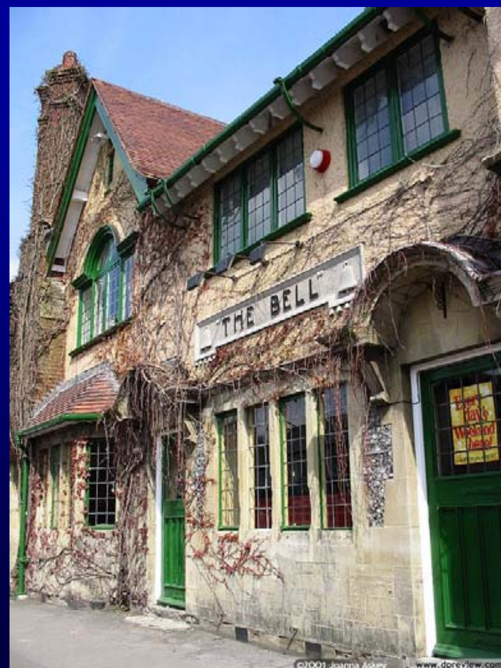
Image Based Rendering

an overview

Photographs

- We have tools that acquire and tools that display photographs at a convincing quality level

2


THE BELL
©2001 Joanna Askey
www.dpreview.com

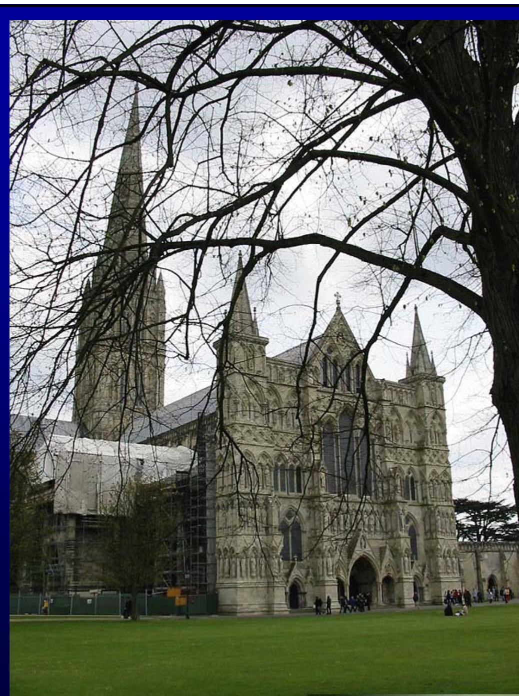
3


©2001 Phil Askey
www.dpreview.com

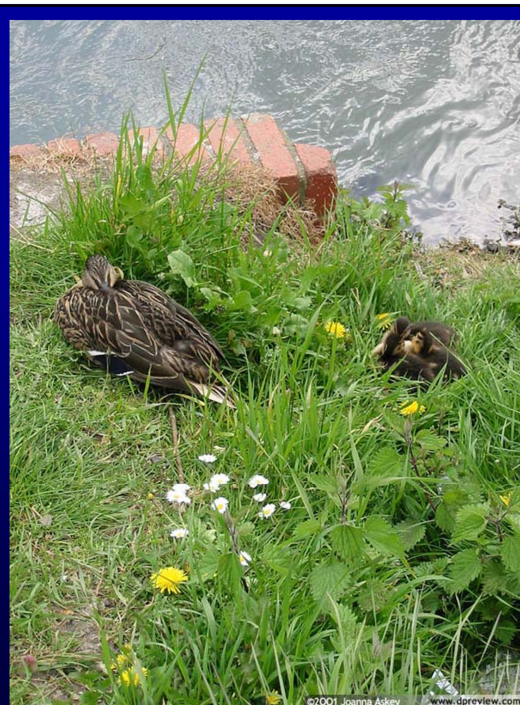
4


©2001 Joanna Askey
www.dpreview.com

5


©2001 Joanna Askey
www.dpreview.com

6


©2001 Joanna Askey
www.dpreview.com

7


©2001 Joanna Askey
www.dpreview.com

8


TIME
©2001 Phil Askey
www.dpreview.com

9


©2001 Joanna Askey
www.dpreview.com

10

Photographs

- We have tools that acquire and tools that display photographs at a convincing quality level, for almost 100 years now

11

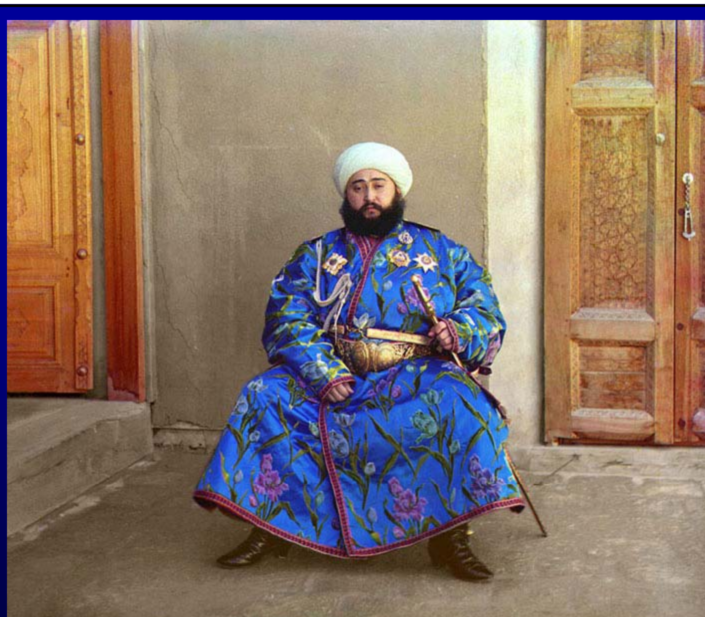
Sergei Mikhailovich Prokudin-Gorskii.

A Settler's Family, ca. 1907-1915.

12

Sergei Mikhailovich Prokudin-Gorskii.
Tea Factory in Chakva.
Chinese Foreman Lau-Dzhen-Dzhau.
ca. 1907-1915.

13

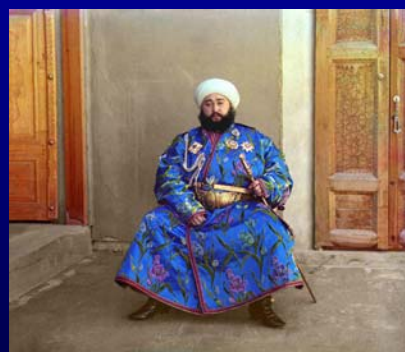


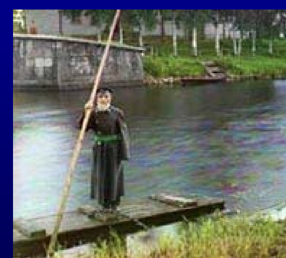
Sergei Mikhailovich Prokudin-Gorskii.

The Emir of Bukhara, 1911.

14

RGB in early 1900's





15

Plenoptic function

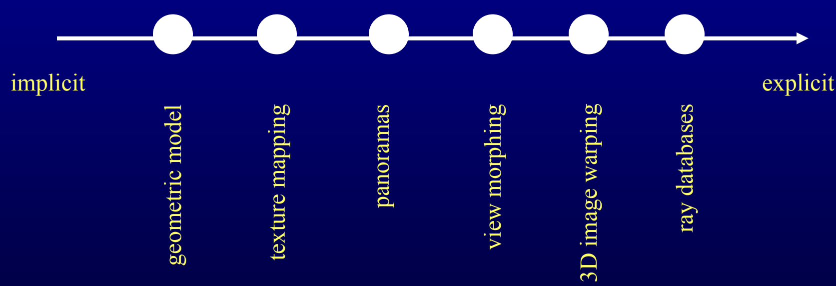
- Defines all the rays
  - through any point in space (x, y, z)
  - with any orientation (θ, φ)
  - over all wavelenghts (λ)
  - at any given moment in time (t)

$$\rho = P(x, y, z, \phi, \varphi, \lambda, t)$$

16

IBR summary

Representation of plenoptic function



17

Lightfield – Lumigraph approach
[Levoy96, Gortler96]

- Take all photographs you will ever need to display
- Model becomes database of rays
- Rendering becomes database querying

18

Overview

- Introduction
- Lightfield – Lumigraph
  - definition
  - construction
  - compression

19

Overview

- Introduction
- Lightfield – Lumigraph
  - definition
  - construction
  - compression

20

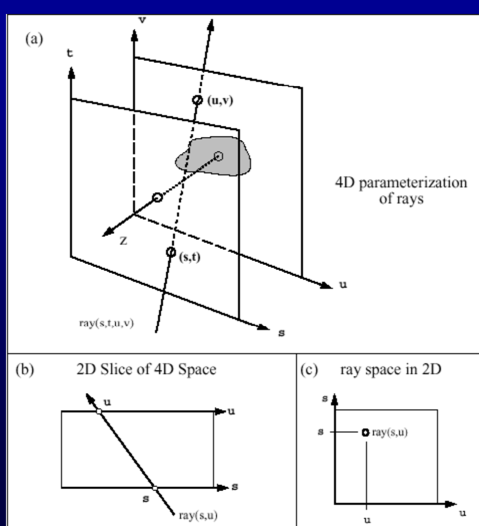
From 7D to 4D

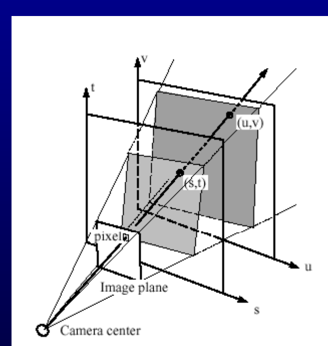
$$\rho = P(x, y, z, \phi, \varphi, \lambda, t)$$

- Static scene, t constant
- λ approximated with RGB
- consider only convex hull of objects, so the origin of the ray does not matter

21

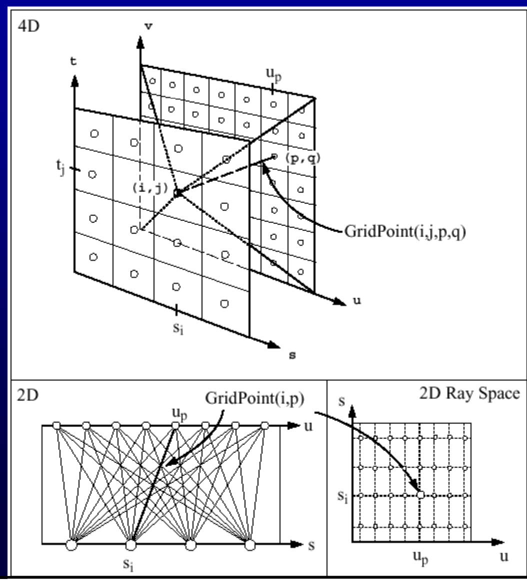
4D Lightfield / Lumigraph





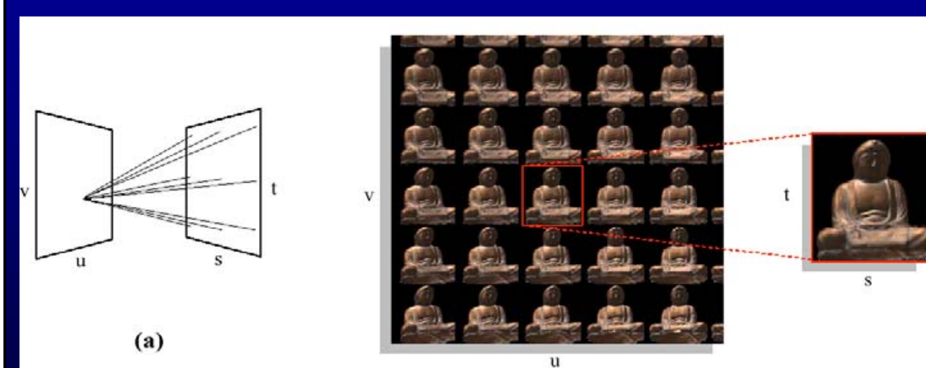
22

Discreet 4D Lightfield



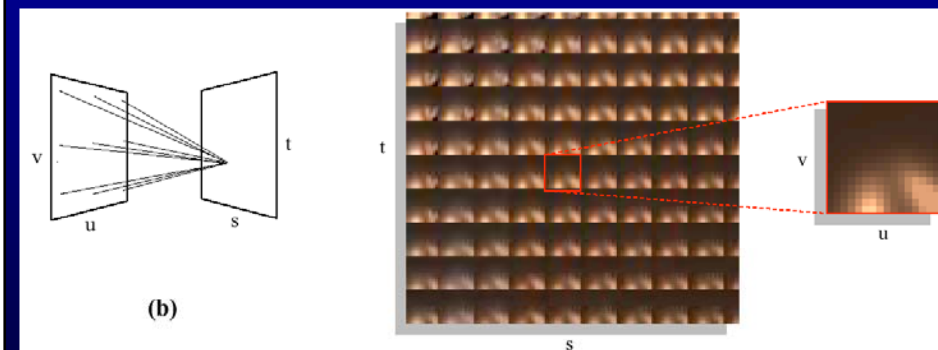
23

Lightfield: set of images with COPs on regular grid

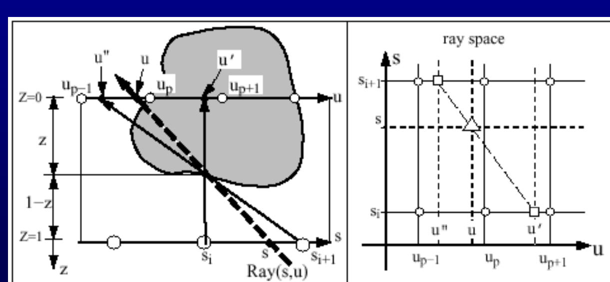


24

or Lightfield: set of images of a point seen at various angles



Depth correction of rays



Overview

- Introduction
- Lightfield – Lumigraph
  - definition
  - construction
  - compression

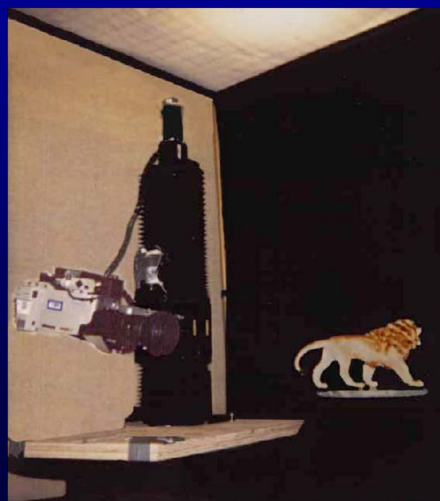
27

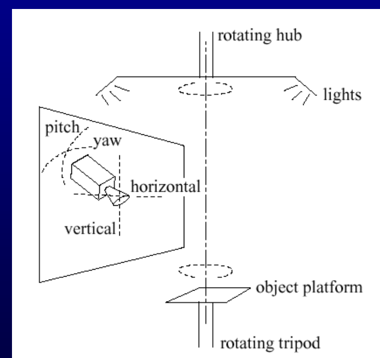
Overview

- Introduction
- Lightfield – Lumigraph
  - definition
  - construction
  - compression

28

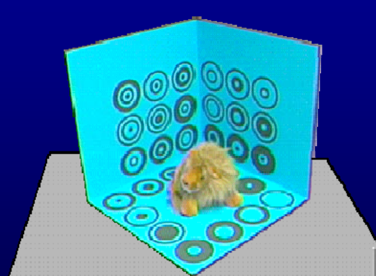
Construction from dense set of photographs



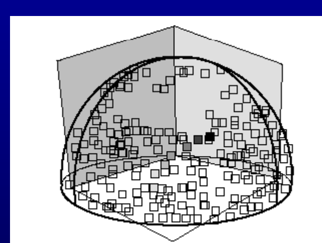


29

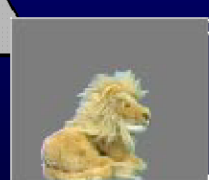
Construction from sparse set of photographs



acquisition stage



camera positions

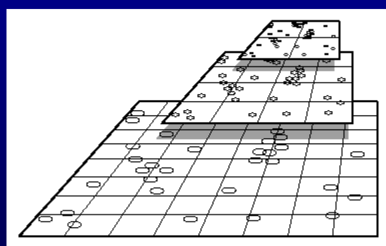


blue screening

space carving

30

Filling in gaps using pull-push algorithm



- Pull phase
  - low res levels are created
  - gaps are shrunk
- Push phase
  - gaps at high res levels are filled using low res levels

Overview

- Introduction
- Lightfield – Lumigraph
  - definition
  - construction
  - compression

Overview

- Introduction
- Lightfield – Lumigraph
  - definition
  - construction
  - compression

33

Compression

- Large size uncompressed: 1.125GB
  - 32x32 (s, t) x 256x256 (u, v) x 6 faces x 3 B
- Compression
  - jpeg + mpeg (200:1 to 6MB)
  - or vector quantization + entropy encoding

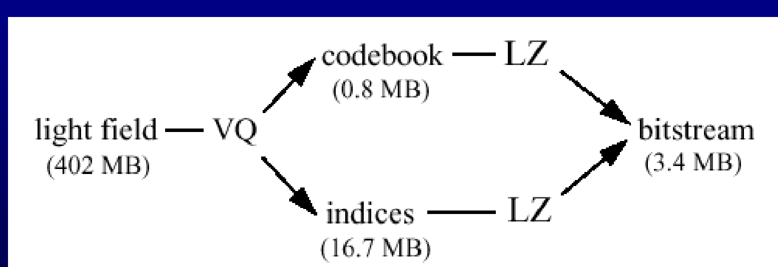
34

Vector Quantization (VQ)

- Principle
  - codebook made of codewords
  - replace actual word with closest codeword
- Implementation
  - training on representative set of words to derive best codebook
  - compression: replacing word with index to closest codeword
  - decompression: retrieve indexed codeword from codebook

35

Lightfield compression using VQ

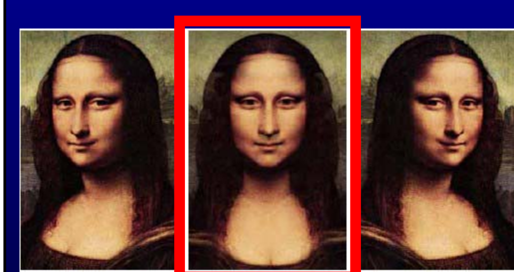


36

View morphing

37

Motivation – rendering from images



[Seitz96]

- Given
  - left image
  - right image
- Create intermediate images
  - simulates camera movement

38

Previous work

- Panoramas ([Chen95], etc)
  - user can look in any direction at few given locations
- Image-morphing ([Wolberg90], [Beier92], etc)
  - linearly interpolated intermediate positions of features
  - input: two images and correspondences
  - output: metamorphosis of one image into other as sequence of intermediate images

39

Previous work limitations

- Panoramas ([Chen95], etc.)
  - no camera translations allowed
- Image morphing ([Wolberg90], [Beier92], etc.)
  - not *shape-preserving*
  - image morphing is also a morph of the object
  - to simulate rendering with morphing, the object should be rigid when camera moves

40

Overview

- Introduction
- Image morphing
- View morphing
  - image pre-warping
  - image morphing
  - image post-warping

41

Overview

- Introduction
- Image morphing
- View morphing
  - image pre-warping
  - image morphing
  - image post-warping

42

Image morphing

1. Correspondences

43

Image morphing

1. Correspondences

44

Image morphing

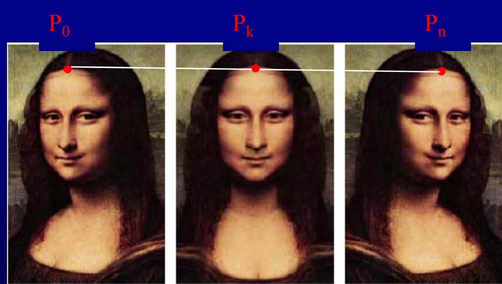
1. Correspondences

Image morphing

1. Correspondences

Image morphing

1. Correspondences
2. Linear interpolation



$$\dot{P}_k = (1-\frac{k}{n})\,\dot{P}_0 + \frac{k}{n}\,\dot{P}_n$$

47

Image morphing

- Image morphing
  - not shape preserving

48

Early IBR research

Soft watch at moment of first explosion – Salvador Dali 1954

49

Overview

- Introduction
- Image morphing
- View morphing
  - image pre-warping
  - image morphing
  - image post-warping

50

Overview

- Introduction
- Image morphing
- View morphing
  - image pre-warping
  - image morphing
  - image post-warping

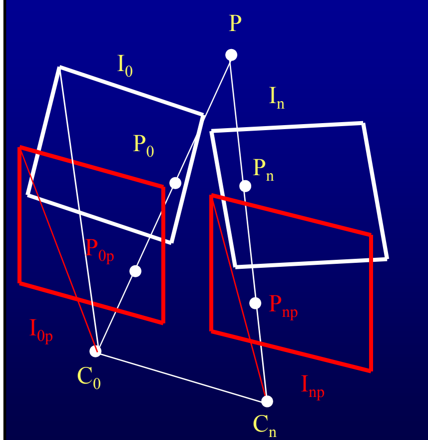
51

View morphing

- Shape preserving morph
- Three step algorithm
  1. Prewarp first and last images to parallel views
  2. Image morph between prewarped images
  3. Postwarp to interpolated view

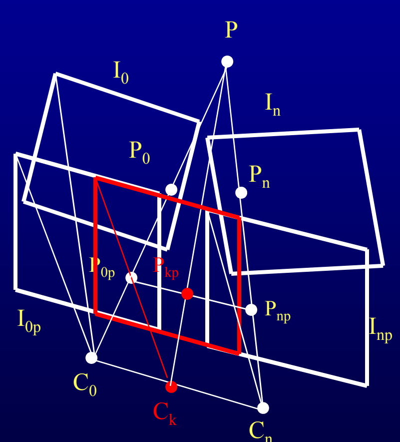
52

Step 1: prewarp to parallel views



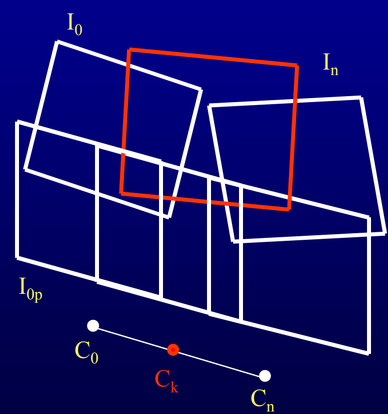
- Parallel views
  - same image plane
  - image plane parallel to segment connecting the two centers of projection
- Prewarp
  - compute parallel views I_{0p}, I_{np}
  - rotate I_0 and I_n to parallel views
  - prewarp corrs. (P_0, P_n) -> (P_{op}, P_{np})

Step 2: morph parallel images



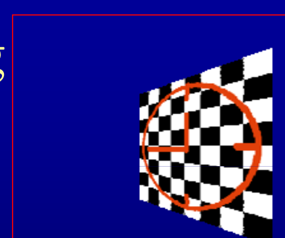
- Shape preserving
- Use prewarped correspondences
- Interpolate C_k from C_0 C_n

Step 3: Postwarping

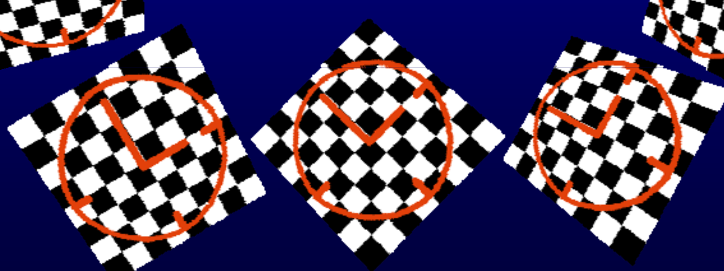


- Postwarp morphed image
  - create intermediate view
    - C_k is known
    - interpolate view direction and tilt
  - rotate morphed image to intermediate view

View morphing



- View morphing
  - shape preserving

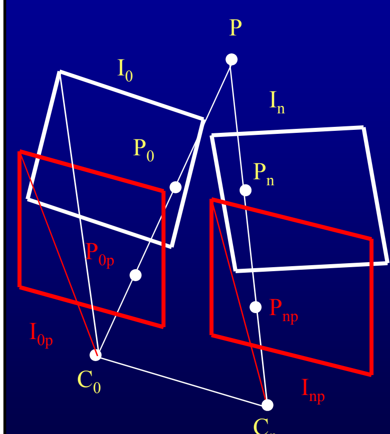


Overview

- Introduction
- Image morphing
- View morphing, more details
  - image pre-warping
  - image morphing
  - image post-warping

57

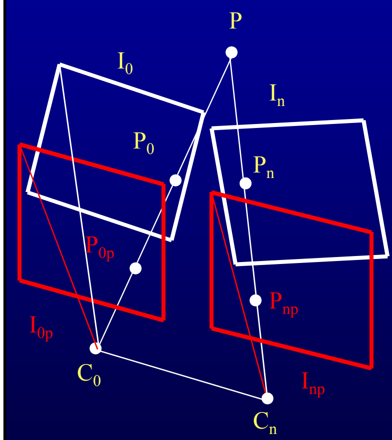
Step 1: prewarp to parallel views



- Parallel views
  - use C_0C_n for x (a_p vector)
  - use (a_0 x b_0) x (a_n x b_n) as y ($-b_p$)
  - pick a_p and b_p to resemble a_0 b_0 as much as possible
  - use same pixel size
  - use wider field of view

58

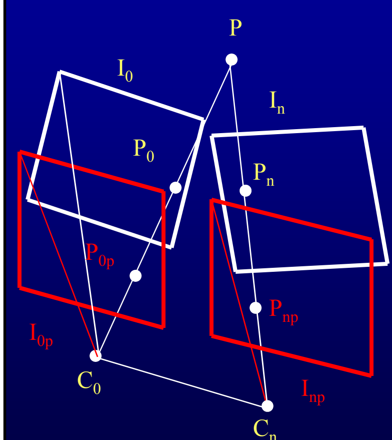
Step 1: prewarp to parallel views



- prewarping using texture mapping
  - create polygon for image plane
  - consider it texture mapped with the image itself
  - render the "scene" from prewarped view
  - if you go this path you will have to implement clipping with the COP plane
  - you have texture mapping already
- alternative: prewarping using reprojection of rays
  - look up all the rays of the prewarped view in the original view

59

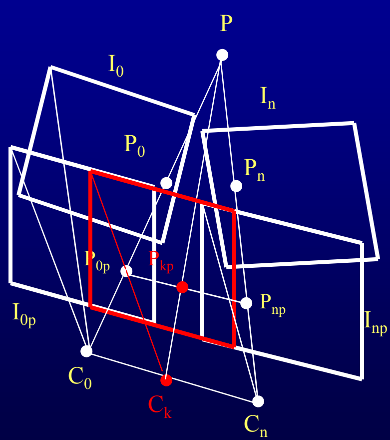
Step 1: prewarp to parallel views



- prewarping correspondences
  - for all pairs of correspondence P_0 P_n
    - project P_0 on I_{0p}, computing P_{0p}
    - project P_n on I_{np}, computing P_{np}
    - prewarped correspondence is P_{op} P_{np}

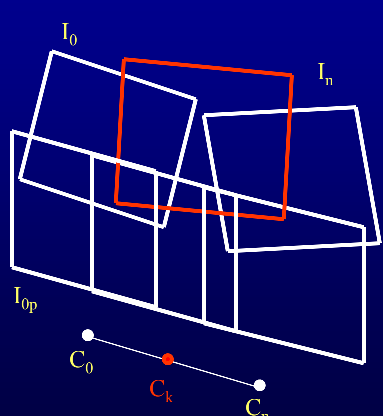
60

Step 2: morph parallel images



- Image morphing
  - use prewarped correspondences to compute a correspondence for all pixels in I_{0p}
  - linearly interpolate I_{0p} to intermediate positions
  - useful observation
    - corresponding pixels are on same line in prewarped views
  - preventing holes
    - use larger footprint (ex 2x2)
    - or linearly interpolate between consecutive samples
    - or postprocess morphed image looking for background pixels and replacing them with neighboring values
  - visibility artifacts
    - collision of samples
      - zbuffer on disparity
    - holes
      - morph I_{np} to I_{kp}
      - use additional views

Step 3: Postwarping



- create intermediate view
  - C_k is known
  - current view direction is a linear interpolation of the start and end view directions
  - current up vector is a linear interpolation of the start and end up vectors
- rotate morphed image to intermediate view
  - same as prewarping

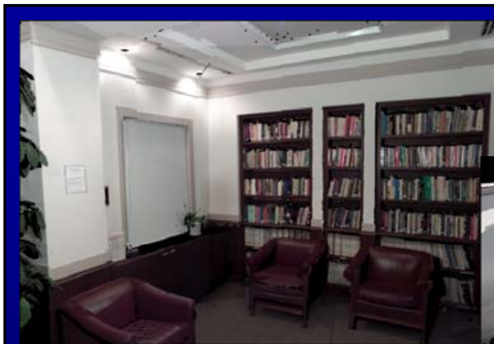


Image-Based Rendering by Warping

63

Overview

- Introduction
- Depth extraction methods
- Reconstruction for IBRW
- Visibility without depth
- Sample selection

64

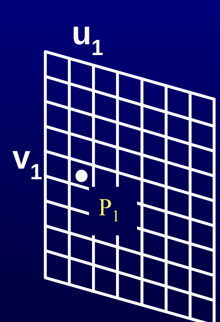
Overview

- Introduction
  - comparison to other IBR methods
  - 3D warping equation
  - reconstruction

65

IBR by Warping (IBRW)

- Images enhanced with per-pixel depth [McMillan95]

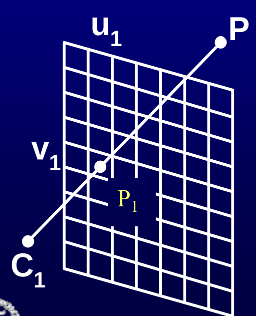


66

IBR by Warping (IBRW)

$$\dot{P} = \dot{C}_1 + (\bar{c}_1 + u_1\bar{a}_1 + v_1\bar{b}_1)w_1$$

$$w_1 = \frac{C_1P}{C_1P_1}$$



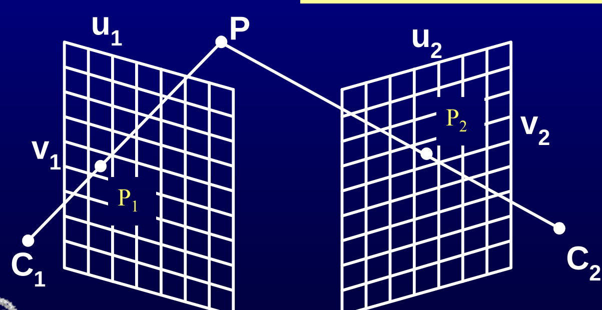
- $1/w_1$ also called generalized disparity
- another notation $\delta(u_1, v_1)$

67

IBR by Warping (IBRW)

$$\dot{P} = \dot{C}_1 + (\bar{c}_1 + u_1\bar{a}_1 + v_1\bar{b}_1)w_1$$

$$\dot{P} = \dot{C}_2 + (\bar{c}_2 + u_2\bar{a}_2 + v_2\bar{b}_2)w_2$$

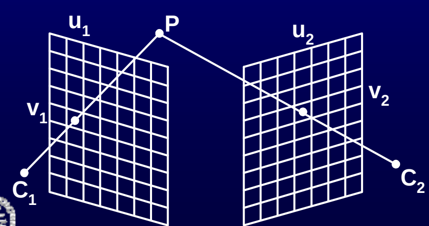


68

3D warping equations

$$u_2 = \frac{w_{11} + w_{12} \cdot u_1 + w_{13} \cdot v_1 + w_{14} \cdot \delta(u_1, v_1)}{w_{31} + w_{32} \cdot u_1 + w_{33} \cdot v_1 + w_{34} \cdot \delta(u_1, v_1)}$$

$$v_2 = \frac{w_{21} + w_{22} \cdot u_1 + w_{23} \cdot v_1 + w_{24} \cdot \delta(u_1, v_1)}{w_{31} + w_{32} \cdot u_1 + w_{33} \cdot v_1 + w_{34} \cdot \delta(u_1, v_1)}$$



A complete IBR method

- Façade system
  - (coarse) geometric model needed
- Panoramas
  - viewer confined to center of panorama
- View morphing
  - correspondences needed
- IBRW
  - rendering to arbitrary new views

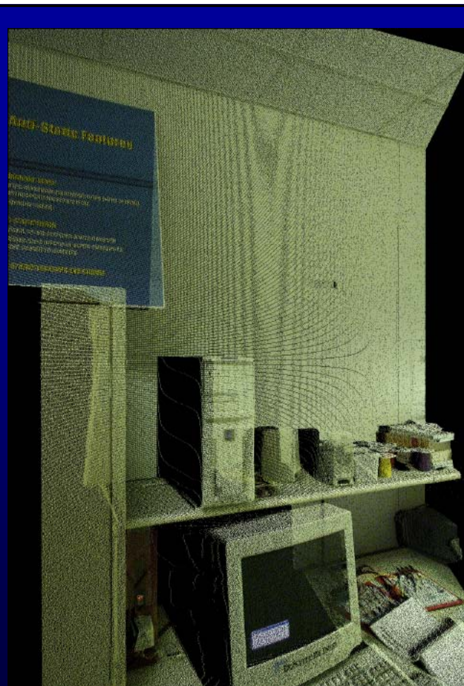
DeltaSphere - depth&color acquisition device

- Lars Nyland *et al.*

71

72

73



74

75

Reconstructing by splatting

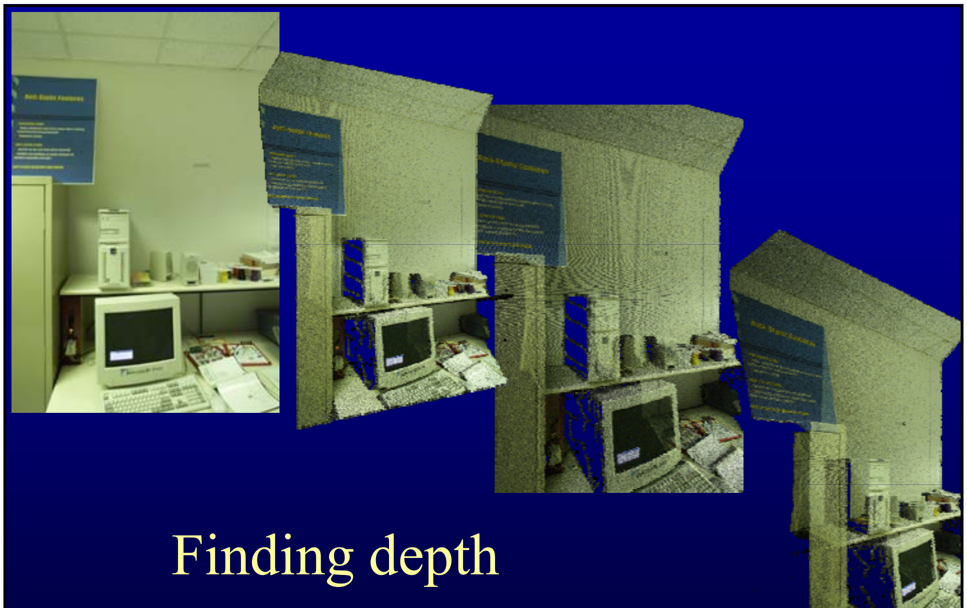
- Estimate shape and size of footprint of warped samples
  - expensive to do accurately
  - lower image quality if crudely approximated
- Samples are z-buffered

76

Overview

- Introduction
- Depth extraction methods
- Reconstruction for IBRW
- Visibility without depth
- Sample selection

77



Finding depth

78

Overview

- Depth from stereo
- Depth from structured light
- Depth from focus / defocus
- Laser rangefinders

79

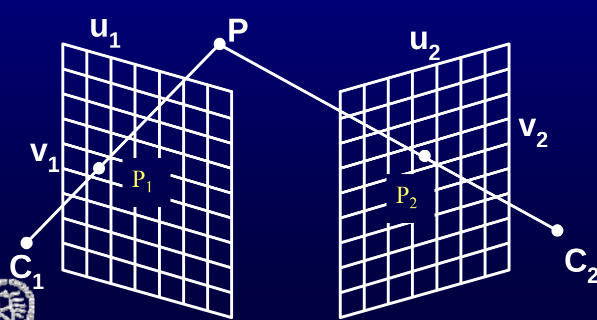
Overview

- Depth from stereo
- Depth from structured light
- Depth from focus / defocus
- Laser rangefinders

80

Depth from stereo

- two cameras with known parameters
- infer 3D location of point seen in both images
- sub problem: correspondences
  - for a point seen in the left image, find its projection in the right image

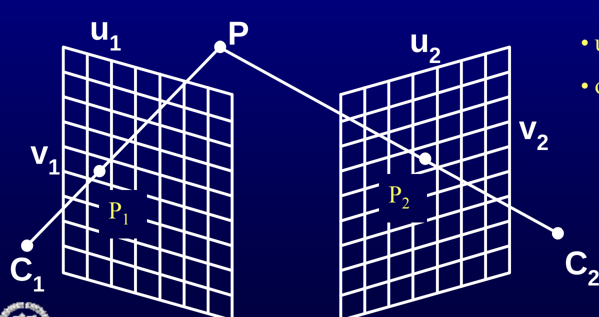


81

Depth from stereo: *déjà-vu math*

$$\dot{P} = \dot{C}_1 + (\bar{c}_1 + u_1\bar{a}_1 + v_1\bar{b}_1)w_1$$

$$\dot{P} = \dot{C}_2 + (\bar{c}_2 + u_2\bar{a}_2 + v_2\bar{b}_2)w_2$$

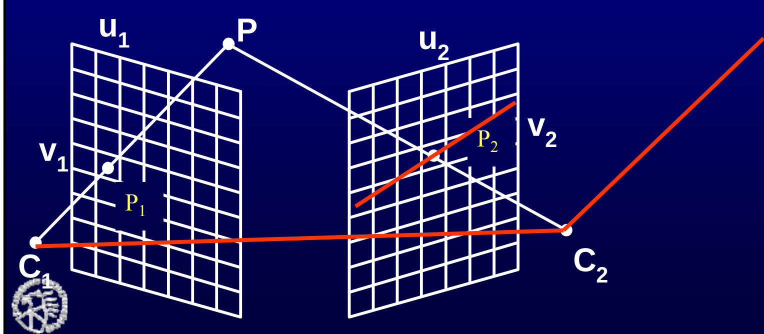


- unknowns are w_1 and w_2
- overconstrained system
  - the u_2v_2 coordinates of a point seen at u_1v_1 are constrained to an *epipolar* line

82

Epipolar line

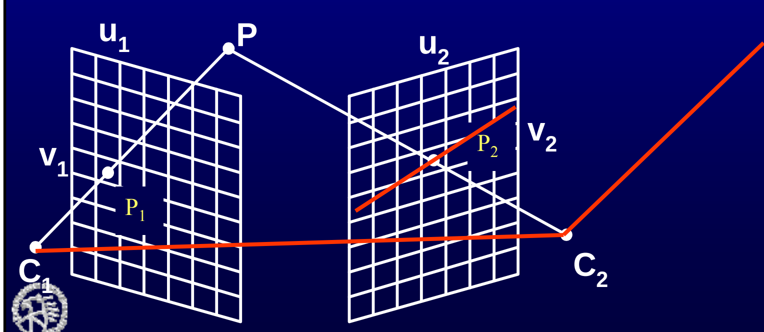
- C_1, C_2, P_1 define a plane
- P_2 will be on that plane
- P_2 is also on the image plane $_2$
- So P_2 will be on the line defined by the two planes' intersection



83

Search for correspondences on epipolar line

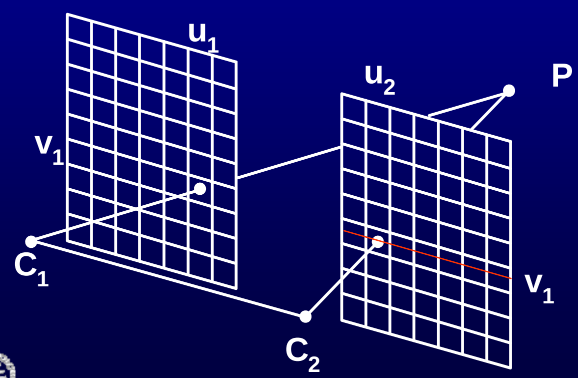
- Reduces the dimensionality of the search space
- Walk on epipolar segment rather than search in entire image



84

Parallel views

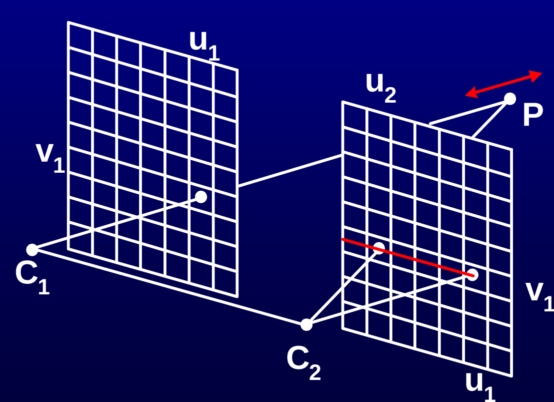
- Preferred stereo configuration
  - epipolar lines are horizontal, easy to search



85

Parallel views

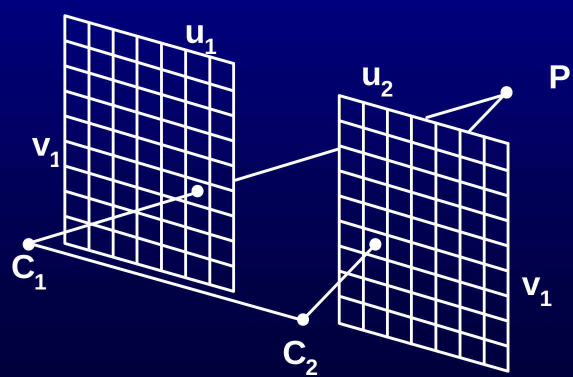
- Limit search to epipolar segment
  - from $u_2 = u_1$ (P is infinitely far away) to 0 (P is close)



86

Depth precision analysis

- $1/z$ linear with disparity $(u_1 - u_2)$
- better depth resolution for nearby objects
- important to determine correspondences with subpixel accuracy



87

Overview

- Depth from stereo
- Depth from structured light
- Depth from focus / defocus
- Laser rangefinders

88

Overview

- Depth from stereo
- Depth from structured light
- Depth from focus / defocus
- Laser rangefinders

89

Depth from stereo problem

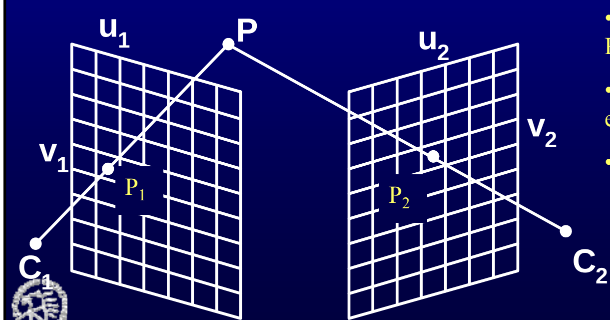
- Correspondences are difficult to find
- Structured light approach
  - replace one camera with projector
  - project easily detectable patterns
  - establishing correspondences becomes a lot easier

90

Depth from structured light

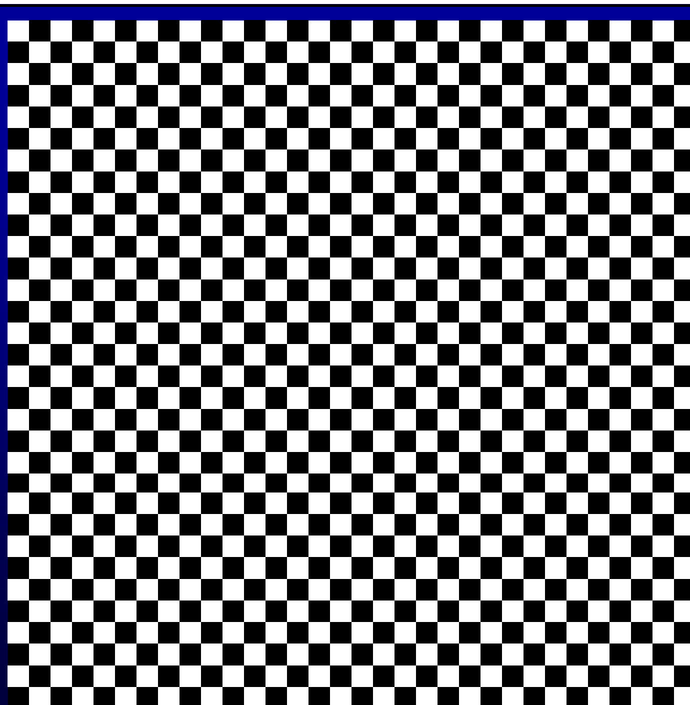
$$\dot{P} = \dot{C}_1 + (\bar{c}_1 + u_1\bar{a}_1 + v_1\bar{b}_1)w_1$$

$$\dot{P} = \dot{C}_2 + (\bar{c}_2 + u_2\bar{a}_2 + v_2\bar{b}_2)w_2$$



- C_1 is a projector
- Projects a pattern centered at u_1v_1
- Pattern center hits object scene at P
- Camera C_2 sees pattern at u_2v_2, easy to find
- 3D location of P is determined

91



93

Depth from structured light challenges

- Associated with using projectors
  - expensive, cannot be used outdoors, not portable
- Difficult to identify pattern
  - I found a corner, which corner is it?
- Invasive, change the color of the scene
  - one could use invisible light, IR

94

Overview

- Depth from stereo
- Depth from structured light
- Depth from focus / defocus
- Laser rangefinders

95

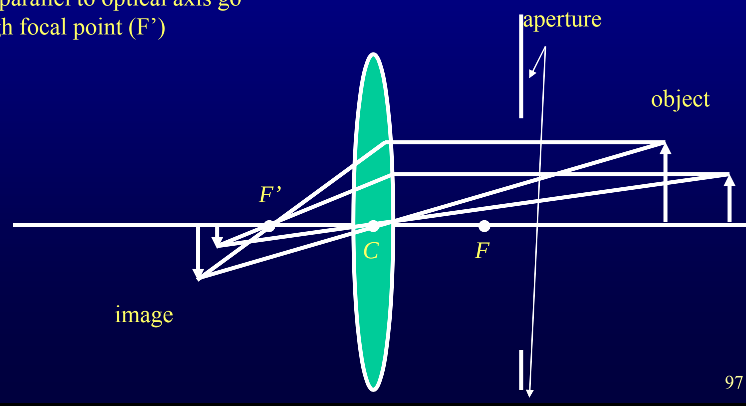
Overview

- Depth from stereo
- Depth from structured light
- Depth from focus / defocus
- Laser rangefinders

96

Depth of field

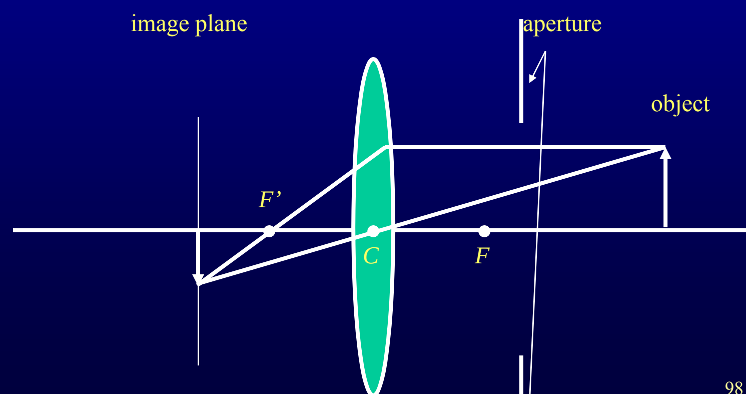
- Thin lenses
  - rays through lens center (C) do not change direction
  - rays parallel to optical axis go through focal point (F')



97

Depth of field

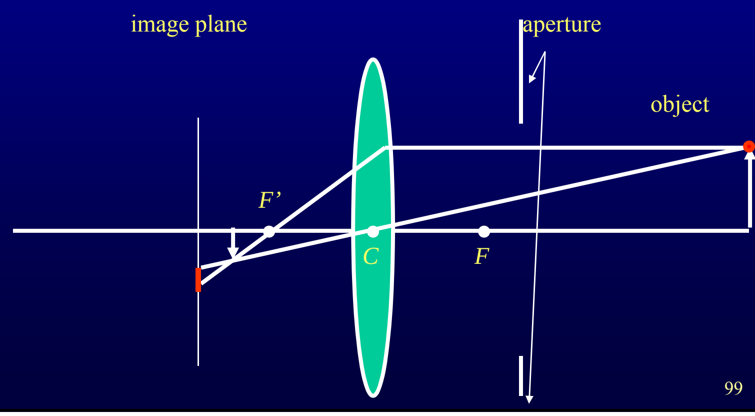
- For a given focal length, only objects that are at a certain depth are in focus



98

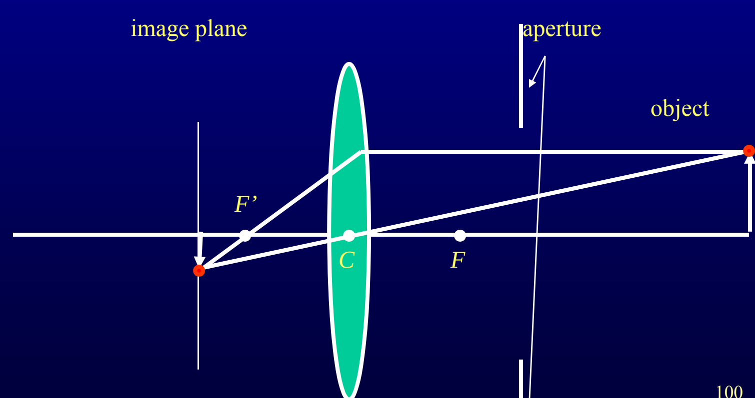
Out of focus

- When object at different depth
  - One point projects to several locations in the image
  - Out of focus, blurred image



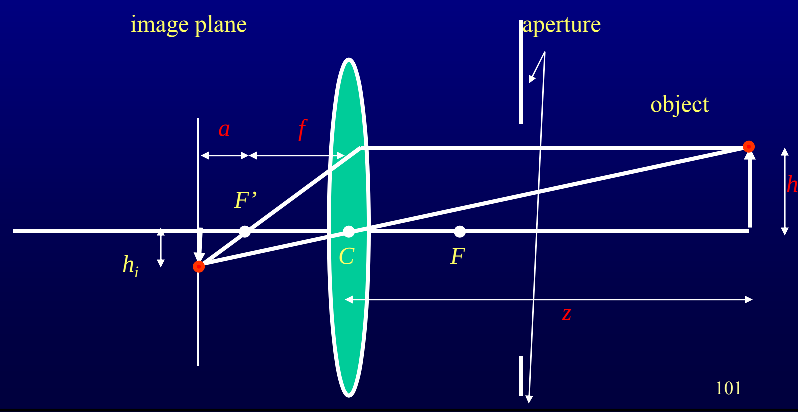
Focusing

- Move lens to focus for new depth
- Relationship between focus and depth can be exploited to extract depth



Determine z for points in focus

$$\frac{a}{f} = \frac{h_i}{h} = \frac{a+f}{z}$$



101

Depth from defocus

- Take images of a scene with various camera parameters
- Measuring defocus variation, infer range to objects
- Does not need to find the best focusing planes for the various objects
- Examples by Shree Nayar, Columbia U

102

Overview

- Depth from stereo
- Depth from structured light
- Depth from focus / defocus
- Laser rangefinders

103

Overview

- Depth from stereo
- Depth from structured light
- Depth from focus / defocus
- Laser rangefinders

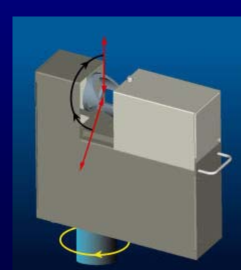
104

Laser range finders

- Send a laser beam to measure the distance
  - like RADAR, measures time of flight

DeltaSphere - depth&color acquisition device

- Lars Nyland *et al.*



courtesy 3rd Tech Inc.

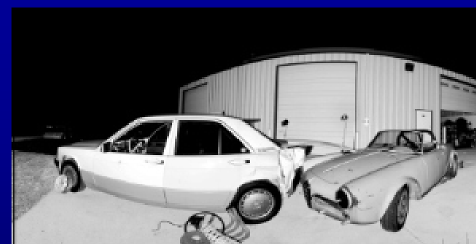
- 300^o x 300^o panorama
- this is the reflected light

107

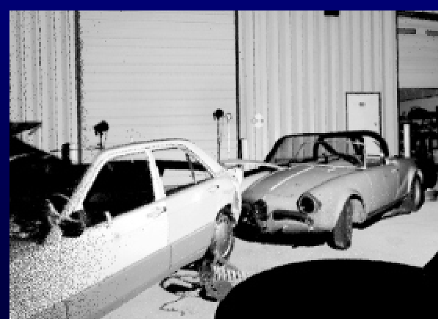
- 300^o x 300^o panorama
- this is the range light

108

spherical range panoramas



courtesy 3rd Tech Inc.

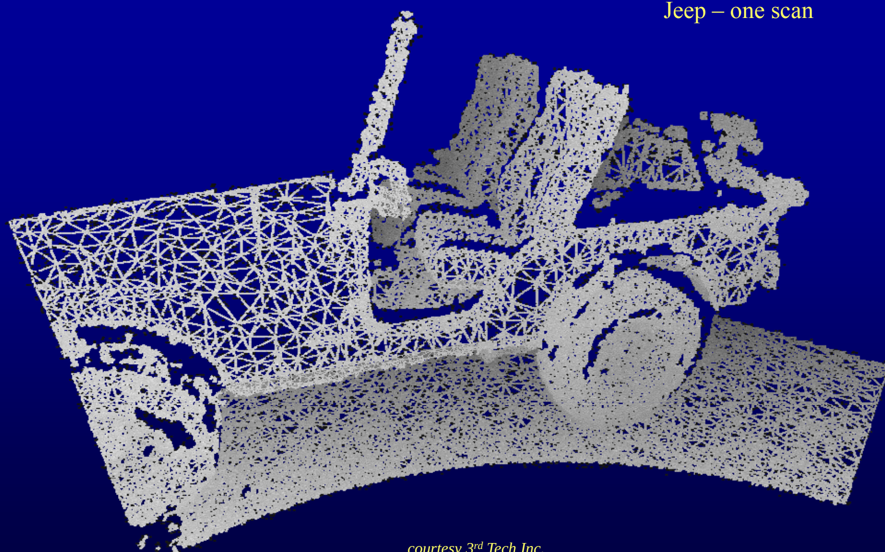


planar re-projection

Jeep – one scan

courtesy 3rd Tech Inc.

Jeep – one scan



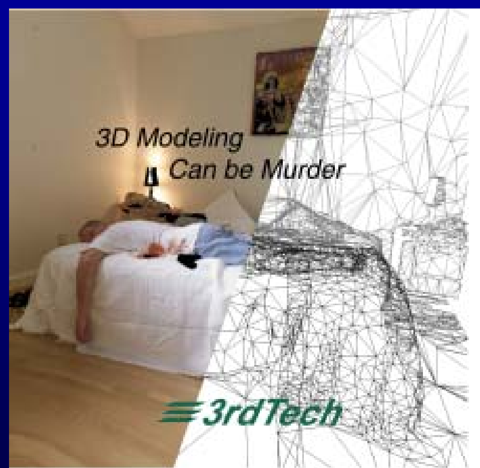
courtesy 3rd Tech Inc.

111

Complete Jeep model

courtesy 3rd Tech Inc.

112



113