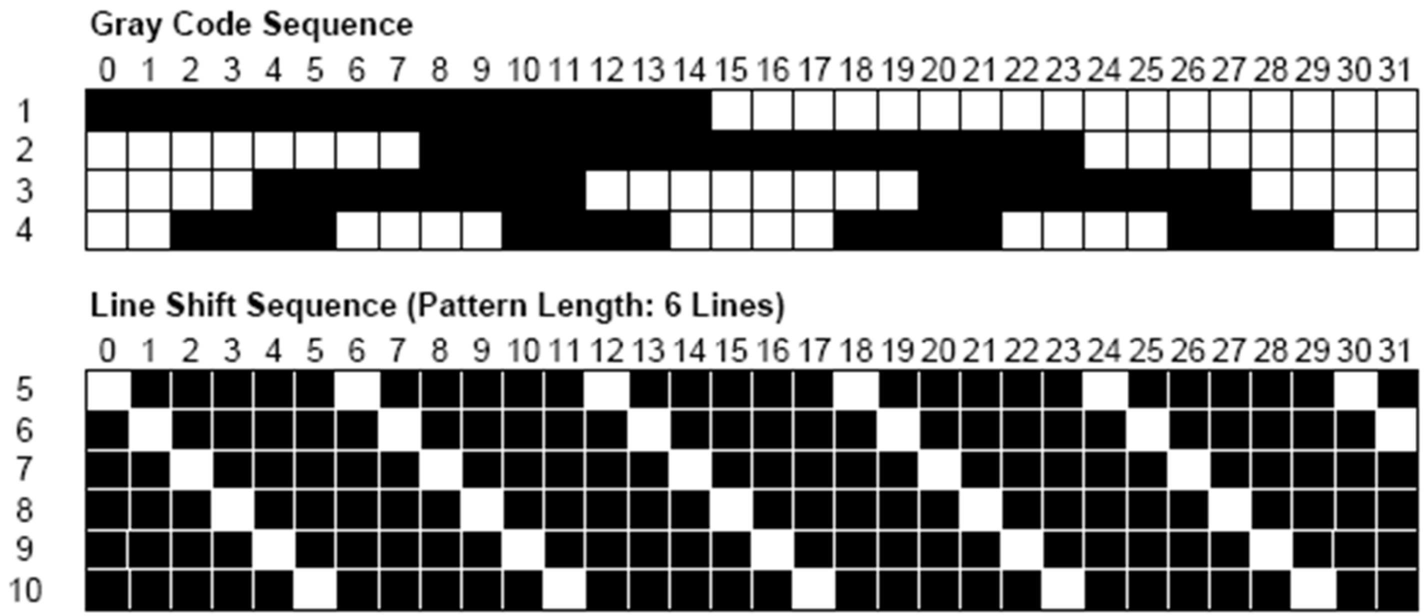
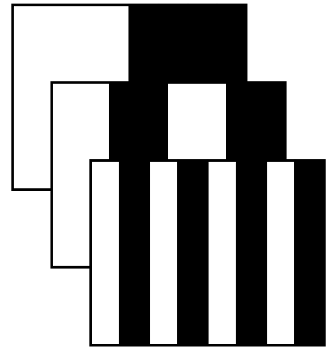


Figure 4. Gray code (top) and Line Shift (bottom) of a $n = 32$ stripe code sequence.

- Project Gray Code sequence
- Project set of 6 thin line patterns, detect line centers in image
- Combine line center positions with gray code encoding to resolve for ambiguities
- Obtain camera coordinates with subpixel accuracies for projector coordinates
- See Videometrics01-Guehring-4309-24.pdf for details

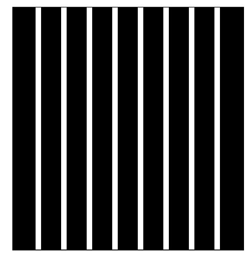
Gray Code + Phase Shifting (I)

• A sequence of binary patterns (Gray encoded) are projected in order to divide the object in regions



Example: three binary patterns divide the object in 8 regions

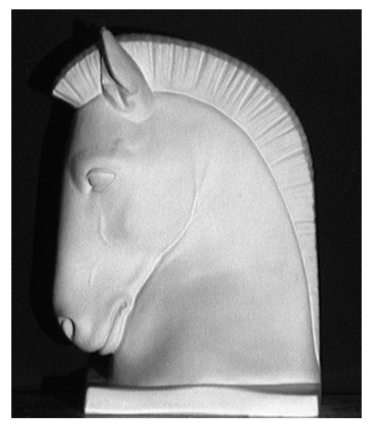
• An additional periodical pattern is projected



Without the binary patterns we would not be able to distinguish among all the projected slits

• The periodical pattern is projected several times by shifting it in one direction in order to increase the resolution of the system → similar to a laser scanner

Every slit always falls in the same region



Gühring’s line-shift technique

Industrial Inspection via line shift sequence

(a) (b) (c)

Figure 7. (a) One image of the *line shift* sequence. (b) Computed range image (z component). (c) Rendered view of the obtained surface. The holes are caused by points that have been eliminated by consistency checks, e.g. due to saturated pixels.

Phase Shift Method (Guehring etal.)

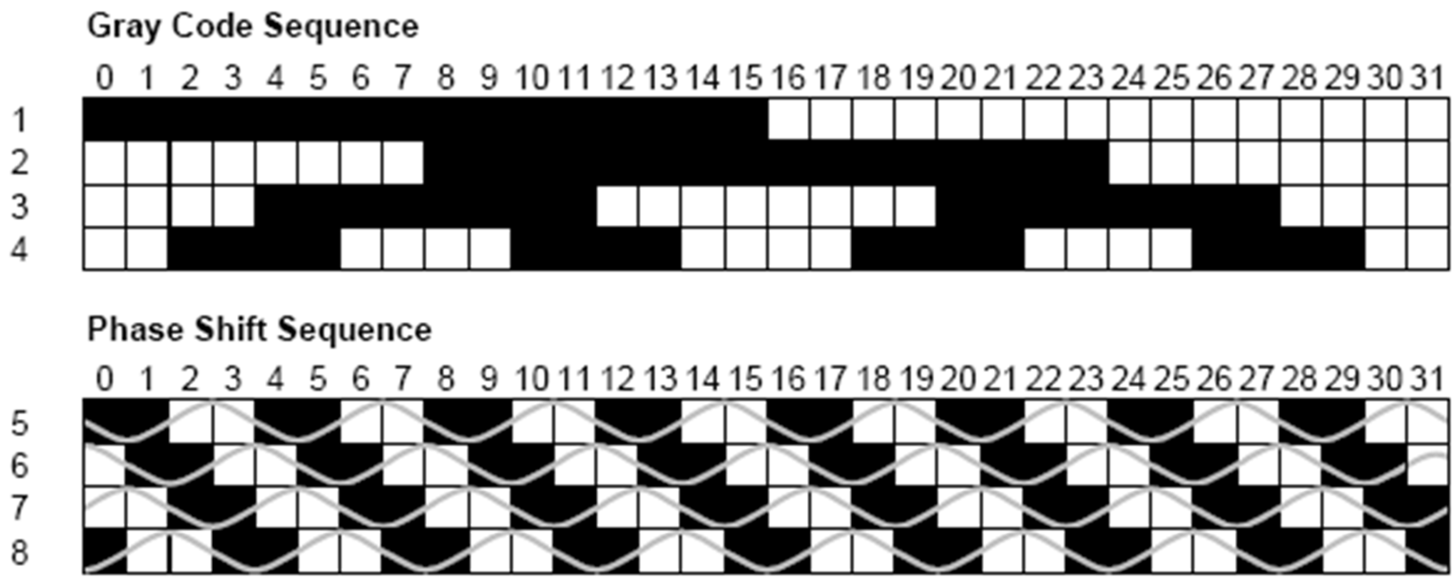


Figure 3. Gray code (top) and Phase Shift (bottom) of a $n = 32$ stripe code sequence.

- Project set of patterns
- Project sin functions
- Interpolation between adjacent light planes
- Yields for each camera pixel corresponding strip with sub-pixel accuracy!
- See Videometrics01-Guehring-4309-24.pdf for details

Gray Code + Phase Shifting (II)

- **Gray code + phase shifting**
  - Grey code: easy codification, low resolution
  - Phase shifting: high resolution, neighborhood ambiguity
  - Gray code + phase shifting: robust codification, no ambiguity and high resolution, but increase the number of projecting patterns

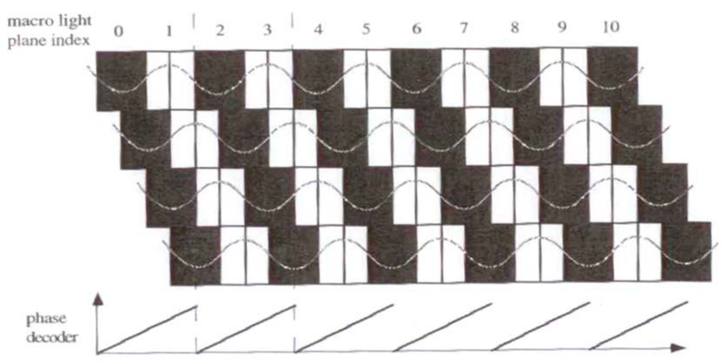
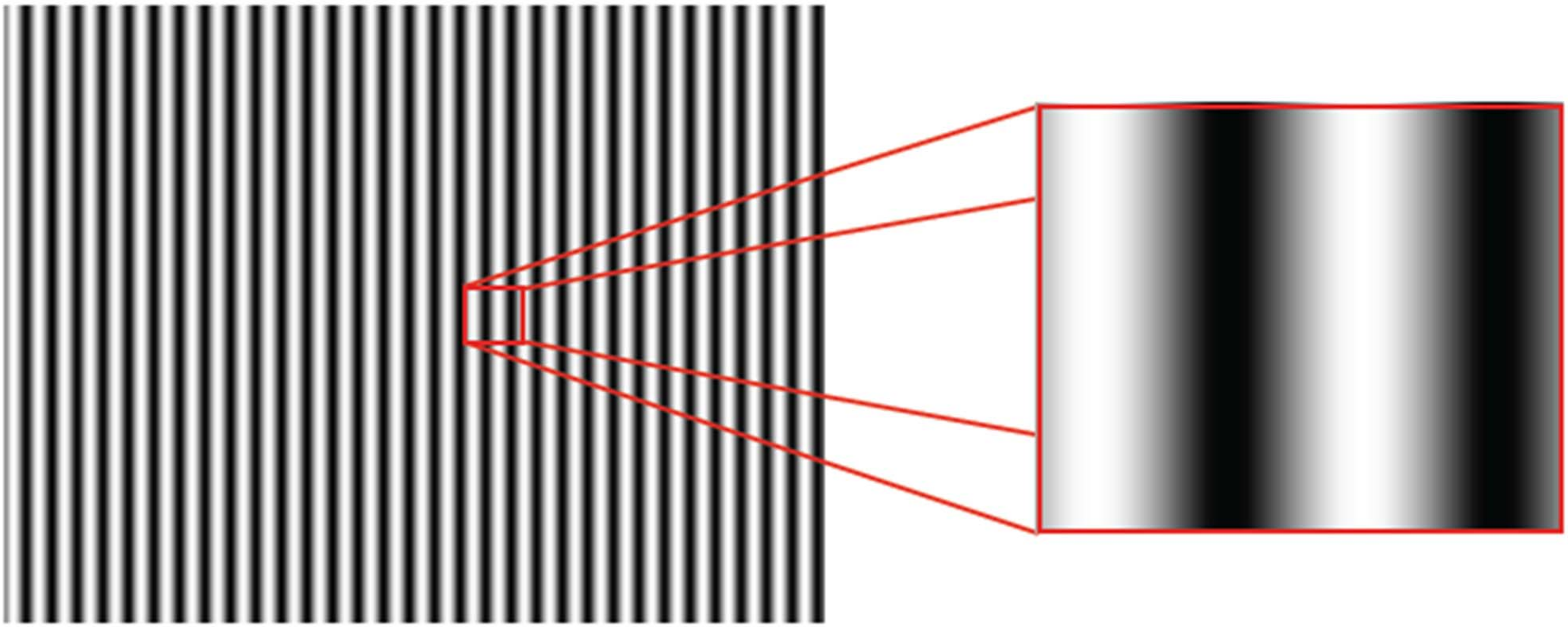


Figure 9.15: Sketch of the spatial arrangement of light patterns used to implement a simple phase shift method.

Phase-shift projection

Source:
http://www.igp.ethz.ch/photogrammetry/education/lehrveranstaltungen/MachineVisionFS2011/coursematerial/MV-SS2011-structured.pdf

- Increasing the resolution
  - project three phase-shifted sinusoidal patterns
  - can be projected sequentially, or simultaneously in different colours
  - the recorded intensities allow to compute the phase angle of a pixel within a wavelength



Phase-shift projection

- Phase angle from brightness values
  - computing the phase angle from the three images
  - although the method relies on brightness, the ambient light and the power of the projector need not be known

observed intensities

$$I_{-} = I_{base} + I_{var} \cos(\phi - \theta)$$
$$I_{0} = I_{base} + I_{var} \cos(\phi)$$
$$I_{+} = I_{base} + I_{var} \cos(\phi + \theta)$$

$$\frac{I_{-} - I_{+}}{2I_0 - I_{-} - I_{+}} =$$

Phase-shift projection

- Phase angle from brightness values
  - computing the phase angle from the three images
  - although the method relies on brightness, the ambient light and the power of the projector need not be known

observed intensities

$$I_- = I_{base} + I_{var}\cos(\phi - \theta)$$
$$I_0 = I_{base} + I_{var}\cos(\phi)$$
$$I_+ = I_{base} + I_{var}\cos(\phi + \theta)$$

➡ removed dependence on I_{base}

➡ removed dependence on I_{var}

$$\frac{I_- - I_+}{2I_0 - I_- - I_+} =$$
$$\frac{\cancel{I_{base}} + \cancel{I_{var}}\cos(\phi - \theta) - \cancel{I_{base}} - \cancel{I_{var}}\cos(\phi + \theta)}{2\cancel{I_{base}} + 2\cancel{I_{var}}\cos\phi - \cancel{I_{base}} - \cancel{I_{var}}\cos(\phi - \theta) - \cancel{I_{base}} - \cancel{I_{var}}\cos(\phi + \theta)}$$

Phase-shift projection

- Phase angle from brightness values
  - computing the phase angle from the three images
  - although the method relies on brightness, the ambient light and the power of the projector need not be known

observed intensities

$$I_- = I_{base} + I_{var}\cos(\phi - \theta)$$
$$I_0 = I_{base} + I_{var}\cos(\phi)$$
$$I_+ = I_{base} + I_{var}\cos(\phi + \theta)$$

from trigonometry

$$\tan\left(\frac{\theta}{2}\right) = \frac{1 - \cos(\theta)}{\sin(\theta)}$$
$$\cos(\phi - \theta) = \cos(\phi)\cos(\theta) + \sin(\phi)\sin(\theta)$$
$$\cos(\phi + \theta) = \cos(\phi)\cos(\theta) - \sin(\phi)\sin(\theta)$$

$$\frac{\cos(\phi - \theta) - \cos(\phi + \theta)}{2\cos\phi - \cos(\phi - \theta) - \cos(\phi + \theta)} = \frac{2\sin(\phi)\sin(\theta)}{2\cos(\phi)(1 - \cos(\theta))}$$

Phase-shift projection

- Phase angle from brightness values
  - computing the phase angle from the three images
  - although the method relies on brightness, the ambient light and the power of the projector need not be known

observed intensities

$$I_- = I_{base} + I_{var}\cos(\phi - \theta)$$
$$I_0 = I_{base} + I_{var}\cos(\phi)$$
$$I_+ = I_{base} + I_{var}\cos(\phi + \theta)$$

from trigonometry

$$\tan\left(\frac{\theta}{2}\right) = \frac{1 - \cos(\theta)}{\sin(\theta)}$$
$$\cos(\phi - \theta) = \cos(\phi)\cos(\theta) + \sin(\phi)\sin(\theta)$$
$$\cos(\phi + \theta) = \cos(\phi)\cos(\theta) - \sin(\phi)\sin(\theta)$$

$$\frac{2\sin(\phi)\sin(\theta)}{2\cos(\phi)(1 - \cos(\theta))} = \frac{\tan(\phi)\sin(\theta)}{1 - \cos(\theta)} = \frac{\tan(\phi)}{\tan(\theta/2)}$$

Phase-shift projection

- Phase angle from brightness values
  - computing the phase angle from the three images
  - although the method relies on brightness, the ambient light and the power of the projector need not be known

observed intensities

$$I_- = I_{base} + I_{var} \cos(\phi - \theta)$$
$$I_0 = I_{base} + I_{var} \cos(\phi)$$
$$I_+ = I_{base} + I_{var} \cos(\phi + \theta)$$

phase angle

$$\frac{I_- - I_+}{2I_0 - I_- - I_+} = \frac{\tan(\phi)}{\tan(\theta/2)}$$

$$\phi'(0, 2\pi) = \arctan\left(\tan\left(\frac{\theta}{2}\right)\frac{I_- - I_+}{2I_0 - I_- - I_+}\right)$$

Phase-shift projection

- Total phase
  - phase angle within a period from intensity
  - number of period from stereo triangulation (or light stripe)
  - stereo matching is easy: only N possibilities

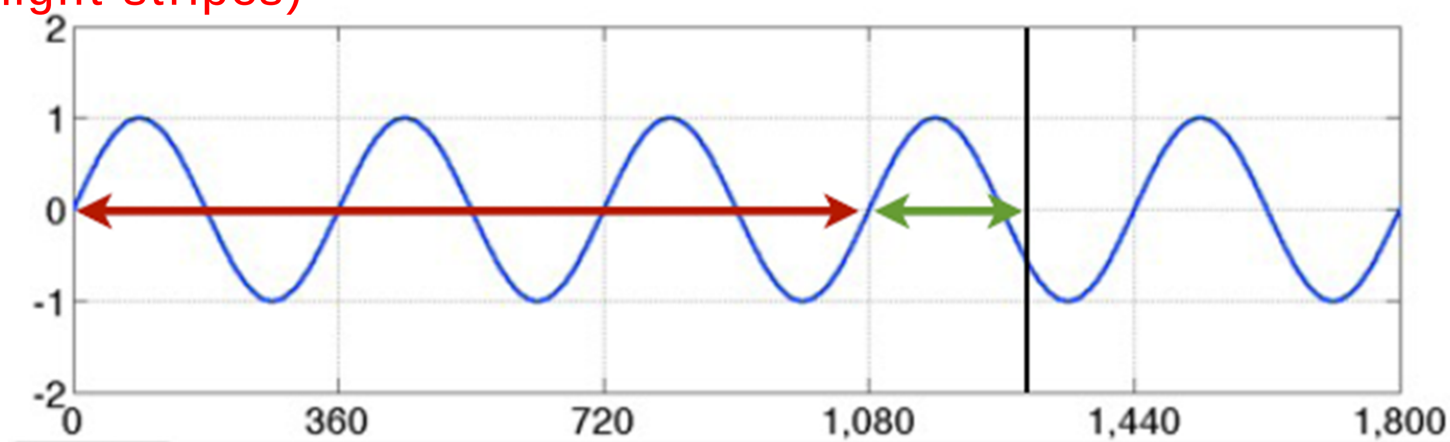
absolute phase

$$\phi(x,y) = 2\pi\, k(x,y) + \phi'(x,y) \qquad k \in [0 \ldots N-1]$$

from stereo (or light stripes)

from phase-shift

number of periods (stripes)

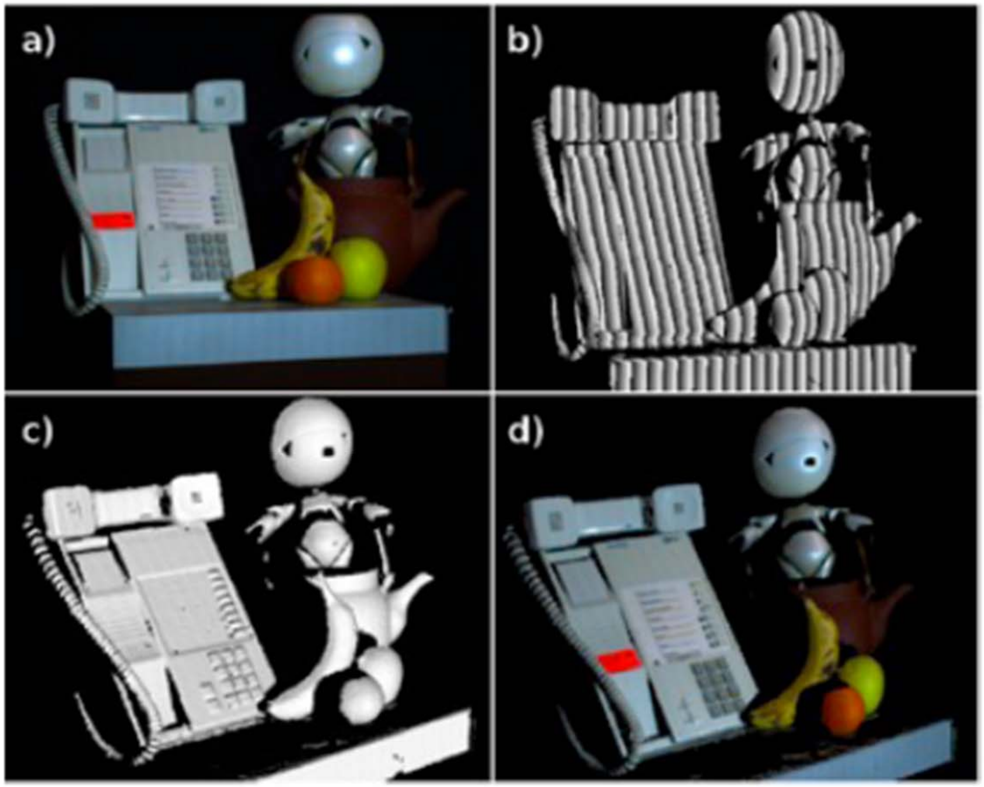


Phase-shift projection

- Total phase

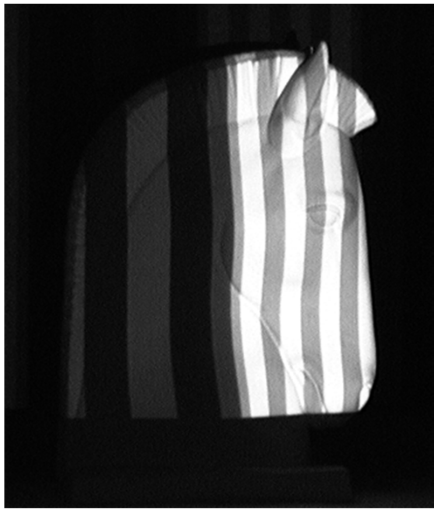
See also: http://www.youtube.com/watch?v=a6pgzNUjh_s

  - the phase angle only determines the relative position within one cycle of the periodic sine wave
  - need to know which stripe we are in (c.f. GPS phase ambiguity)
  - achieved by ordering assumption, or combination with stereo



N-ary codes

- Reduce the number of patterns by increasing the number of intensity levels used to encode the stripes.
  - Multi grey levels instead of binary
  - Multilevel gray code based on color.
- Alphabet of *m* symbols encodes m^n stripes



3 patterns based on a n-ary code
of 4 grey levels (Horn & Kiryati)
→ 64 encoded stripes

Direct Codification

- Every encoded pixel is identified by its own intensity/color
- Since the codification is usually condensed in a unique pattern, the **spectrum** of intensities/colors used is very large
- Additional **reference patterns** must be projected in order to differentiate among all the projected intensities/colors:
    - Ambient lighting (black pattern)
    - Full illuminated (white pattern)
    - ...

Introducing color in coding

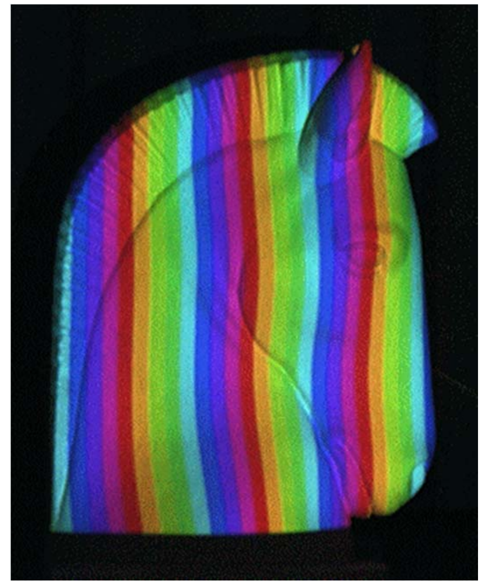
- Allowing colors in coding is the same as augmenting code basis. This gives us more words with the same length.
- If the scene changes the color of projected light, then information can be lost.
- Reflectivity restrictions (neutral scene colors) have to be imposed to guarantee the correct decoding.

Direct encoding with color

- Every encoded point of the pattern is identified by its colour

Tajima and Iwakawa rainbow pattern

(the rainbow is generated with a source of white light passing through a crystal prism)

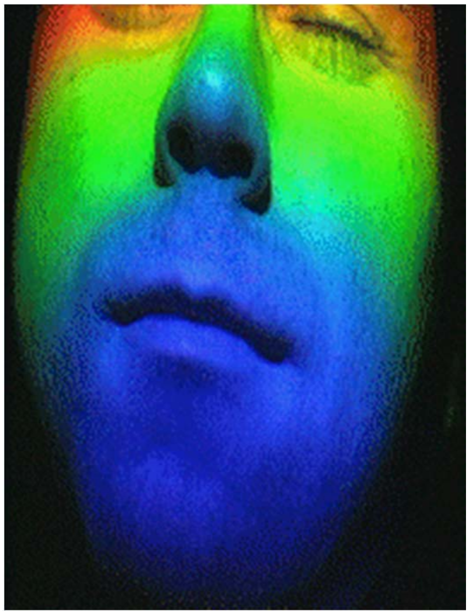


T. Sato patterns capable of cancelling the object colour by projecting three shifted patterns

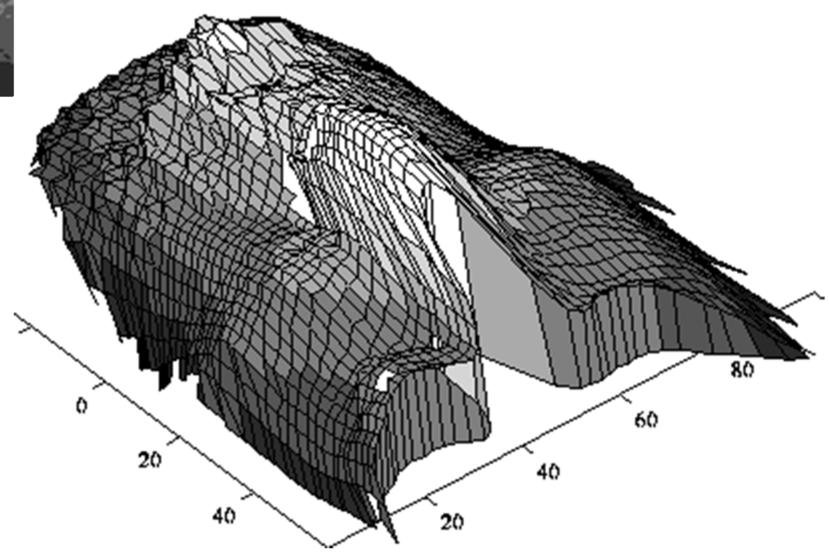
(it can be implemented with an LCD projector if few colours are projected)

Rainbow Pattern

http://cmp.felk.cvut.cz/cmp/demos/RangeAcquisition.html



Assumes that the scene does not change the color of projected light



Direct Codification

- Every encoded pixel is identified by its own intensity/color
- Since the codification is usually condensed in a unique pattern, the **spectrum** of intensities/colors used is very large
- Additional **reference patterns** must be projected in order to differentiate among all the projected intensities/colors:
    - Ambient lighting (black pattern)
    - Full illuminated (white pattern)
    - ...

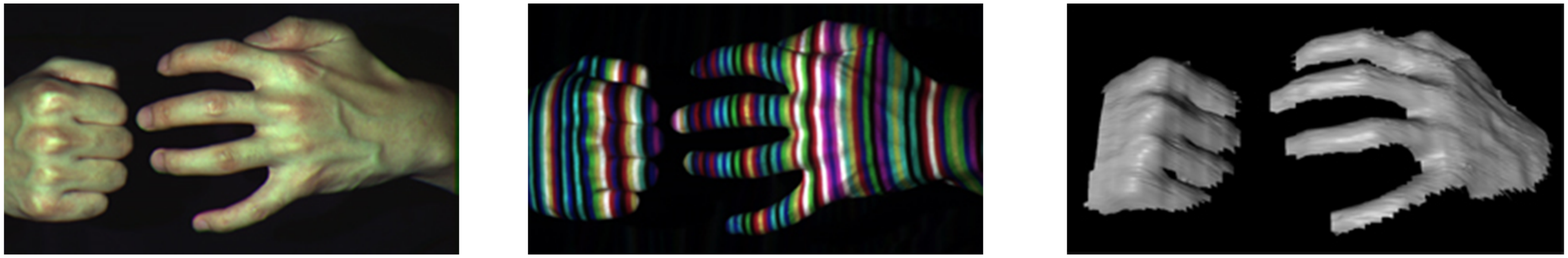
- **Advantages**:
    - Reduced number of patterns
    - High resolution can be theoretically achieved

- **Drawbacks**:
    - Very noisy in front of reflective properties of the objects, non-linearities in the camera spectral response and projector spectrum ⇒ non-standard light emitters are required in order to project **single wave-lengths**
    - Low accuracy (order of 1 mm)

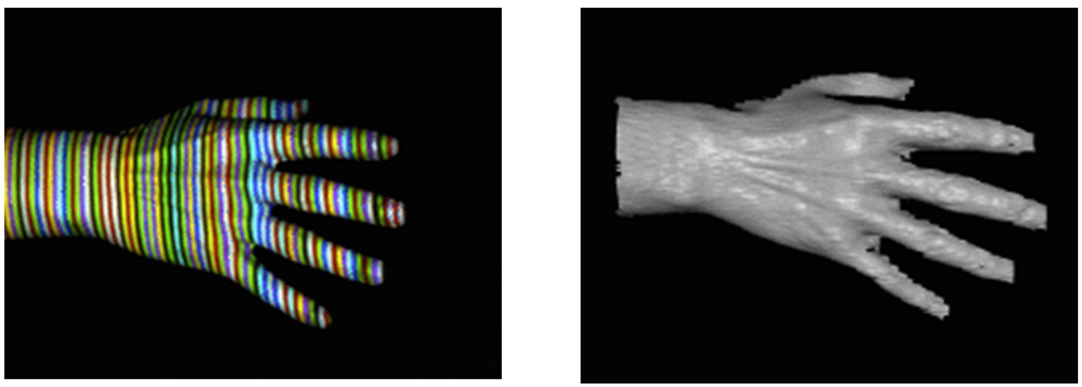
Spatial Coherence

- Coding in a single frame.
- Spatial Coherence can be local or global.
- The minimum number of pixels used to identify the projected code defines the accuracy of details to be recovered in the scene.

Real time by direct encoding



Works despite complex appearances



Works in real-time and on dynamic scenes

- Need very few images (one or two).
- But needs a more complex correspondence algorithm

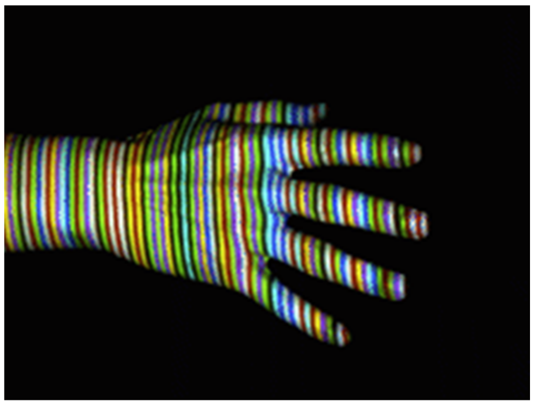
Zhang et al

De Bruijn Sequences

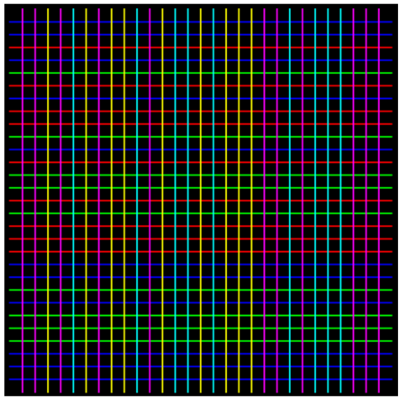
- A De Bruijn sequence (or pseudorandom sequence) of order *m* over an alphabet of *n* symbols is a circular string of length n^m that contains every substring of length *m* exactly once (in this case the windows are one-dimensional).

1000010111101001 { m=4 (window size) ; n=2 (alphabet symbols) }

- The De Bruijn sequences are used to define colored slit patterns (single axis codification) or grid patterns (double axis codification)
- In order to decode a certain slit it is only necessary to identify one of the windows in which it belongs to) can **resolve occlusion problem.**



Zhang et al.: 125 slits encoded with a De Bruijn sequence of 8 colors and window size of 3 slits



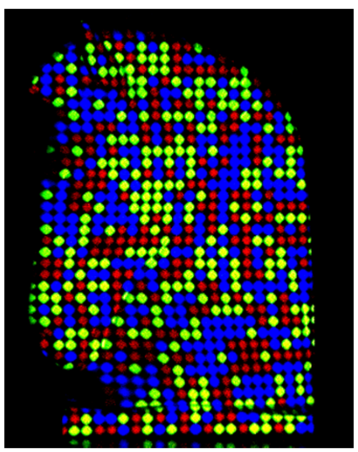
Salvi et al.: grid of 29×29 where a De Bruijn sequence of 3 colors and window size of 3 slits is used to encode the vertical and horizontal slits

M-Arrays

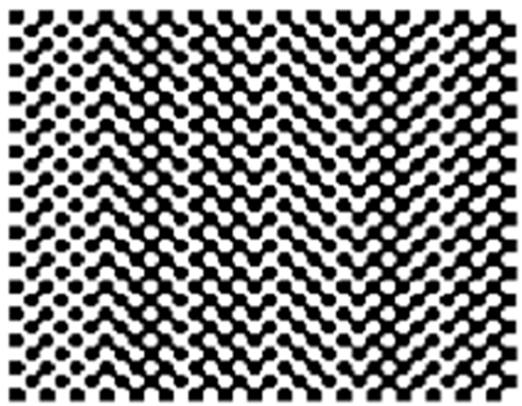
- An m-array is the bidimensional extension of a De Bruijn sequence. Every window of w×h units appears only once. The window size is related with the size of the m-array and the number of symbols used

0	0	1	0	1	0
0	1	0	1	1	0
1	1	0	0	1	1
0	0	1	0	1	0

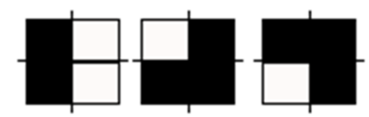
Example: binary m-array of size 4×6 and window size of 2×2



Morano et al. M-array represented with an array of coloured dots

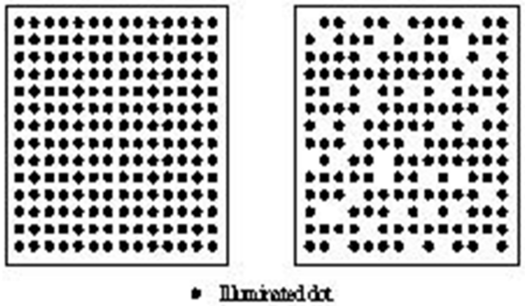


M-array proposed by Vuylsteke et al. Represented with shape primitives



Shape primitives used to represent every symbol of the alphabet

Some examples

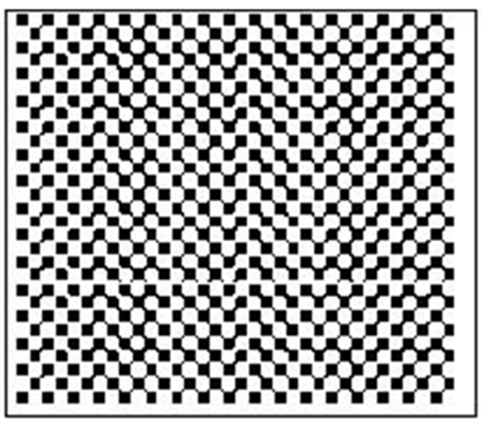


Morita - Yakima - Sakata 1988

- Initial projection of a whole illuminated dot matrix to extract dot position.
- Window coded pattern.

Column Coded / Static / Binary / Absolute

Lavoie '96. A grid pattern with random binary dots in the cross-points.



Vuylsteke - Oosterlinck 1990

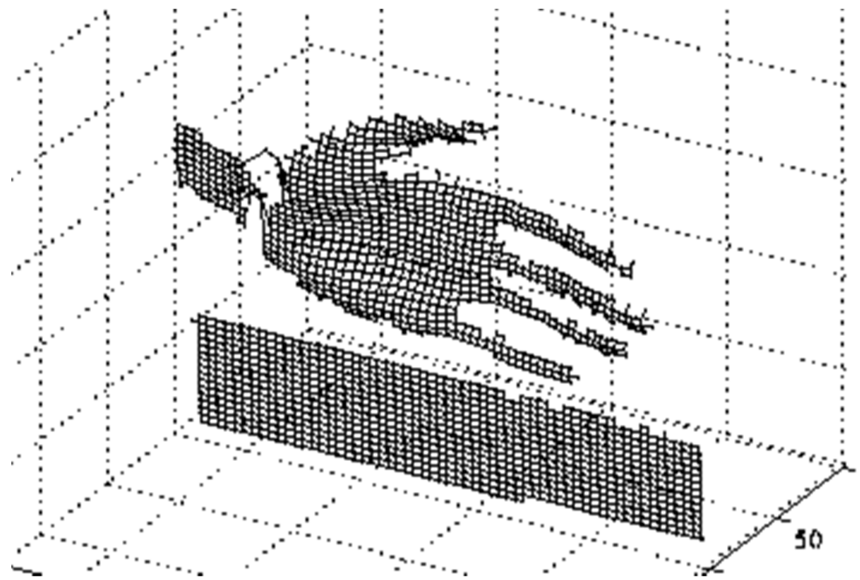
- Chess-board pattern projection with coded squares.
- Window coded pattern.

Column Coded / Dynamic / Binary / Absolute

Pajdla '95 : Re-implementation.

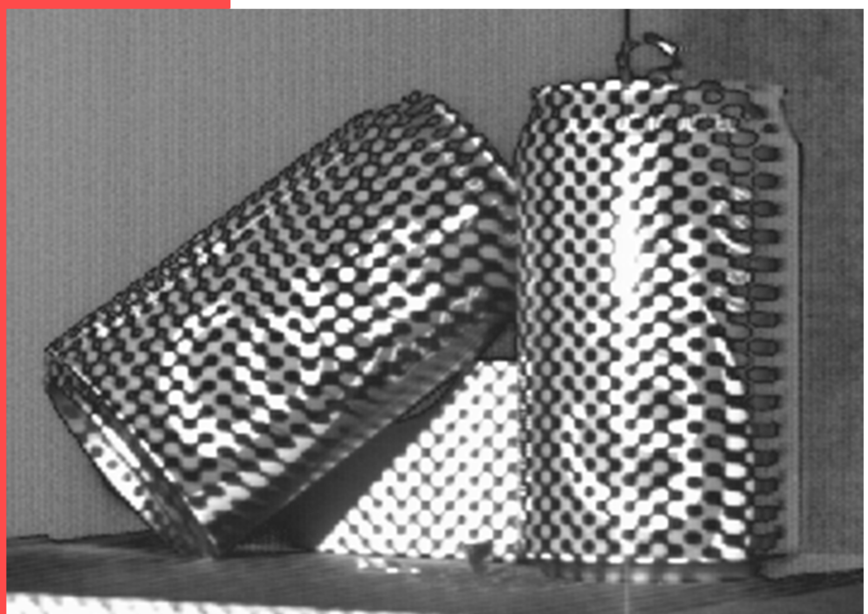
Ito '95 : A three grey level checkerboard pattern.

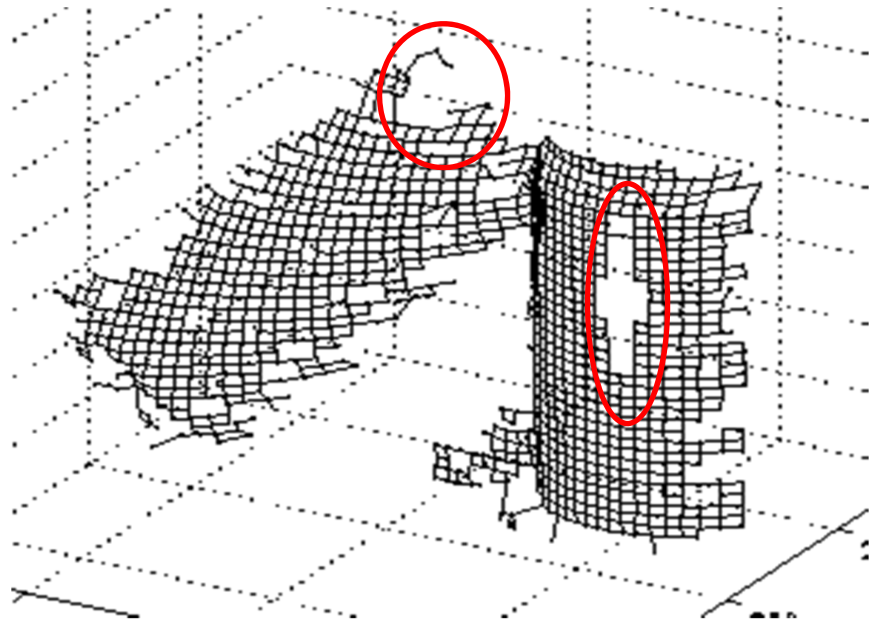
Binary spatial coding



http://cmp.felk.cvut.cz/cmp/demos/RangeAcquisition.html

Problems in recovering pattern





Examples

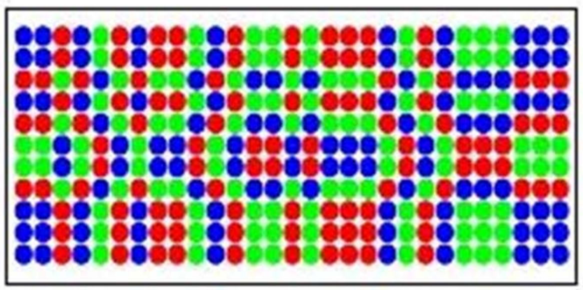
Boyer - Kak 1987

- Multiple coloured vertical slits.
- Codification from slit colour sequence.

Column Coded / Dynamic / Colour / Absolute

Monks '93 : Utilisation of the same pattern for speech interpretation.

Chen '97 : Unique codification and colour improvement.



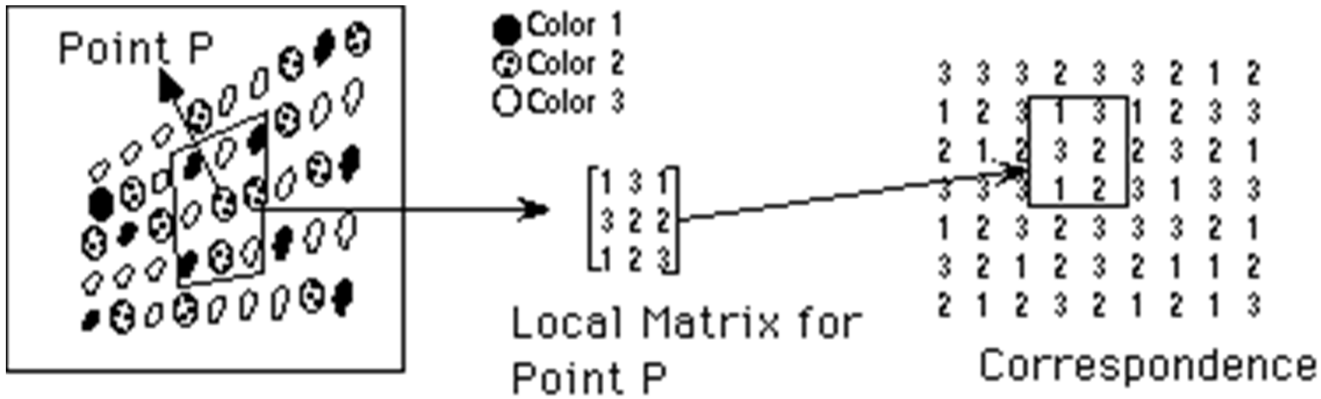
Griffin - Narasimhan - Yee 1992

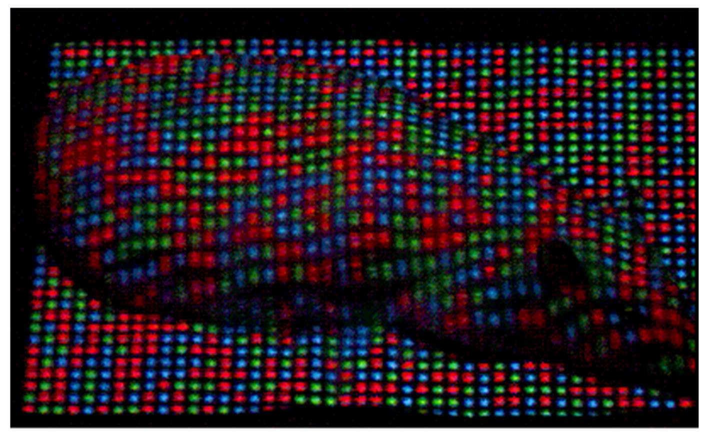
- Mathematical study to obtain the largest codification matrix from a fixed number of colours.
- Dot position coded by the colour of its four neighbours.

Both axis coded / Static / Colour / Absolute

Davies '96 : Re-implementation.

Local spatial Coherence





http://www.mri.jhu.edu/~cozturk/sl.html

•Medical Imaging Laboratory
Departments of Biomedical Engineering and Radiology
Johns Hopkins University School of Medicine
Baltimore, MD 21205

Experimental results

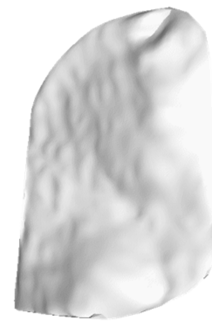
s) 4 stripes)

De Bruijn

Gühring

Time-
multiplexing

Morano (45x45 dot array)



Spat
codific

Direct
codification

Discussion Structured Light

- Advantages
  - robust - solves the correspondence problem
  - fast - instantaneous recording, real-time processing
- Limitations
  - less flexible than passive sensing: needs specialised
  - equipment and suitable environment
- Applications
  - industrial inspection
  - entertainment
  - healthcare
  - heritage documentation
  - .....

www.ipg.ethz.ch/photogrammetry

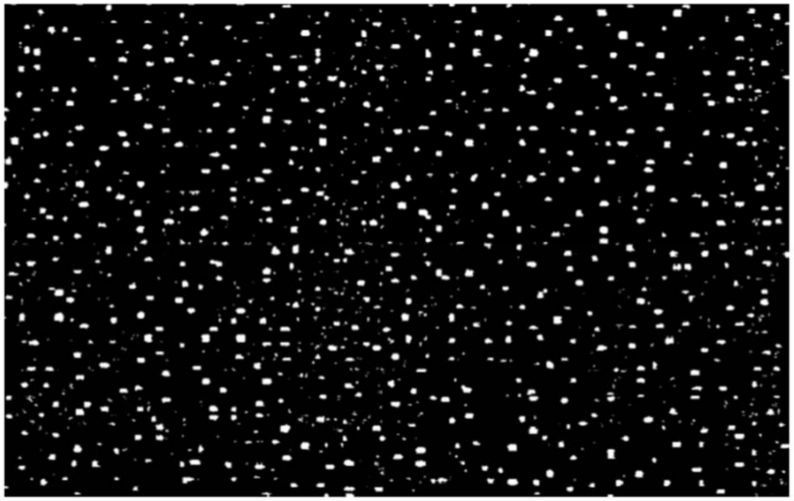
Microsoft Kinect

Microsoft Kinect

The Kinect combines structured light with two classic computer vision techniques: depth from focus, and depth from stereo.

Stage 1: The depth map is constructed by analyzing a speckle pattern of infrared laser light

The Kinect uses infrared laser light, with a speckle pattern

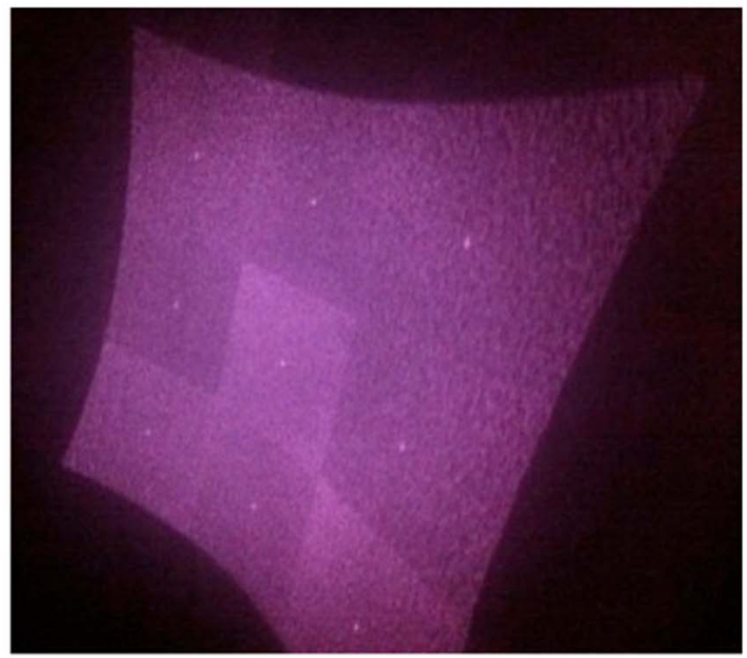


Shpunt et al, PrimeSense patent application US 2008/0106746

http://users.dickinson.edu/~jmac/selected-talks/kinect.pdf

Consumer application

- Now people have it in their living room
  - Xbox Kinect - periodic infrared dot pattern

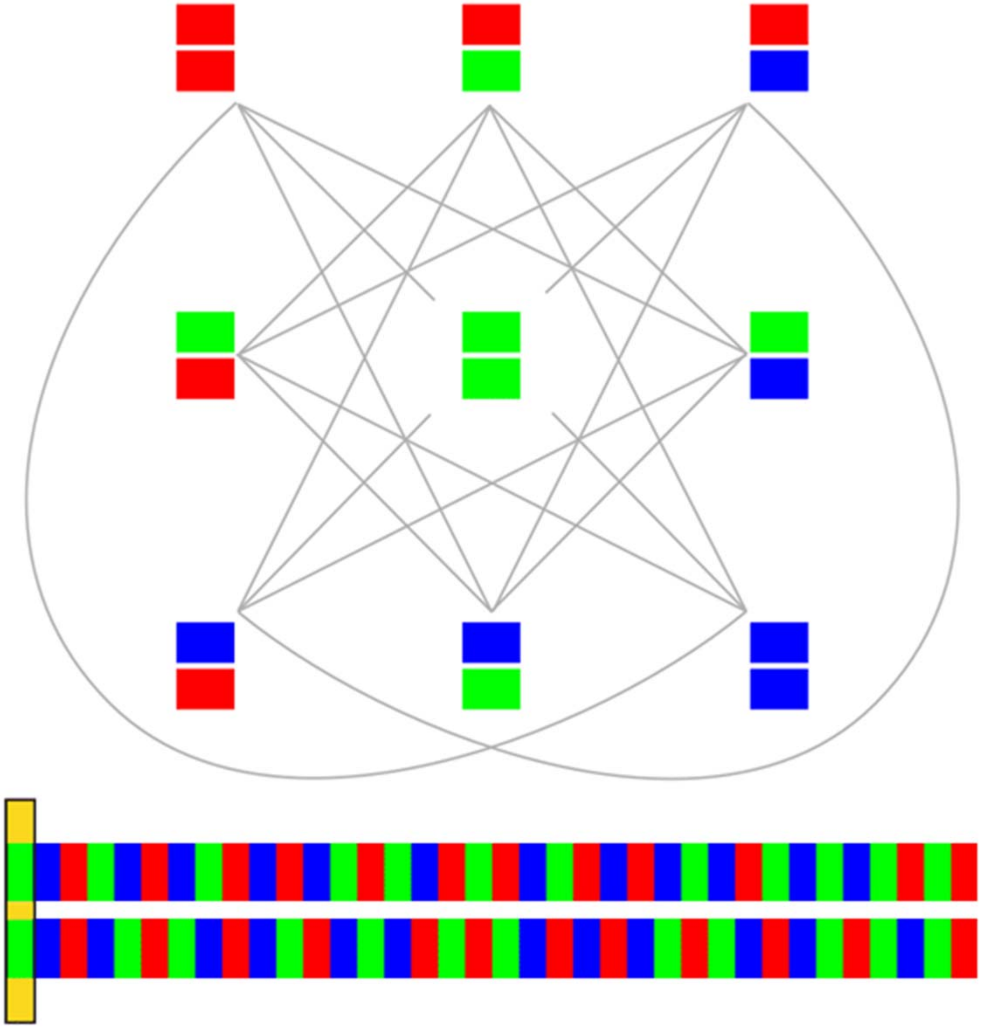


Microsoft Kinect

Inferring body position is a two-stage process: first compute a depth map, then infer body position

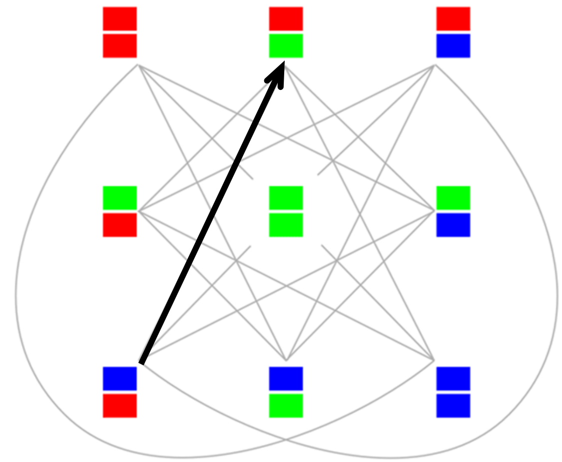
http://users.dickinson.edu/~jmac/selected-talks/kinect.pdf

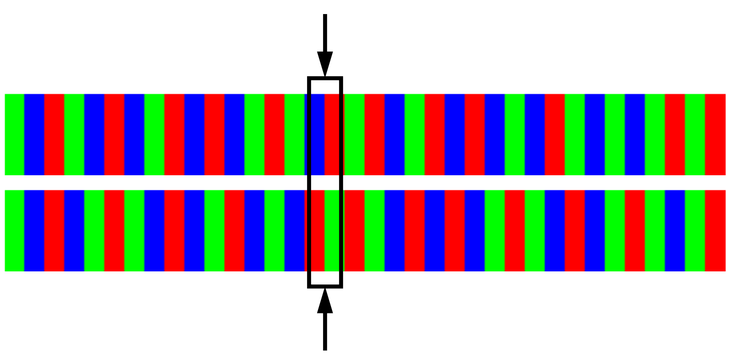
Decoding table



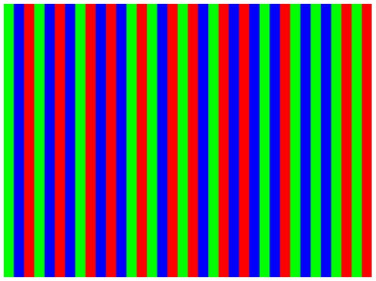
Decoding table

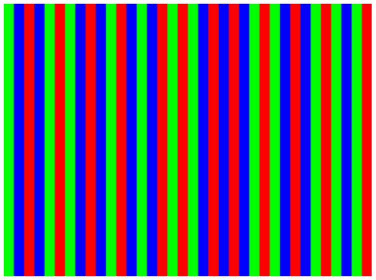
vertices	d(0)	d(1)	d(2)	d(3)
V(00)	0	3	6	9
V(01)	14	17	19	11
V(02)	28	34	22	24
V(10)	26	29	18	21
V(11)	1	31	33	35
V(12)	15	4	8	13
V(20)	16	23	32	12
V(21)	27	5	7	25
V(22)	2	10	20	30

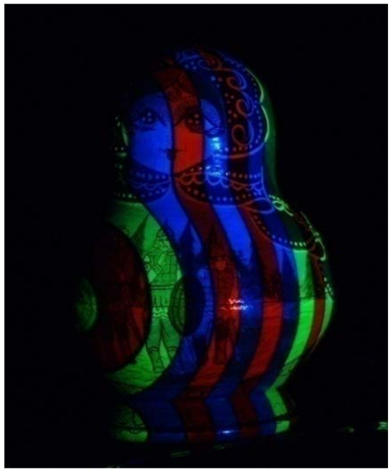




Experiment







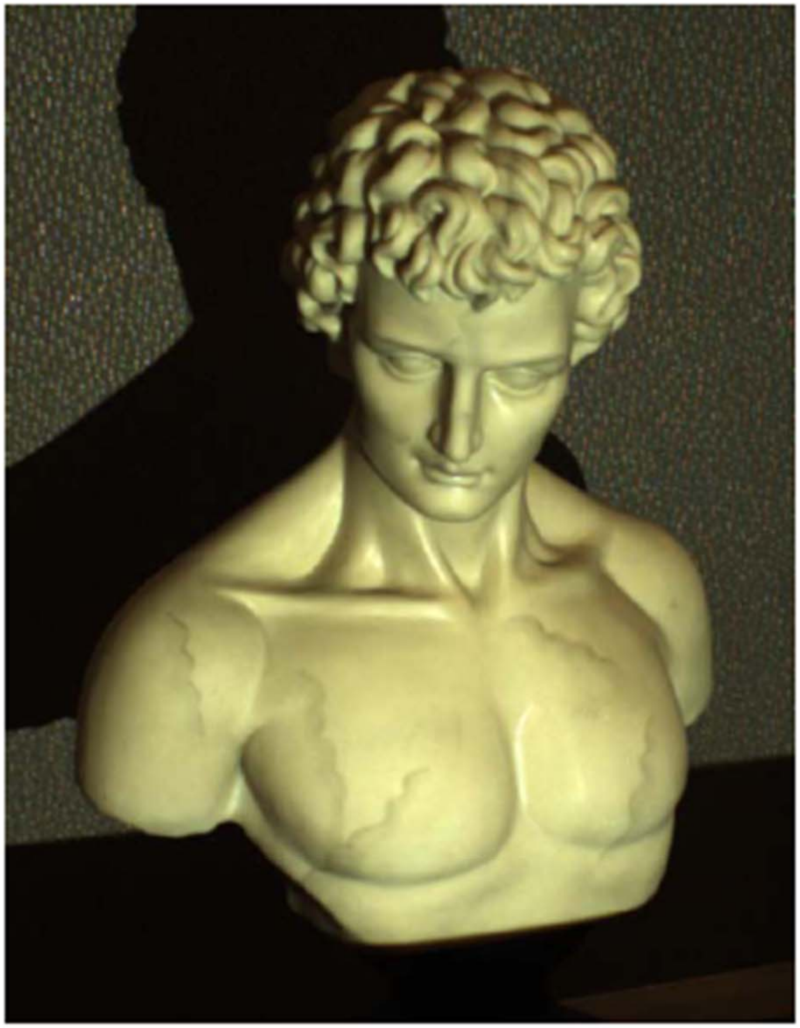
Spatial Codification

- Project a certain kind of spatial pattern so that a set of neighborhood points appears in the pattern only once. Then the codeword that labels a certain point of the pattern is obtained from a neighborhood of the point around it.
- The codification is condensed in a unique pattern instead of multiplexing it along time
- The size of the neighborhood (**window size**) is proportional to the number of encoded points and inversely proportional to the number of used colors
- The aim of these techniques is to obtain a **one-shot measurement** system ⇒ moving objects can be measured

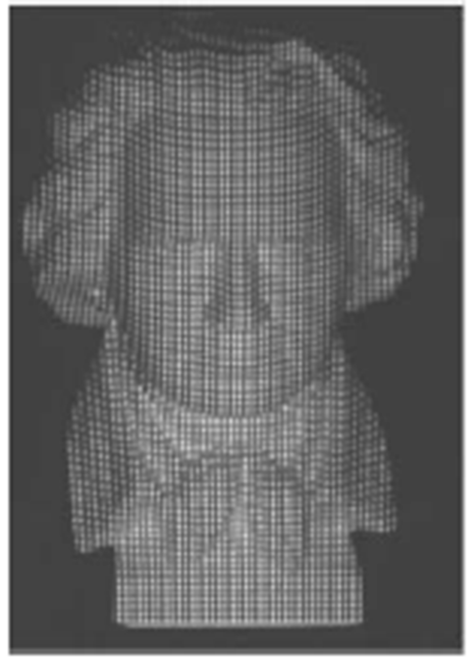
- **Advantages**:
  - Moving objects supported
  - Possibility to condense the codification to a unique pattern

- **Drawbacks**:
  - Discontinuities on the object surface can produce erroneous window decoding (occlusions problem)
  - The higher the number of used colours, the more difficult to correctly identify them when measuring non-neutral surfaces
  - Maximum resolution cannot be reached

Gray Code Structured Lighting: Results



Results



igure 5: Beethoven bust with projected dot pattern

Including exposures with phase-shifted pattern the complete surface dataset consists of 43,000 projected dots. Results of a subset of about 18,000 dots are shown in Figure 6 - Figure 8. Figure 8 shows a photo-realistic visualization of the dataset, which has been generated from the photogrammetrically determined object surface data by a raytracer program.

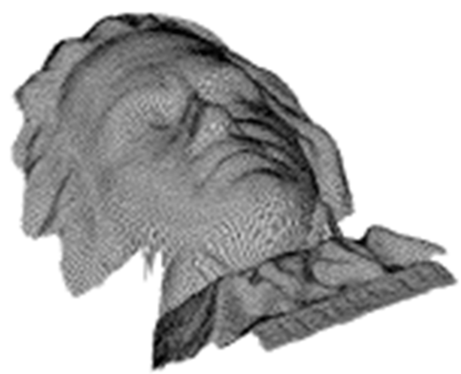


Figure 6: Beethoven - grid Model

Figure 7: Beethoven - Isoline plot

Figure 8: Beethoven - Photorealistic visualisation

Conclusions

Types of techniques		
Time-multiplexing	• Highest resolution • High accuracy • Easy implementation	• Inapplicability to moving objects • Large number of patterns
Spatial codification	• Can measure moving objects • A unique pattern is required	• Lower resolution than time-multiplexing • More complex decoding stage • Occlusions problem
Direct codification	• High resolution • Few patterns	• Very sensitive to image noise • Inapplicability to moving objects

Guidelines

Requirements	Best technique
• High accuracy • Highest resolution • Static objects • No matter the number of patterns	Phase shift + Gray code → Gühring's line-shift technique
• High accuracy • High resolution • Static objects • Minimum number of patterns	N-ary pattern → Horn & Kiryati Caspi et al.
• High accuracy • Good resolution • Moving objects	De Bruijn pattern → Zhang et al.