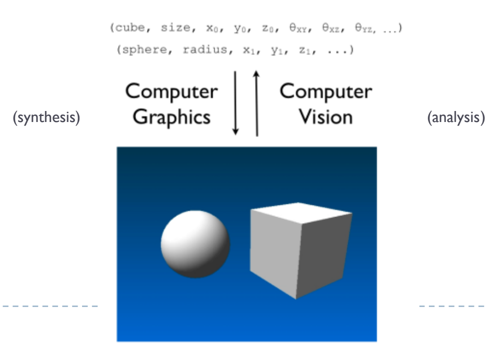
Computer Vision
for Computer Graphics

Mark Borg

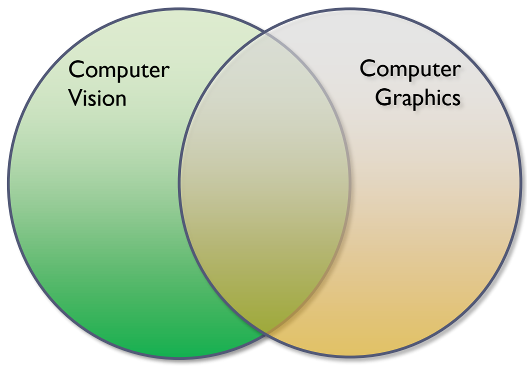
Computer Vision & Computer Graphics I

- Computer Vision
  - Understanding the "content" of an image (normaly by creating a "model" of the observed scene)
- Computer Graphics
  - Creating an image using a computer "model"



Computer Vision & Computer Graphics II

- Recent confluence between computer vision (CV) and computer graphics (CG)



Outline

- We will look at the following CV areas:
  - Stereovision
    - Recovering depth information
    - Stereo correspondence problem
    - Multi-view imaging and the Plenoptic function
    - Applications to CG:
      - 3D Model Acquisition
      - View Morphing, "bullet time" effect
  - Automated Visual Surveillance
    - Motion Detection
    - Background Subtraction techniques
    - Object Tracking
    - Applications to CG:
      - Motion Capture
      - Basis for Behaviour Recognition in HCI interfaces, Project Natal

Stereo Vision

- "Stereo Vision" generally means two synchronised cameras or eyes capturing images
- Allows recovery of depth information / sensation of depth
- (stereo vision = stereoscopic vision = stereopsis)

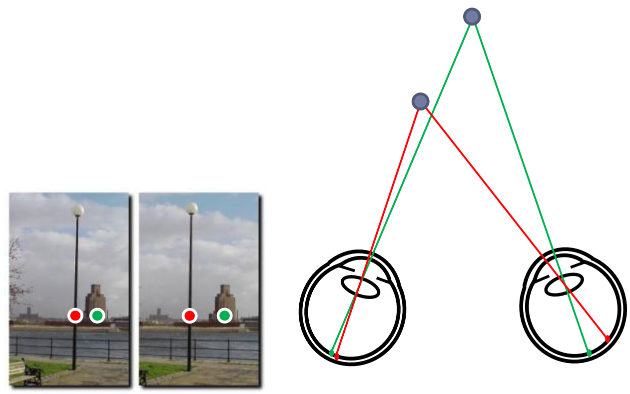
Parallax effect

- Each eye has a slightly different view of the world

- Nearby objects have a larger *parallax* than more distant objects

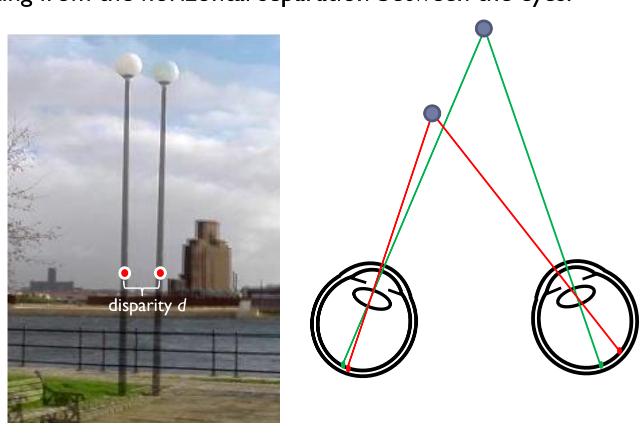
Depth from Binocular Disparity

- Binocular disparity:
  - The difference in image location of an object seen by the left and right eyes, resulting from the horizontal separation between the eyes.

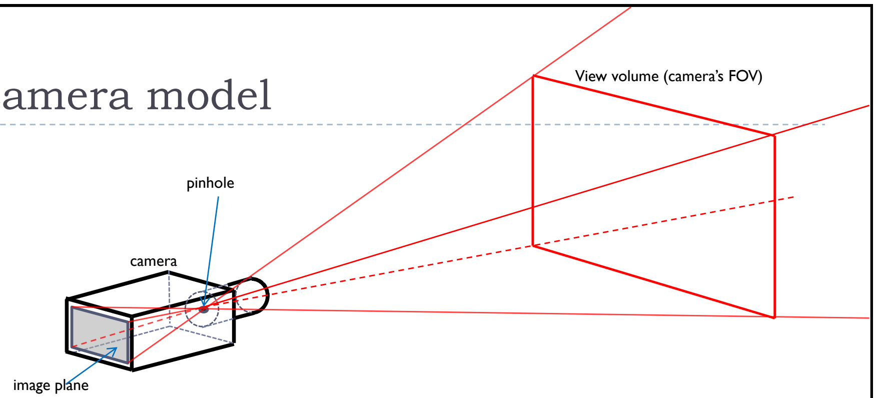


Depth from Binocular Disparity II

- Binocular disparity:
  - The difference in image location of an object seen by the left and right eyes, resulting from the horizontal separation between the eyes.

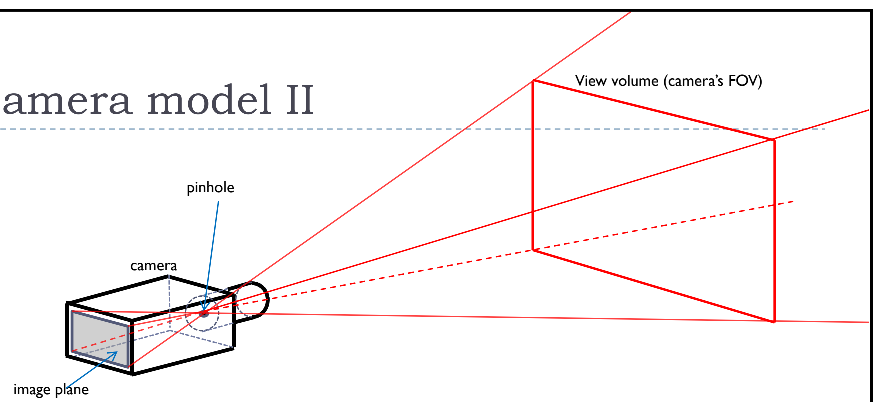


Camera model



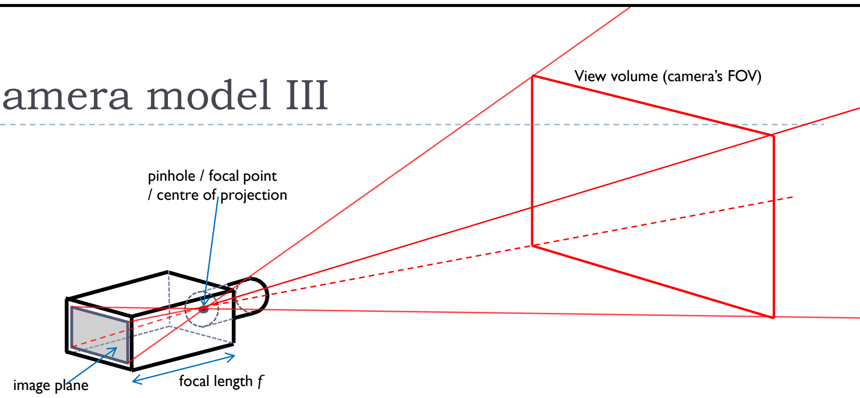
- 'Pinhole' camera model
  - Simplified camera model; most often used in CV.
  - No lens; just a single very small aperture (infinitely small).
  - Models perspective projection.
  - Assumes all light rays pass through a single point (the 'pinhole').
  - Assumes the View Volume (camera's FOV) is an infinite pyramid.

Camera model II



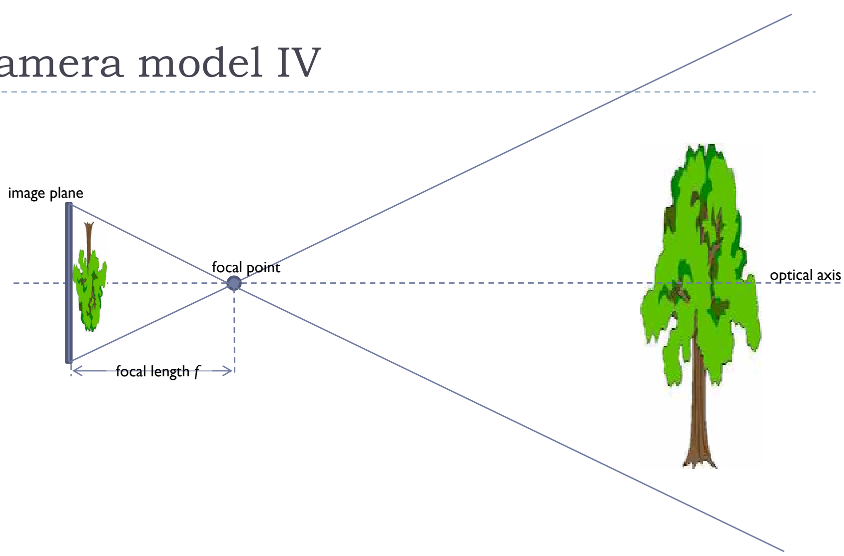
- Limitations of the pinhole camera model:
  - Depth-of-field effects and light attenuation ignored in this model.
  - View volume is not really infinite.

Camera model III

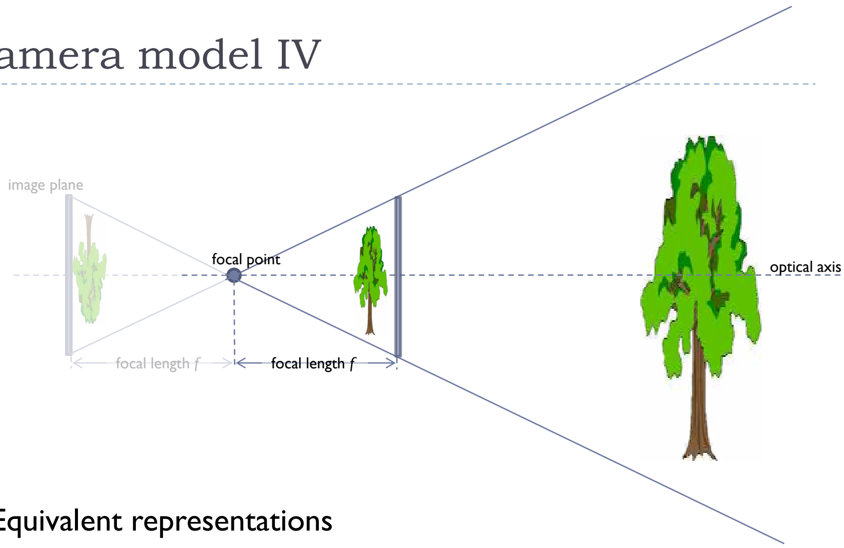


- Single Viewpoint Constraint
  - Image plane is situated at a distance f (the *focal length*) from the pinhole.
  - The pinhole is also called the *focal point*, or *centre of projection*, or *lens centre*.
  - A camera with a single centre of projection is called a *central projection camera* (*central camera* for short) and obey the *single viewpoint constraint*.
  - Such a camera 'sees' the world from a single point (has a single viewpoint).

Camera model IV



Camera model IV

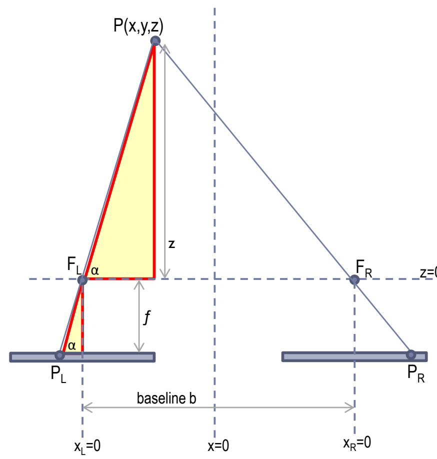


- Equivalent representations

Stereo vision cameras

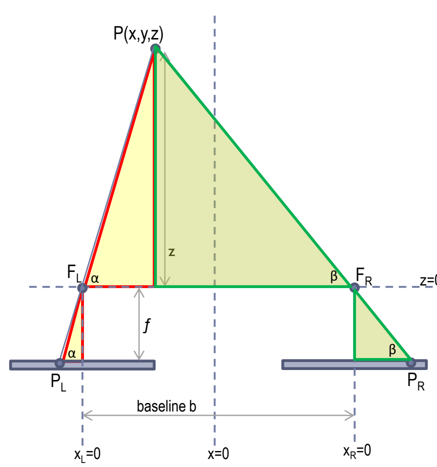
Depth from Binocular Disparity III

- Ratio of sides of similar triangles:
- $\frac{P_L}{f} = \frac{-\frac{1}{2}b + x}{Z}$



Depth from Binocular Disparity III

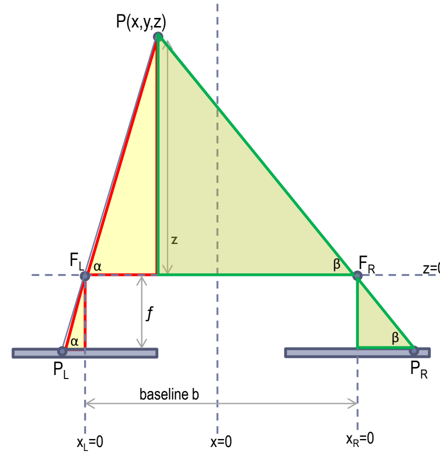
- Ratio of sides of similar triangles:
- $\frac{P_L}{f} = \frac{-\frac{1}{2}b + x}{Z}$
- $\frac{P_R}{f} = \frac{\frac{1}{2}b + x}{Z}$



Depth from Binocular Disparity III

- Ratio of sides of similar triangles:
- $\frac{P_L}{f} = \frac{-\frac{1}{2}b + x}{z}$
- $\frac{P_R}{f} = \frac{\frac{1}{2}b + x}{z}$
- $z(P_R - P_L) = bf$
- $z = \frac{bf}{P_R - P_L}$

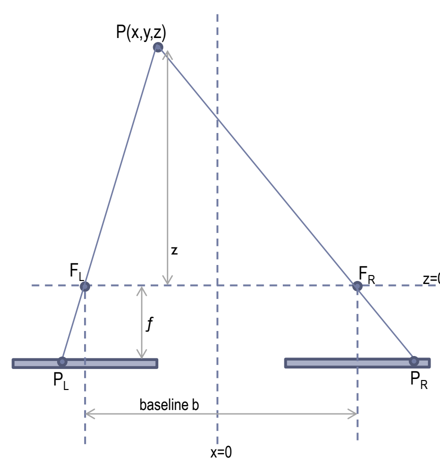
  where $(P_R - P_L)$ is the disparity



Depth from Binocular Disparity IV

- $z = \frac{bf}{P_R - P_L}$

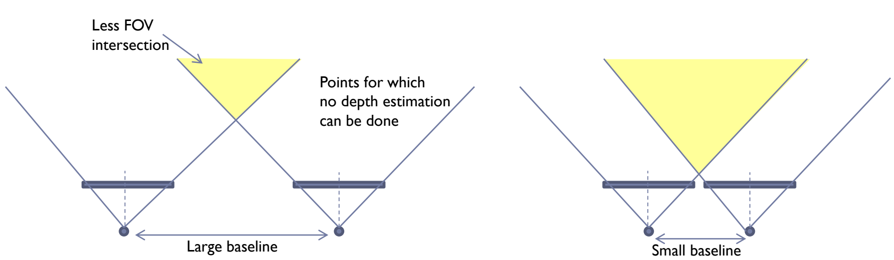
  where $(P_R - P_L)$ is the disparity
- $z \propto \frac{1}{disparity}$
- As disparity $(P_R - P_L) \rightarrow 0$, depth z $\rightarrow \infty$
- In reality, depth resolution determined by minimum disparity



Choosing the stereo baseline

- What's the optimal baseline?
  - Small baseline:
    - Smaller disparities → larger depth error; limited depth range.

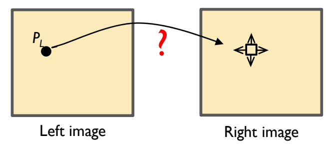
Choosing the stereo baseline



- What's the optimal baseline?
  - Small baseline:
    - Smaller disparities → larger depth error; limited depth range.
  - Large baseline:
    - Less FOV intersection → less scene points for which depth can be measured; more difficult search problem.
- Examples:
  - Human binocular vision: baseline = ~6cm; depth estimation = up to ~10m.
  - Stellar parallax estimation: baseline = Earth's orbital diameter; depth estimation = up to ~100 LY.

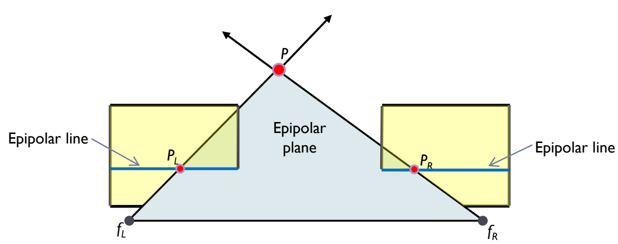
Correspondence Problem

- Basic stereo algorithm:
  - For a pixel P_L in the left image
    - Find the matching pixel P_R in the right image
    - Compute the depth value $z = \frac{bf}{\text{disparity}}$, where disparity = $(P_R - P_L)$
- Matching the points is a hard problem



- Called the (Stereo) Correspondence Problem
- Can reduce the search problem from 2D to 1D by using constraints from *epipolar geometry*

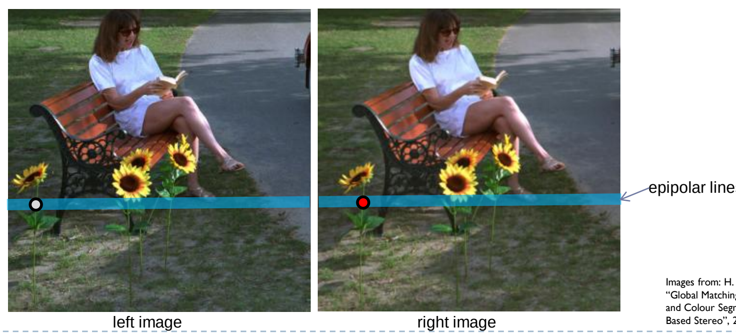
Epipolar Geometry



- Epipolar plane
  - Scene point P + image points P_L, P_R + camera focal points
- Epipolar line
  - Intersection of epipolar plane with image plane of the camera
- Epipolar Constraint
  - Given a point P_L, the corresponding point P_R will always occur along the *conjugate* epipolar line.
  - Reduces the search problem to a 1D search along conjugate epipolar lines.
  - For the camera configuration above, the conjugate epipolar lines correspond to the same image rows.

Basic Stereo Algorithm I

- For each epipolar line:
  - For each pixel P_L in the left image
    - Compare with every pixel on the same epipolar line in the right image
    - Pick the pixel P_R with the best match score/minimum cost.
  - Compute the depth value $z = \frac{bf}{\text{disparity}}$, where disparity = $(P_R - P_L)$

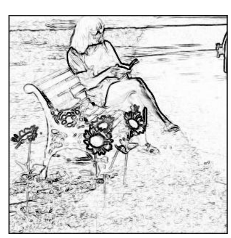


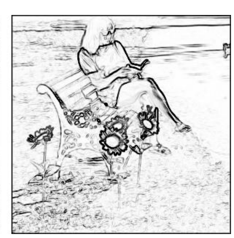
Images from: H. Tao et al., "Global Matching Criterion and Colour Segmentation Based Stereo", 2000.

Basic Stereo Algorithm II

- Matching of pixels:
  - A pixel must be quite distinct from its neighbours (else all neighbouring pixels will be good matches)
  - Therefore, must locate matchable features

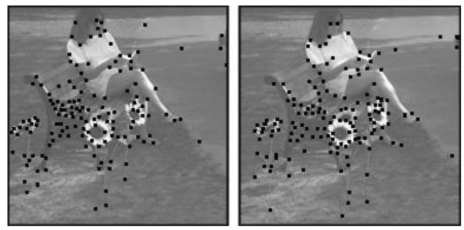
- Examples:
  - Using edge information





Basic Stereo Algorithm II

- Matching of pixels:
  - A pixel must be quite distinct from its neighbours (else all neighbouring pixels will be good matches)
  - Therefore, must locate matchable features
- Examples:
  - Using edge information
  - Using corner features
  - Using region correlation



- End result:
  - Normally a subset of pixels/features are selected for matching and depth computation.
  - Depth at other points can be estimated via interpolation techniques.

Basic Stereo Algorithm III

- 3D point cloud
  - Can be sparse or dense, depending on the features used for matching
  - Can be transformed into a surface model
    - Via depth or shape interpolation techniques
    - Mesh fitting
    - One common method: Dalaunay Triangulation

Some Results I


left image


right image


depth map

Images from: H. Tao et al., “Global Matching Criterion and Colour Segmentation Based Stereo”, 2000.

Some Results II

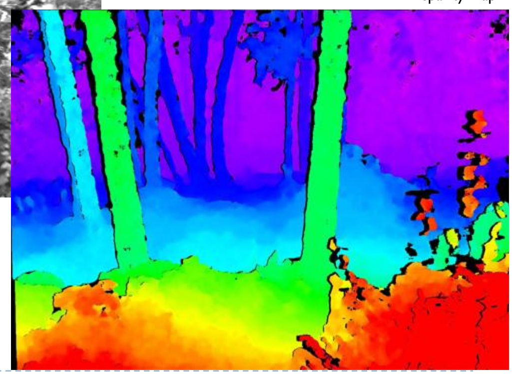

depth map


rendered view

Note the ‘holes’ caused by scene occlusions. These scene points are hidden from both cameras.

Some Results III

Disparity map

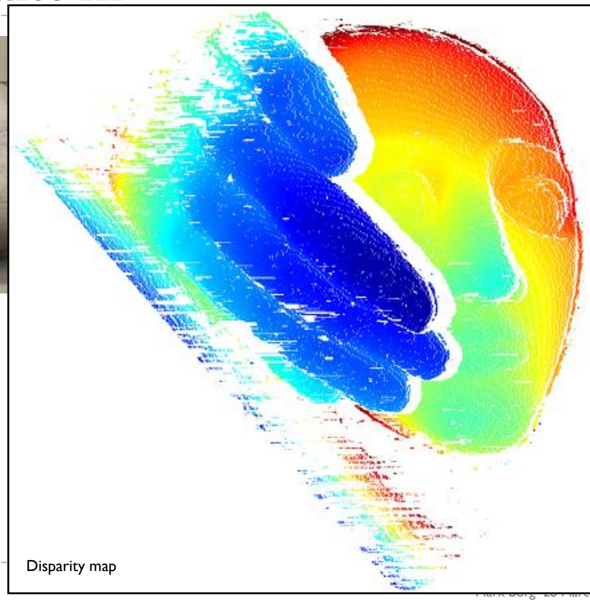


Stereo vision for robot navigation.
Source: Jet Propulsion Laboratory, NASA.

Some Results III

Source: Project IS-3D, Centre for Machine Perception, Czech Academy of Sciences, 2008.

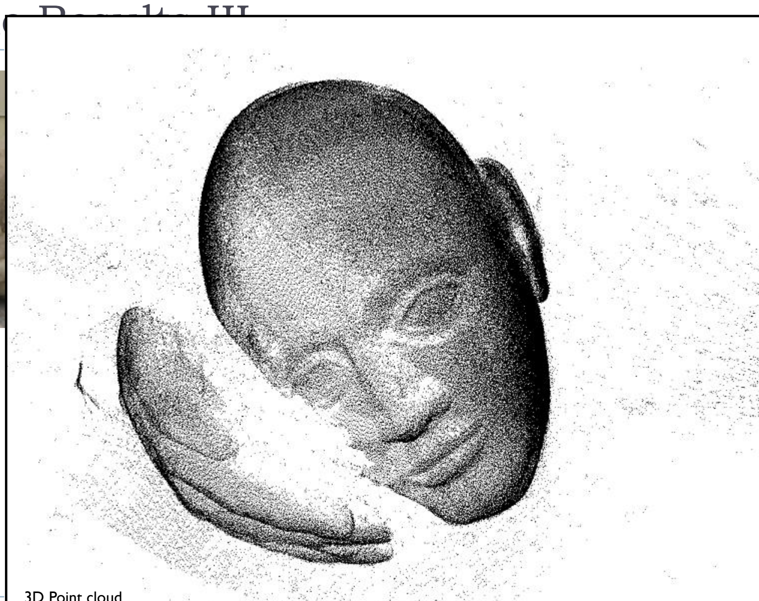
Some Results III



Disparity map

Source: Project IS-3D, Centre for Machine Perception, Czech Academy of Sciences, 2008.

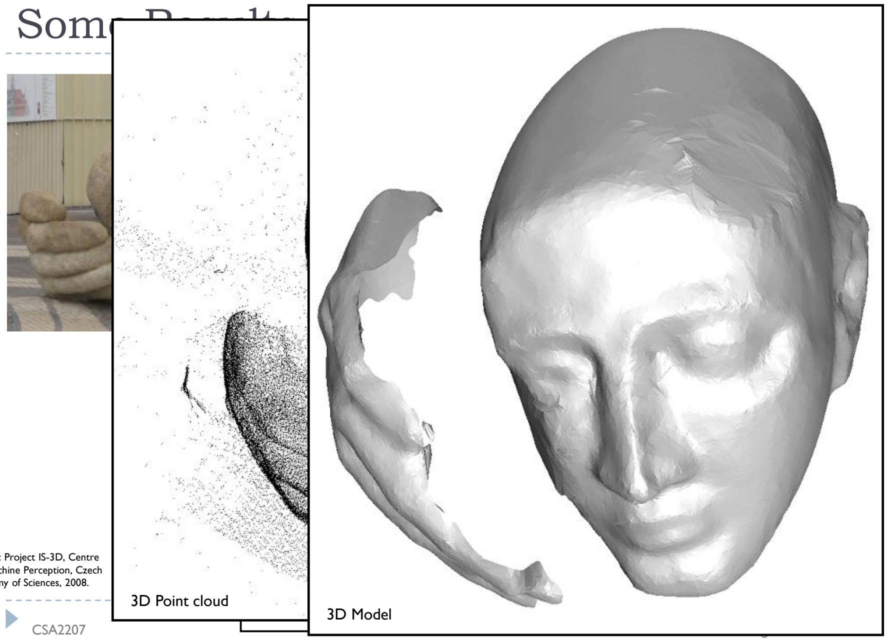
Some Results III



3D Point cloud

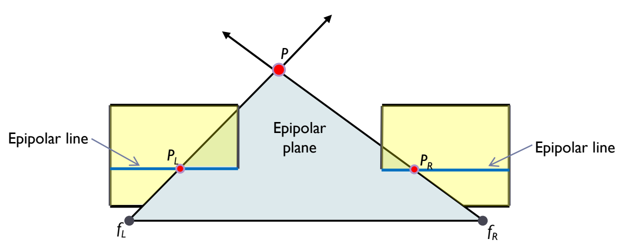
Source: Project IS-3D, Centre for Machine Perception, Czech Academy of Sciences, 2008.

Som



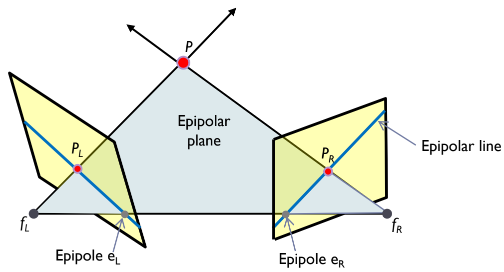
Source: Project IS-3D, Centre for Machine Perception, Czech Academy of Sciences, 2008.

Parallel camera configuration

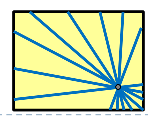


- Cameras oriented parallel to each other
- Conjugate epipolar lines map to the same image rows
- Requires precise positioning and orientation of the cameras
  - Difficult to achieve in practice

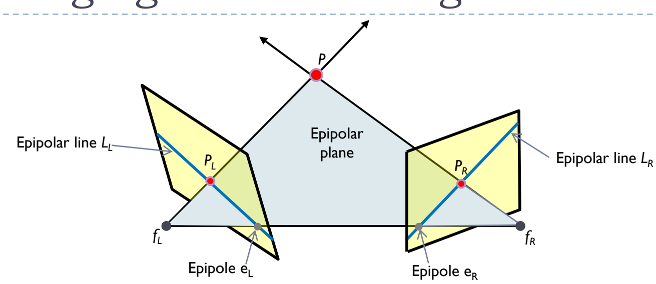
Converging camera configuration I



- Cameras no longer oriented parallel to each other.
- Conjugate epipolar lines no longer correspond to image rows and are not even parallel to each other.
- Each focal point projects onto a distinct point into the other camera's image plane
  - These are called the *epipoles* (or *epipolar points*)
  - All epipolar lines in an image must intersect the epipole.



Converging camera configuration II



- How can we find the epipolar lines since they are not parallel or image rows?
- Fundamental matrix F
  - This is a 3x3 matrix that relates any point P_L with P_R
- Epipolar Constraint can be expressed mathematically:

$$(P_L{}^T)F(P_R) = 0$$

- Also, multiplying the Fundamental matrix with a point gives the corresponding epipolar line in the other image:

$$F(P_L) = L_R \quad \text{and} \quad F^T(P_R) = L_L$$

Fundamental Matrix

- Given 2 corresponding image points $p = (x, y)$ and $p' = (x', y')$
- Epipolar constraint: $[x' \; y' \; 1]\left[\quad F \quad\right]\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$

$$F = \begin{bmatrix} 0 & -e'_w & e'_y \\ e'_w & 0 & -e'_x \\ -e'_y & e'_x & 0 \end{bmatrix} \times P'P^+$$

where $e' = [e'_x \quad e'_y \quad e'_w]$ is the epipole in the right image,
P, P' are the camera projection matrices,
and P^+ is the pseudo-inverse of matrix P.

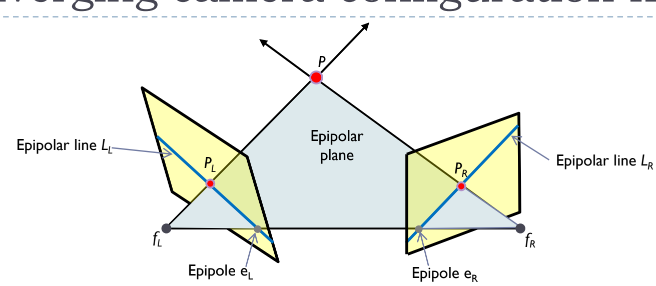
- For the parallel camera configuration, the fundamental matrix F simplifies to:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

- Hence: $[x'y'1]\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0 \quad \rightarrow \quad y' = y$

  and the epipoles are: $e = e' = [1 \;\; 0 \;\; 0]$ (at infinity).

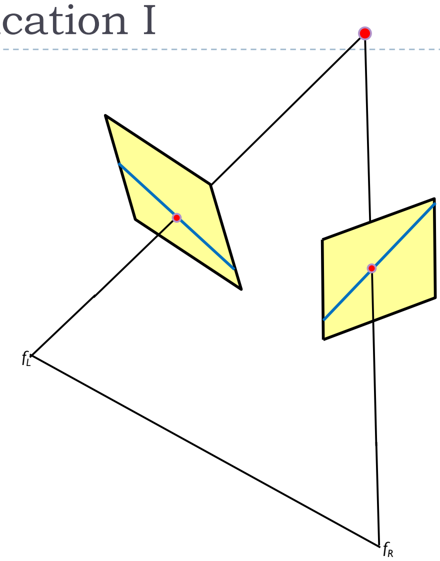
Converging camera configuration III



- Advantage:
  - No need for precise positioning and orientation of the cameras.
- Disadvantage:
  - Difficult to perform a search along epipolar lines during pixel matching.
- Solution:
  - Perform *stereo image rectification.*

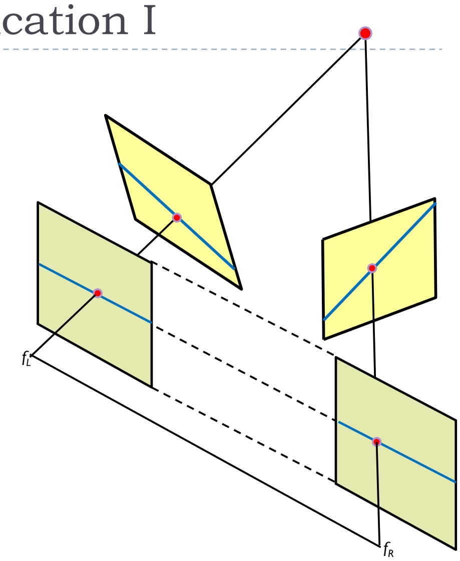
Stereo Image Rectification I

- Rectification is the process of re-sampling the stereo images so that the epipolar lines correspond to image rows.

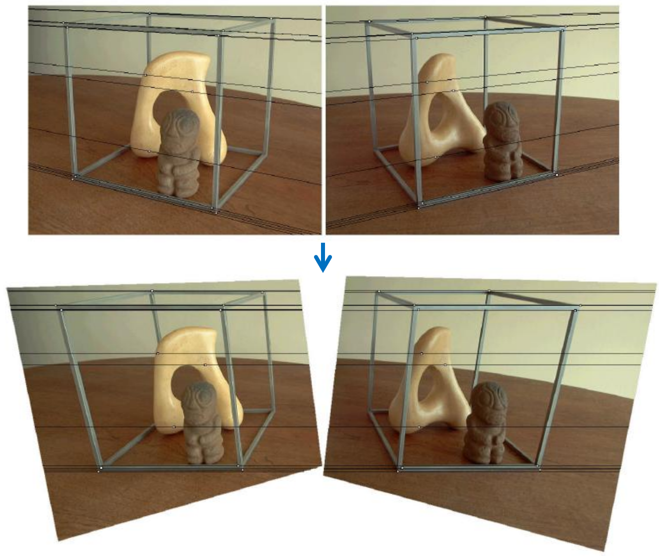


Stereo Image Rectification I

- Rectification is the process of re-sampling the stereo images so that the epipolar lines correspond to image rows.
- Images re-projected onto a common plane parallel to the line between focal points.



Stereo Image Rectification II



Some Results I

Source: M. Bujnak, Centre for Machine Perception, Czech Technical Univ.

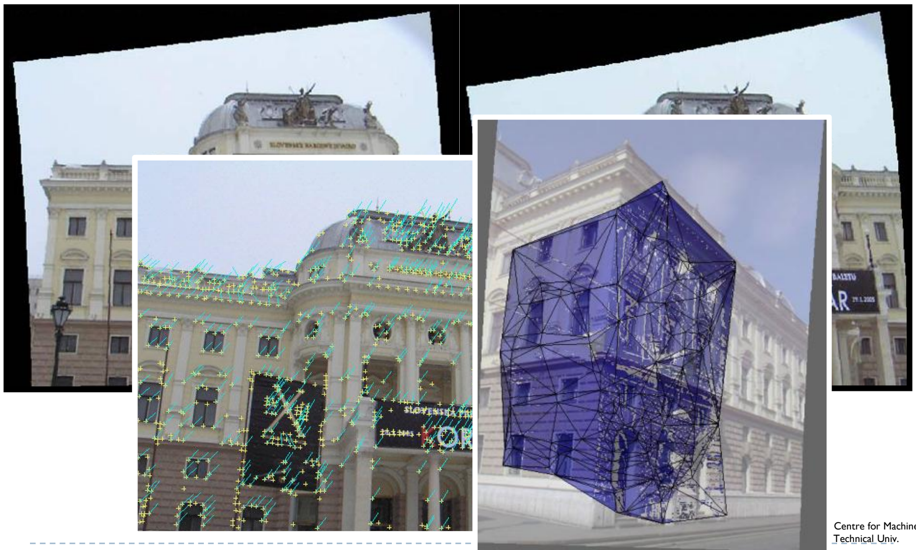
Some Results I

Source: M. Bujnak, Centre for Machine Perception, Czech Technical Univ.

Some Results I

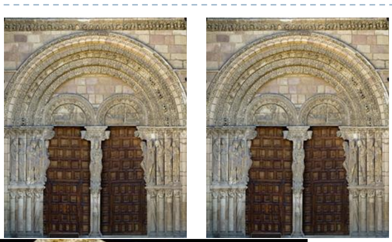
Source: M. Bujnak, Centre for Machine Perception, Czech Technical Univ.

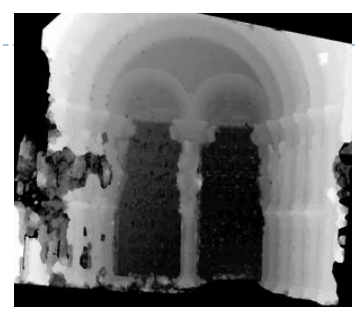
Some Results I



Centre for Machine
Technical Univ.

Some Results II





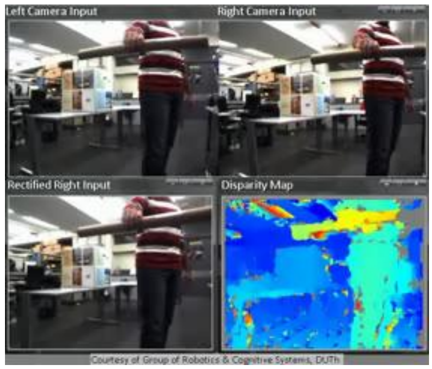
Disparity map

Synthetic image from disparity map

3D disparity point cloud

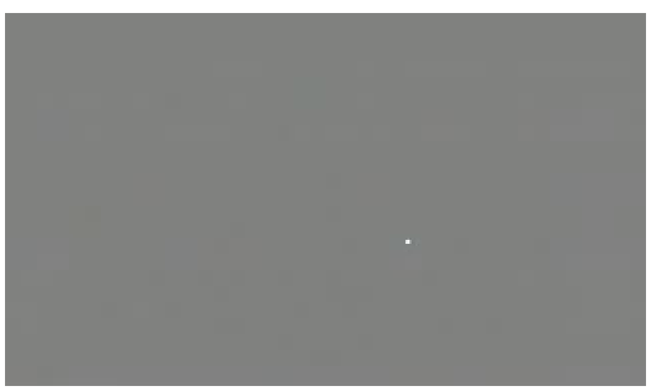
Source: www.metria.es

Some Results III



Source: Group of Robotics and Cognitive Systems, Democritus University of Thrace, Greece.

Some Results IV



Source: Marco Mengelkoch, Universitat Koblenz.

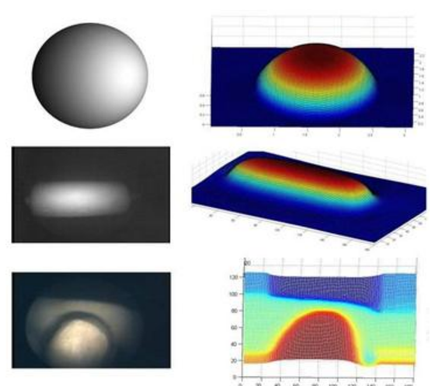
Depth from a single camera I

- What about this guy?

- Can we recover (some) depth information using only one sensor/camera?

Depth from a single camera II

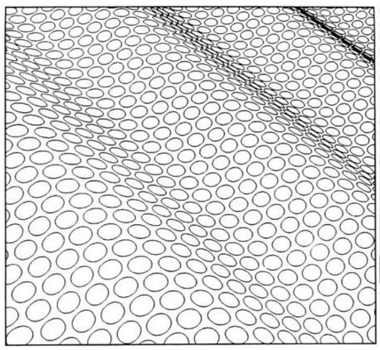
- Human vision does not rely solely on binocular vision for depth estimation
- Other visual cues can be used for 3D
- In CV, "Shape from X" techniques:
  - Shading

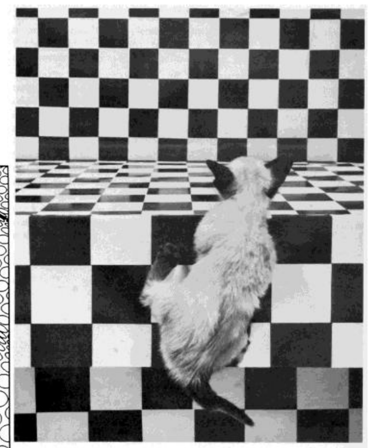


C. Wu et al., "Shape-from-Shading under Near Point Lighting and Partial views for Orthopeadic Endoscopy", 2007.

Depth from a single camera III

- Human vision does not rely solely on binocular vision for depth estimation
- Other visual cues can be used for 3D
- In CV, “Shape from X” techniques:
  - Shading
  - Texture

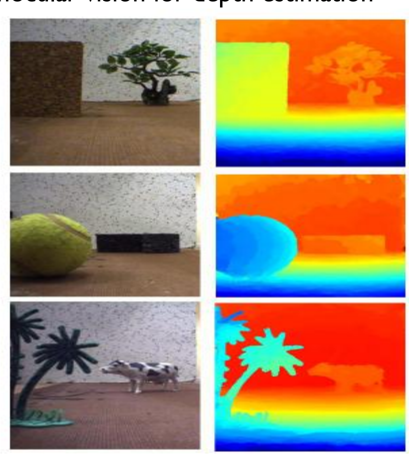




“The Visual Cliff”, William Vandivert, 1960.

Depth from a single camera IV

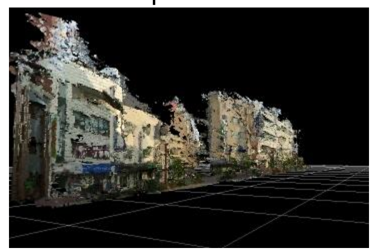
- Human vision does not rely solely on binocular vision for depth estimation
- Other visual cues can be used for 3D
- In CV, “Shape from X” techniques:
  - Shading
  - Texture
  - Focus



T. Aydin et al., “A New Adaptive Focus Measure for Shape From Focus”, 2008.

Depth from a single camera V

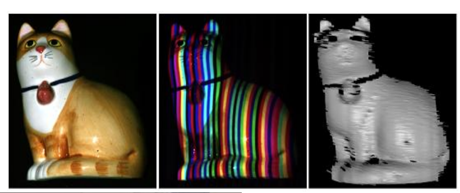
- Human vision does not rely solely on binocular vision for depth estimation
- Other visual cues can be used for 3D
- In CV, "Shape from X" techniques:
  - Shading
  - Texture
  - Focus
  - Motion
    - motion parallax
    - optical flow
  - & many others…

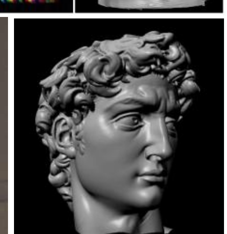


(1) T. Sato et al., "Reconstuction of 3-D Models of an Outdoor Scene from Multiple Image Sequences by Estimating Camera Motion Parameters". (2) K. Kutulakos, "A Theory of Shape by Space Carving".

Depth from a single camera VI

- Human vision does not rely solely on binocular vision for depth estimation
- Other visual cues can be used for 3D
- In CV, "Shape from X" techniques:
  - Shading
  - Texture
  - Focus
  - Motion
  - & many others…
- Structured light & laser scanning

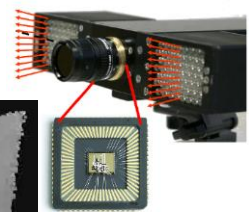




(1) L. Zhang et al., "Rapid Shape Acquisition Using Colour Structured Light and Multi-pass Dynamic Programming"
(2) The Digital Michelangelo project, Stanford Univ.

Depth from a single camera VII

- Human vision does not rely solely on binocular vision for depth estimation
- Other visual cues can be used for 3D
- In CV, "Shape from X" techniques:
  - Shading
  - Texture
  - Focus
  - Motion
  - & many others…
- Structured light & laser scanning
- Time-of-flight cameras

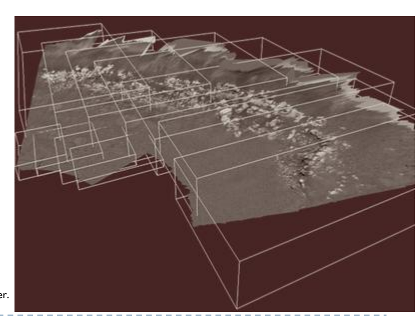


PMDT Technologies, GmbH.

Multi-View Stereo

- Monocular depth estimation
- Stereo Vision systems
  - 2 camera systems (Binocular systems)
  - Can extend the same process and algorithms to:
    - 3 camera systems (Trinocular systems)
    - 4 camera systems and more…
  - Multi-View Stereo systems

Commercial 2- and 3-camera systems, PointGrey Inc.

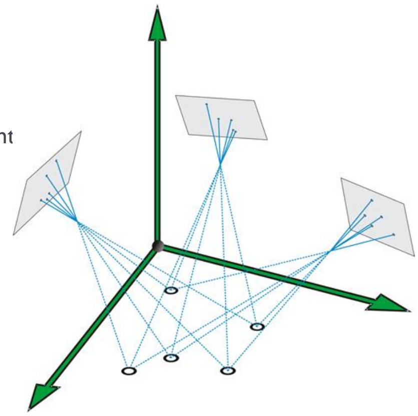


10 stereo reconstructions, MER Opportunity Rover. Source: ExoMars PanCam 3D Vision Team

Multi-View Stereo II

- Finding correspondences between adjacent rectified image pairs
- Pair-wise disparity estimation
- Fusing all the estimates into one 3D model
- *Bundle Adjustment* algorithm
  - Derived from the idea of "bundles" of light rays
  - Iteratively refining the 3D coordinates of the scene points (as well as the cameras' parameters) by minimising the re-projection error between the image locations of the observed and predicted image points
  - Minimisation through the use of the Levenberg-Marquardt algorithm

  - Triggs et al. "Bundle Adjustment – A Modern Synthesis", 1999.

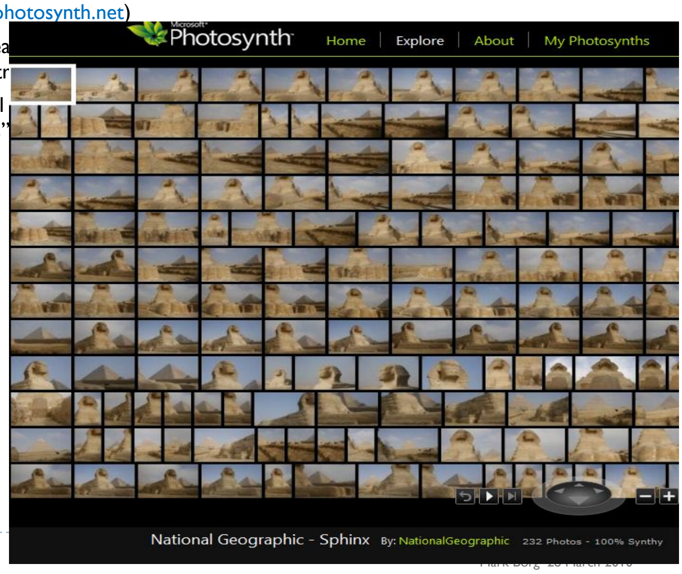


Example I

- Photosynth (http://photosynth.net)
- Extracts distinctive feature points in each image, matching these across the image set, and automatically reconstructs a partial 3D model of the scene and camera geometry.
- The sparse 3D model consists of a point cloud, line segments, and low-resolution "watercolour washes".

Example I

- Photosynth (http://photosynth.net)
- Extracts distinctive fea
  automatically reconstr
- The sparse 3D model
  "watercolour washes"

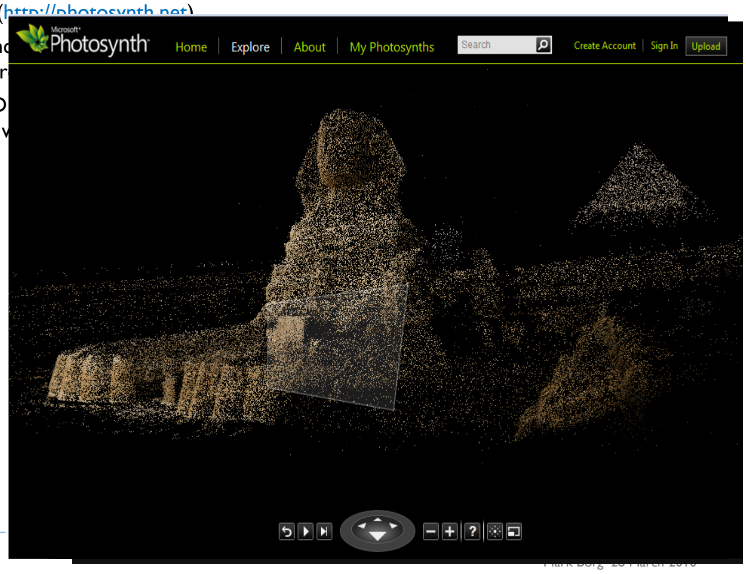


Example I

- Photosynth (http://photosynth.net)
- Extracts distinc
  automatically r
- The sparse 3D
  "watercolour v

Example I

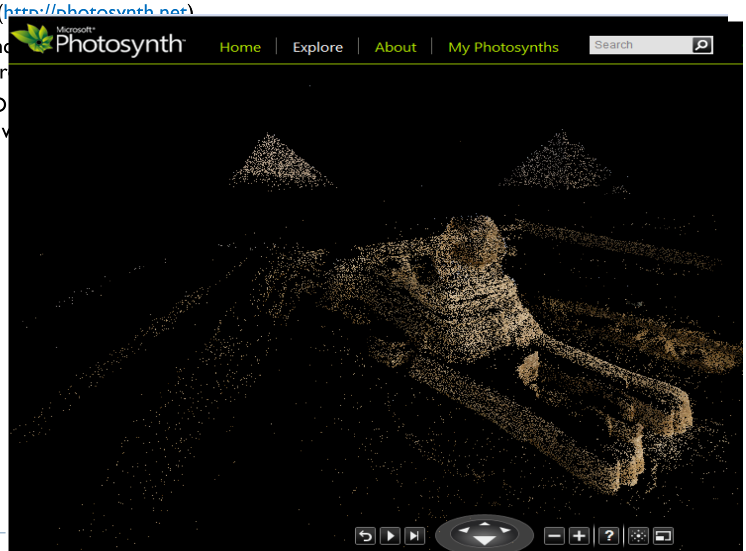
- Photosynth (http://photosynth.net)
- Extracts distinc
  automatically r
- The sparse 3D
  "watercolour v



CSA2207

Example I

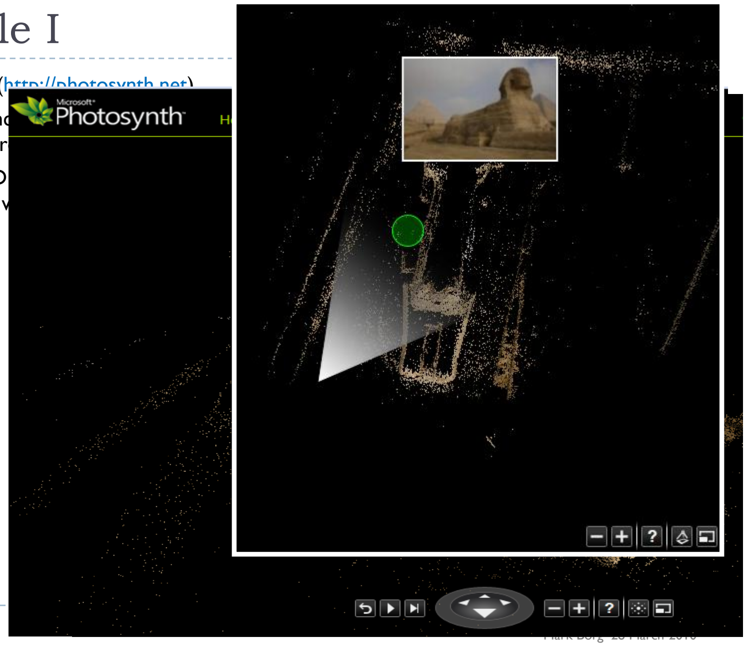
- Photosynth (http://photosynth.net)
- Extracts distinc
  automatically r
- The sparse 3D
  "watercolour v



CSA2207

Example I

- Photosynth (http://photosynth.net)
- Extracts distinc
  automatically r
- The sparse 3D
  "watercolour v



Example II

- Urban 3D modelling project using multi-camera systems (The University of North Carolina at Chapel Hill).
- Uses a 4-camera stereovision system mounted on a car
- Uses a multi-way plane sweeping stereovision algorithm

4-camera system

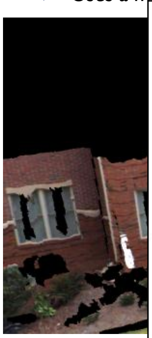
Example II

- Urban 3D modelling project using multi-camera systems (The University of North Carolina at Chapel Hill).
- Uses a 4-camera stereovision system mounted on a car
- Uses a multi-way plane sweeping stereovision algorithm

Reconstructed 3D view of part of a street

Example II

- Urban 3D modelling project using multi-camera systems (The University of North Carolina at Chapel Hill).
- Uses a 4-camera stereovision system mounted on a car
- Uses a multi-way plane sweeping stereovision algorithm



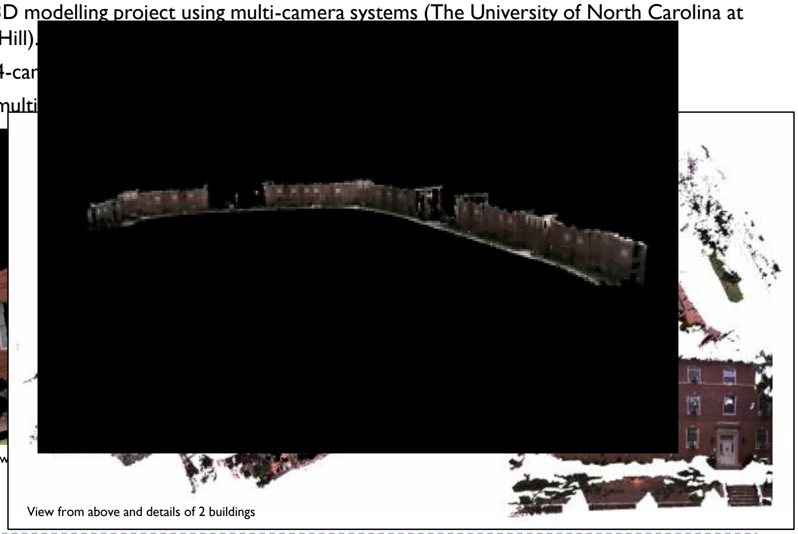
Reconstructed 3D view

View from above and details of 2 buildings

Example II

- Urban 3D modelling project using multi-camera systems (The University of North Carolina at Chapel Hill).
- Uses a 4-car
- Uses a multi

Reconstructed 3D view



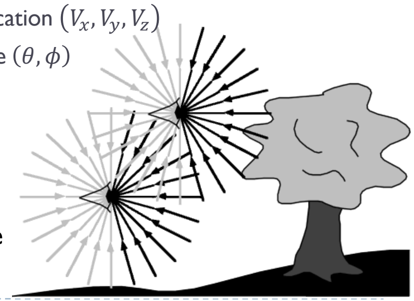
View from above and details of 2 buildings

The Plenoptic function I

- Plenoptic:
  - Plenus = complete/full + Optic = light
- 7-dimensional function:

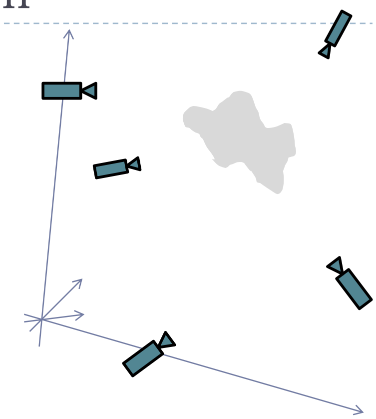
$$P(\theta, \phi, V_x, V_y, V_z, t, \lambda)$$

- To measure the plenoptic function one can imagine:
  - Placing an imaginary eye at every possible location (V_x, V_y, V_z)
  - Recording the intensity of light at every angle (θ, ϕ)
  - For every wavelength λ
  - At every time t
- The Plenoptic function is an idealised concept of a "complete view of the world"



The Plenoptic function II

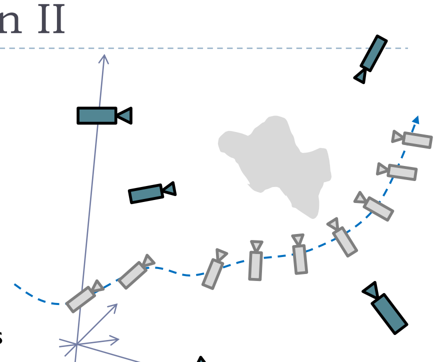
$$P(\theta, \phi, V_x, V_y, V_z, t, \lambda)$$

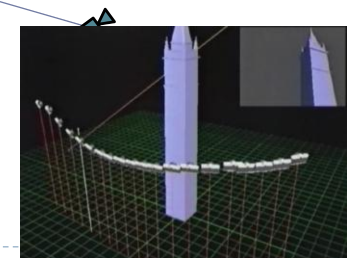


- We can only sample the plenoptic function at a finite number of points in space (*N*-camera system)
- What's the use of the plenoptic function?
  - Allows us to think about novel ways of sampling/navigating in this 7D space

The Plenoptic function II

$$P(\theta, \phi, V_x, V_y, V_z, t, \lambda)$$

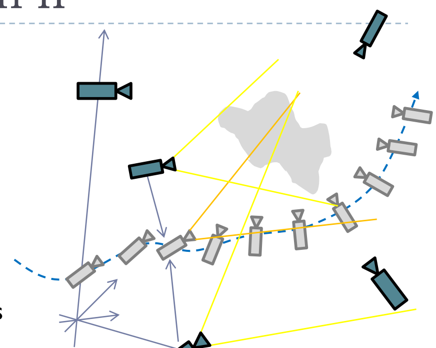




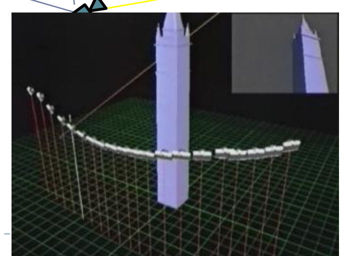
- We can only sample the plenoptic function at a finite number of points in space (*N*-camera system)
- What's the use of the plenoptic function?
  - Allows us to think about novel ways of sampling/navigating in this 7D space
- Given images acquired from *N* cameras (sparse sampling),
  - can we observe the scene by moving freely in space?
    - i.e., create a virtual camera, let it move along some trajectory/manifold in space, thus creating a so-called *free-viewpoint video*?

The Plenoptic function II

$$P(\theta, \phi, V_x, V_y, V_z, t, \lambda)$$

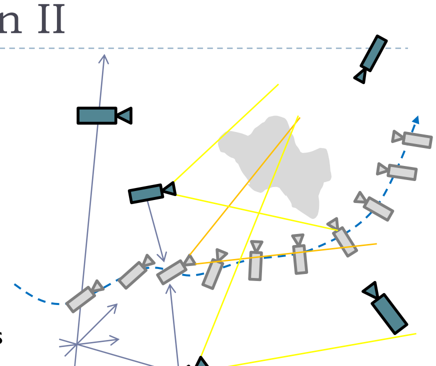


- We can only sample the plenoptic function at a finite number of points in space (*N*-camera system)
- What's the use of the plenoptic function?
  - Allows us to think about novel ways of sampling/navigating in this 7D space
- Given images acquired from *N* cameras (sparse sampling),
  - can we observe the scene by moving freely in space?
    - i.e., create a virtual camera, let it move along some trajectory/manifold in space, thus creating a so-called *free-viewpoint video?*
      - Need to be able to synthesise new views

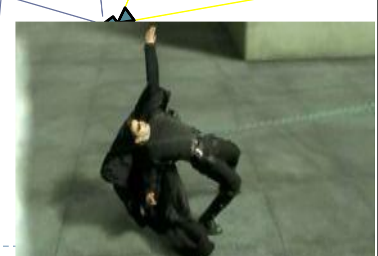


The Plenoptic function II

$$P(\theta, \phi, V_x, V_y, V_z, t, \lambda)$$



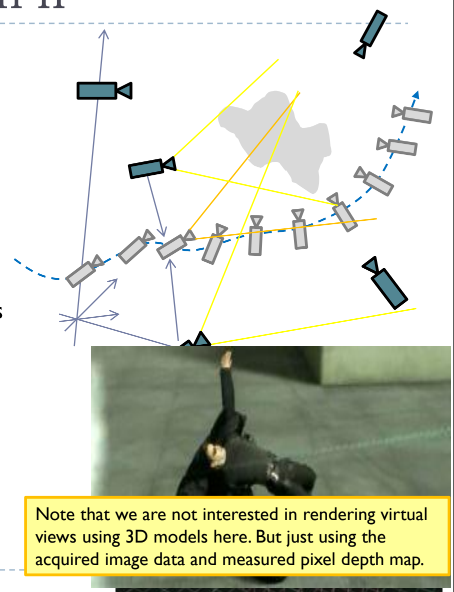
- We can only sample the plenoptic function at a finite number of points in space (*N*-camera system)
- What's the use of the plenoptic function?
  - Allows us to think about novel ways of sampling/navigating in this 7D space
- Given images acquired from *N* cameras (sparse sampling),
  - can we observe the scene by moving freely in space?
    - i.e., create a virtual camera, let it move along some trajectory/manifold in space, thus creating a so-called *free-viewpoint video?*
      - Need to be able to synthesise new views
  - can we let the virtual camera move in space while freezing time?
    - i.e., create a so-called "bullet time" special effect?



The Plenoptic function II

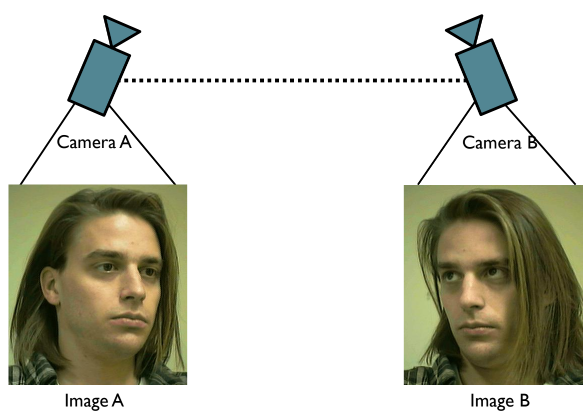
$$P(\theta, \phi, V_x, V_y, V_z, t, \lambda)$$

- We can only sample the plenoptic function at a finite number of points in space (*N*-camera system)
- What's the use of the plenoptic function?
  - Allows us to think about novel ways of sampling/navigating in this 7D space
- Given images acquired from *N* cameras (sparse sampling),
  - can we observe the scene by moving freely in space?
    - i.e., create a virtual camera, let it move along some trajectory/manifold in space, thus creating a so-called *free-viewpoint video*?
      - Need to be able to synthesise new views
  - can we let the virtual camera move in space while freezing time?
    - i.e., create a so-called "bullet time" special effect?



Note that we are not interested in rendering virtual views using 3D models here. But just using the acquired image data and measured pixel depth map.

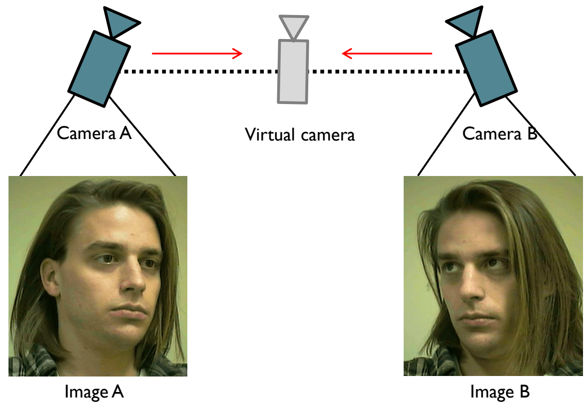
View Morphing I



Source: S. Seitz, C. Dyer, 1996.

View Morphing I

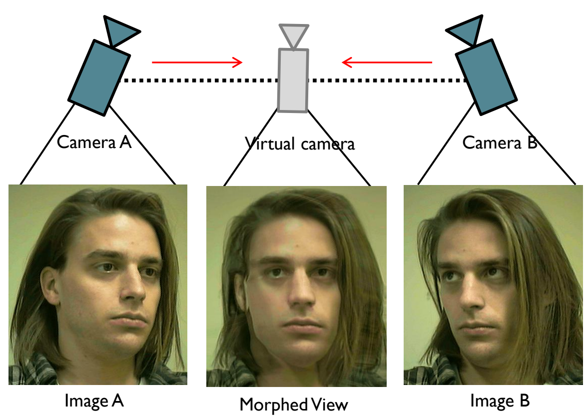
- Virtual camera



Source: S. Seitz, C. Dyer, 1996.

View Morphing I

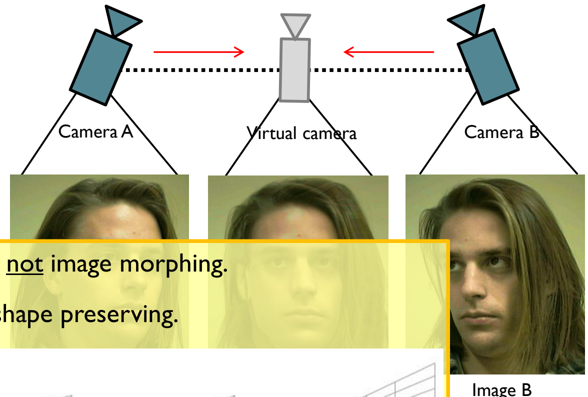
- Virtual camera
- View morphing



Source: S. Seitz, C. Dyer, 1996.

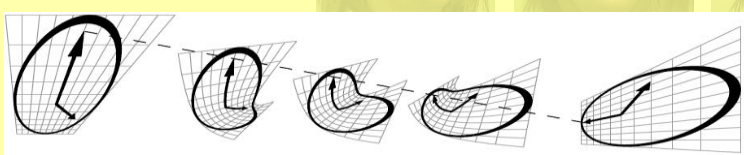
View Morphing I

- Virtual camera
- View morphing



Note that view morphing is <u>not</u> image morphing.

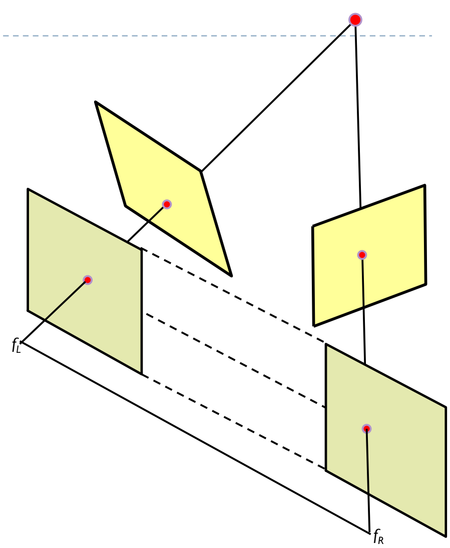
Image morphing is not 3D shape preserving.



Source: S. Seitz, C. Dyer, 1996.

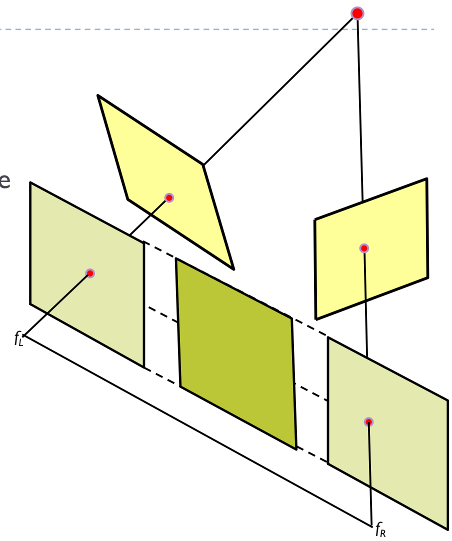
View Morphing II

- Algorithm:
  - Stereo images are first rectified



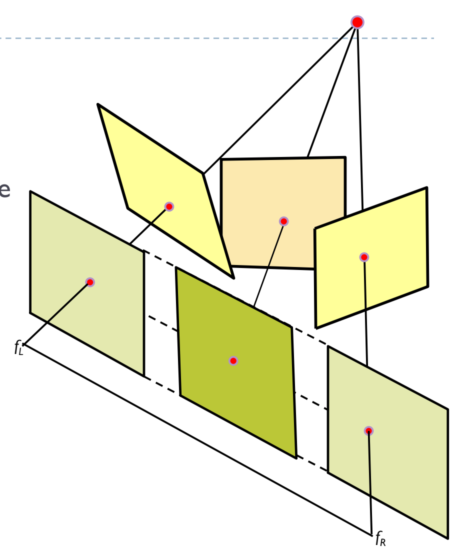
View Morphing III

- Algorithm:
  - Stereo images are first rectified
  - Then a virtual camera is positioned in the common image plane
    - *View Morphing*



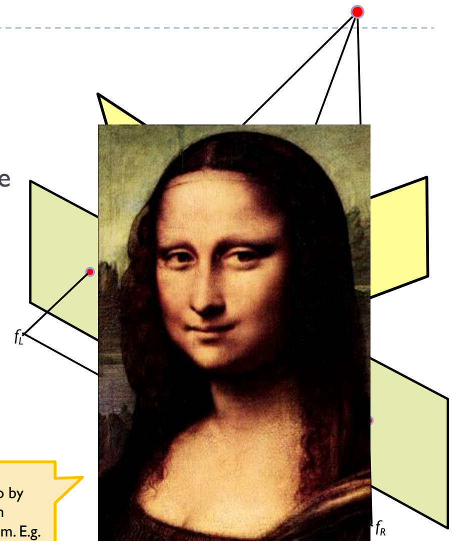
View Morphing IV

- Algorithm:
  - Stereo images are first rectified
  - Then a virtual camera is positioned in the common image plane
    - *View Morphing*
  - The camera is then moved and oriented as required



View Morphing IV

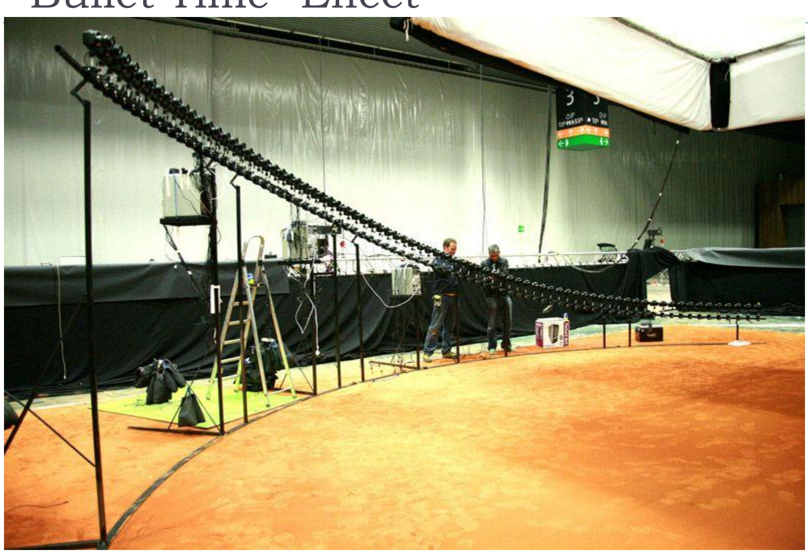
- Algorithm:
  - Stereo images are first rectified
  - Then a virtual camera is positioned in the common image plane
    - *View Morphing*
  - The camera is then moved and oriented as required



Source: S. Seitz, C. Dyer, 1996.

Note that we can also 'acquire' multiple views of a single photo by operations like mirroring. Then perform view morphing on them. E.g. Mona Lisa sequence:

"Bullet Time" Effect



“Bullet Time” Effect

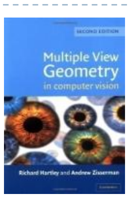
“Bullet Time” Effect

Some demos

- The Campanile movie
  - Paul Debevac, Univ. of California, Berkeley, 1996.
  - http://www.debevac.org/campanile
- ProFORMA
  - Qi Pan, Univ. of Cambridge, 2009.
  - http://mi.eng.cam.ac.uk/~qp202

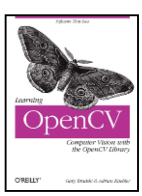
Paul Debevac, 1996.

Further Information…



"Multiple View Geometry in Computer Vision", Richard Hartley, Andrew Zisserman, 2nd Ed., 2004.

OpenCV



Open Source Computer Vision Library

C++ library containing lots of CV algorithms, including stereo vision algorithms.

http://sourceforge.net/projects/opencvlibrary/

Microsoft Research

The MSRC Stereo Vision C# SDK

Microsoft Research in Cambridge

http://research.microsoft.com/en-us/projects/i2i/default.aspx