Chapter 14
Image-based Rendering

Presenter: 胡慶宇

Phone: 0916120650

Mail: xerice58995@gmail.com

- 14.1 View Interpolation (Slide 4)
  - 14.1.1 View-dependent Texture Maps
  - 14.1.2 *Application*: Photo Tourism
- 14.2 Layered Depth Images (Slide 14)
  - 14.2.1 Impostors, Sprites, and Layers
  - 14.2.2 *Application*: 3D Photography
- 14.3 Light Fields and Lumigraphs (Slide 24)
  - 14.3.1 Lumigraph
  - 14.3.2 Surface Light Fields
  - 14.3.3 *Application*: Concentric Mosaics
- 14.4 Environment Mattes (Slide 30)
  - 14.4.1 Higher-dimensional Light Fields
  - 14.4.2 The Modeling to Rendering Continuum
- 14.5 Video-based Rendering (Slide 35)
  - 14.5.1 Video-based Animation
  - 14.5.2 Video Textures
  - 14.5.3 *Application*: Animating Pictures
  - 14.5.4 3D and Free-viewpoint Video
  - 14.5.5 *Application*: Video-based Walkthroughs
- 14.6 Neural Rendering (Slide 46)

Chapter 14
Image-based Rendering

- 14.1 View Interpolation (視像內插)
- 14.2 Layered Depth Images (分層深度影像)
- 14.3 Light Fields and Lumigraphs (光場與光度計)
- 14.4 Environment mattes (環境遮罩)
- 14.5 Video-based Rendering (基於視訊的渲染)
- 14.6 Neural Rendering (神經渲染)

14.1 View Interpolation　(視像内插)

- 14.1.1 View-dependent Texture Maps
- 14.1.2 *Application*: Photo Tourism



(a) (b) (c) (d)

(a) Holes or empty areas from one source image (shown in blue).

(b) holes after combining two widely spaced images. (two cameras far apart)

(c) holes after combining two closely spaced images. (two cameras near position)

(d) after interpolation (hole filling).

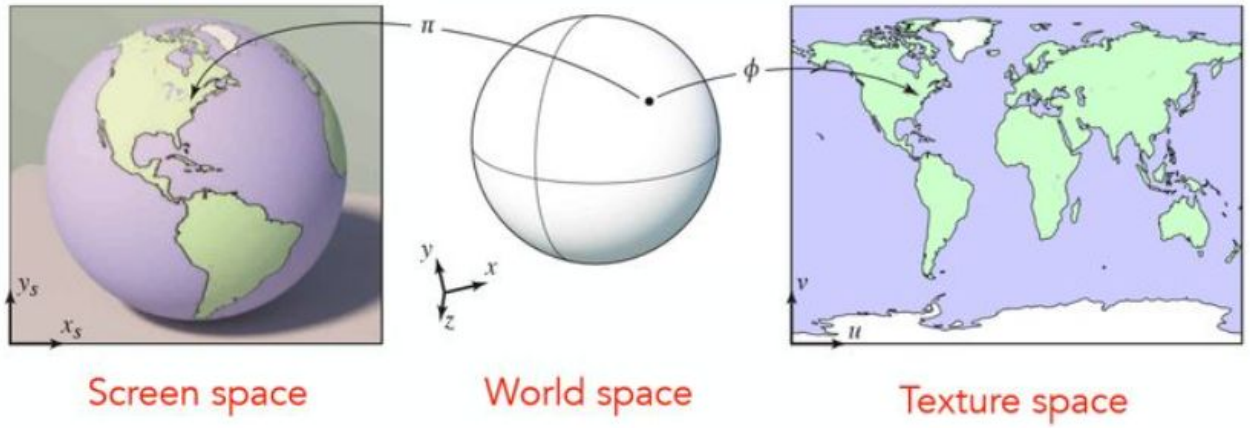
Video Example(0:35~2:05):

https://www.youtube.com/watch?v=uqKTbyNoaxE

14.1.1 View-dependent Texture Maps

View-dependent + Texture Maps = View-dependent texture maps

- **View-dependent** : current view or camera position (What you see will change based on where you are looking from)
- **Texture Maps (Texture Mapping)** : Mapping relationship from 3D world space to 2D texture

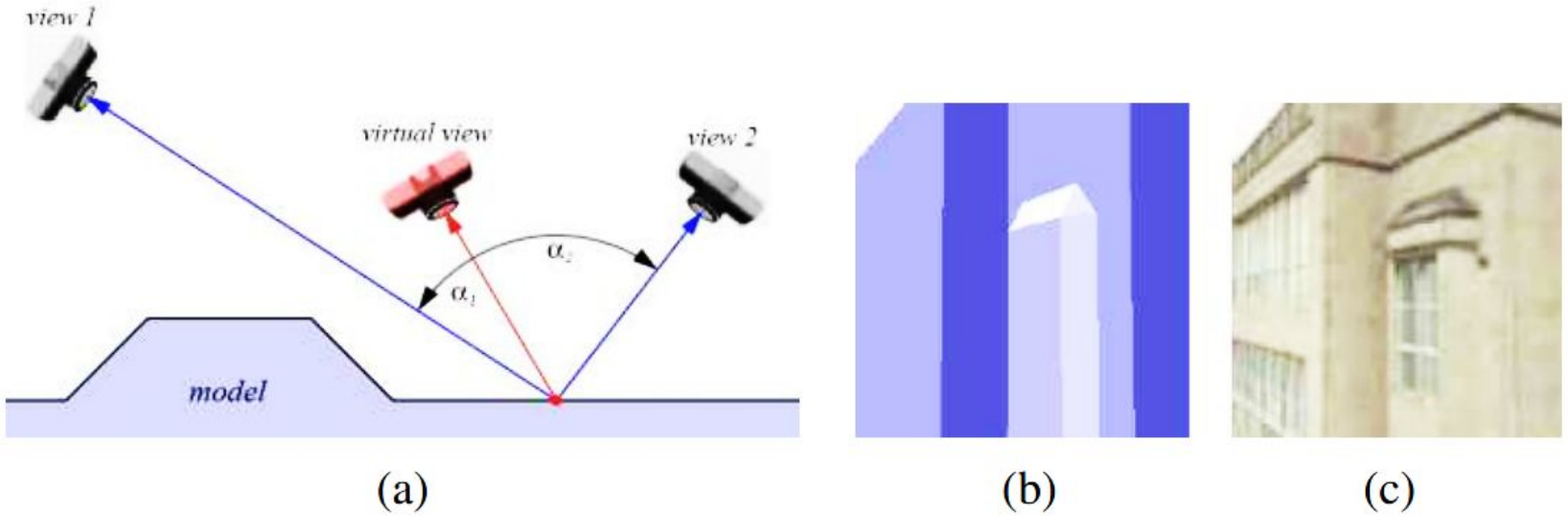


- Hence, **View-dependent Texture Maps** : different images are used as texture map sources depending on the virtual camera's current position

Video Example(2:30~2:40):

https://www.youtube.com/watch?v=uqKTbyNoaxE

- View-dependent Texture Mapping..



(a) The weighting given to each input view depends on the relative angles between the novel (virtual) view and the original views.

(b) simplified 3D model geometry.

(c) with view-dependent texture mapping, the geometry appears to have more detail (recessed windows).

Video Example(2:15~2:45):

https://www.youtube.com/watch?v=KFT4uaAH8_g

14.1.2 *Application* : Photo Tourism

(a) A 3D overview of the scene, with translucent washes and lines painted onto the planar impostors.

(b) Once the user has selected a region of interest, a set of related thumbnails is displayed along the bottom.

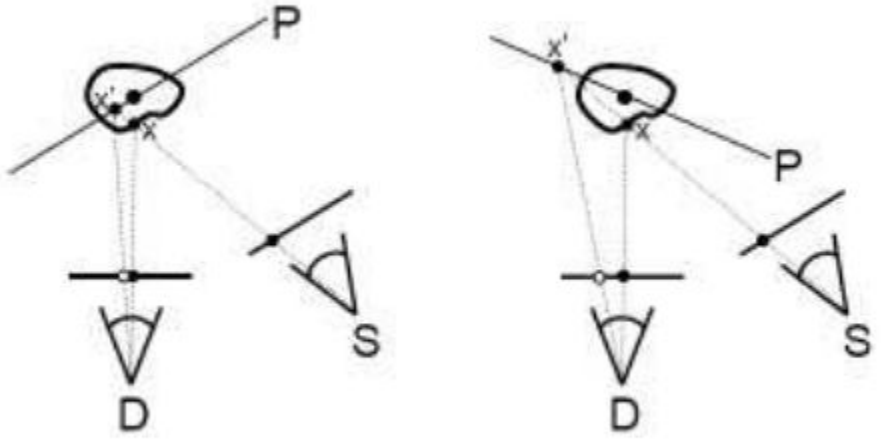
(c) Planar proxy selection for optimal stabilization.



(a)



(b)



(c)

Video Example(0:20~2:05):

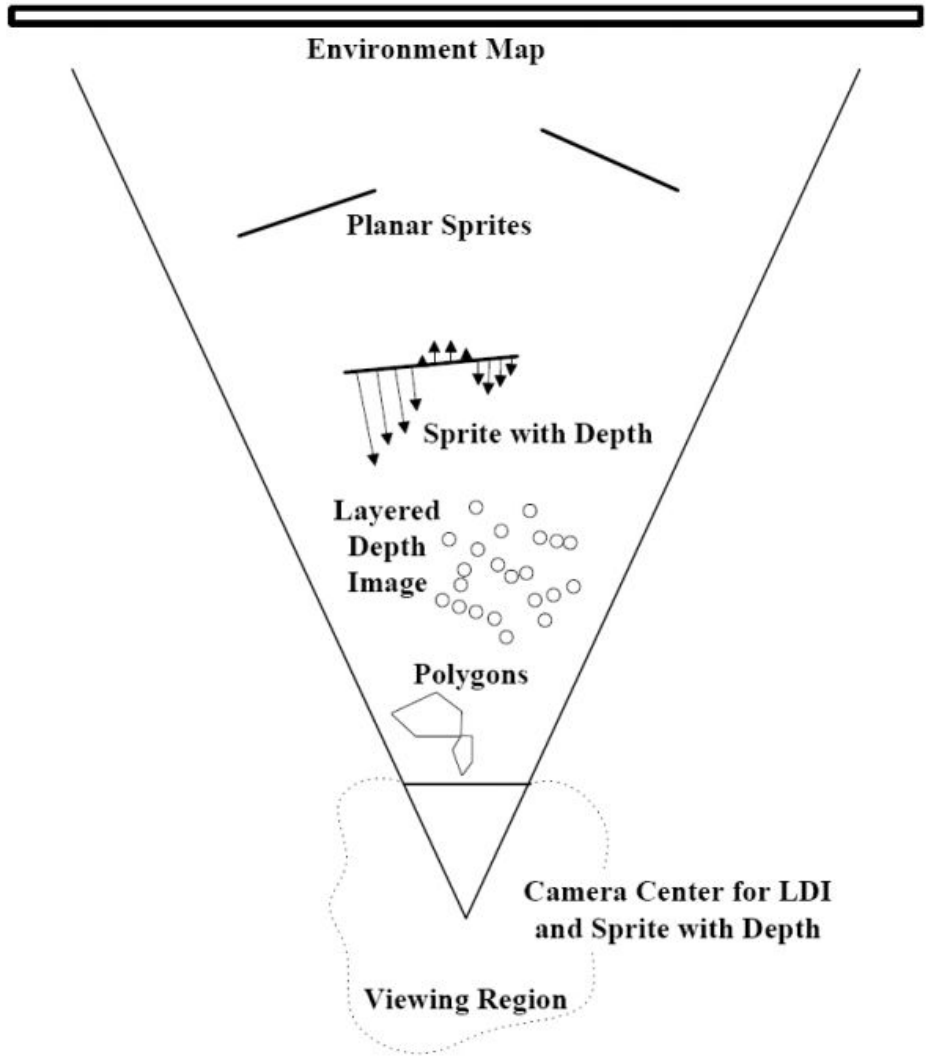
https://www.youtube.com/watch?v=lgBQCoEfiMs

Fun Time

- 14.1 View Interpolation
  - 14.1.1 View-dependent Texture Maps
  - 14.1.2 *Application*: Photo Tourism
- 14.2 Layered Depth Images
  - 14.2.1 Impostors, Sprites, and Layers
  - 14.2.2 *Application*: 3D Photography
- 14.3 Light Fields and Lumigraphs
  - 14.3.1 Lumigraph
  - 14.3.2 Surface Light Fields
  - 14.3.3 *Application*: Concentric Mosaics
- 14.4 Environment mattes
  - 14.4.1 Higher-dimensional Light Fields
  - 14.4.2 The Modeling to Rendering Continuum
- 14.5 Video-based Rendering
  - 14.5.1 Video-based Animation
  - 14.5.2 Video Textures
  - 14.5.3 *Application*: Animating Pictures
  - 14.5.4 3D and Free-viewpoint Video
  - 14.5.5 *Application*: Video-based Walkthroughs
- 14.6 Neural Rendering

14.2 Layered Depth Images

- 14.2.1 Impostors, Sprites, and Layers
- 14.2.2 *Application*: 3D Photography



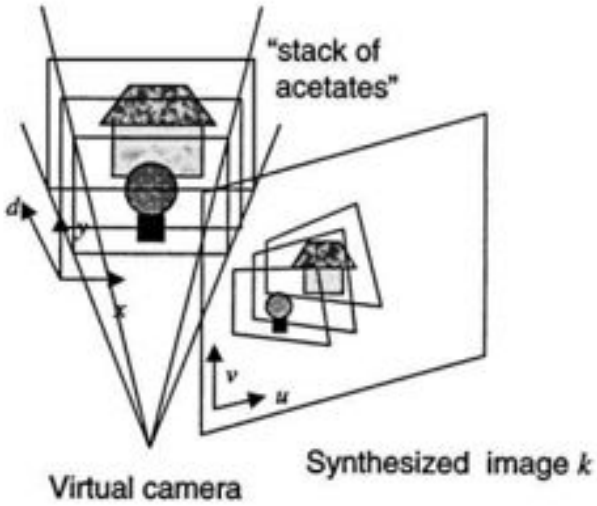
14.2.1 Impostors, Sprites, and Layers



Sprites

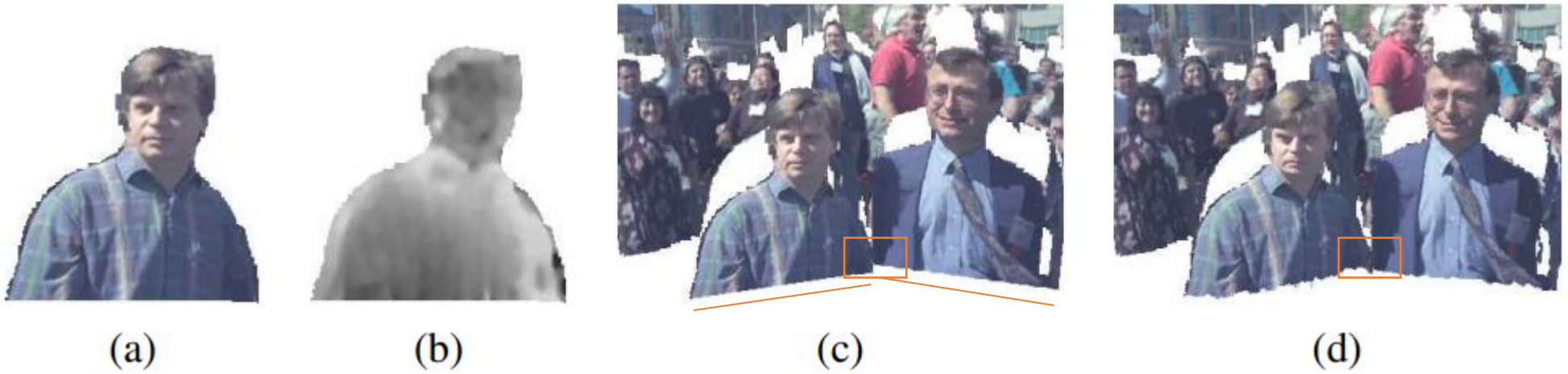


Impostor



Layers

14.2.1 Impostors, Sprites, and Layers



(a) alpha-matted color sprite

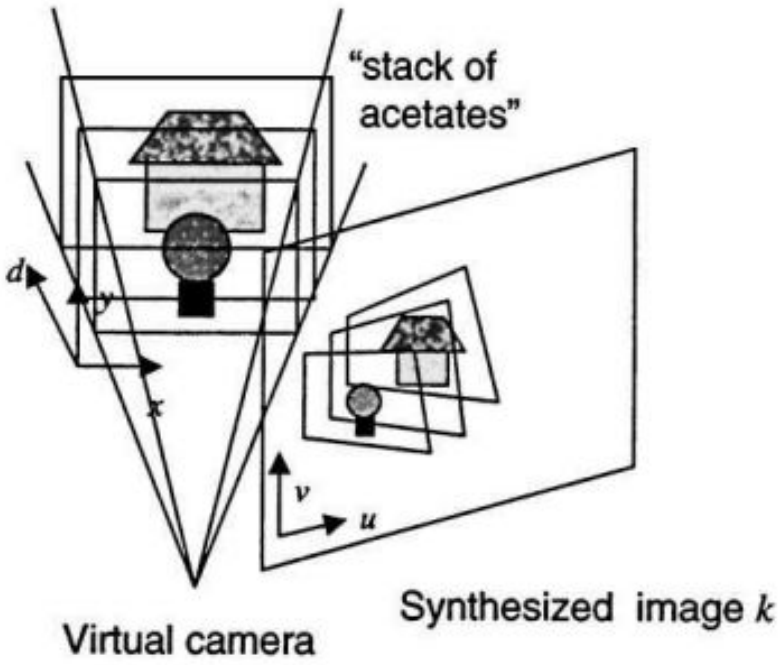
(b) corresponding relative depth or parallax (相對深度跟視差)

(c) rendering without relative depth (沒有做相關深度資訊的渲染)

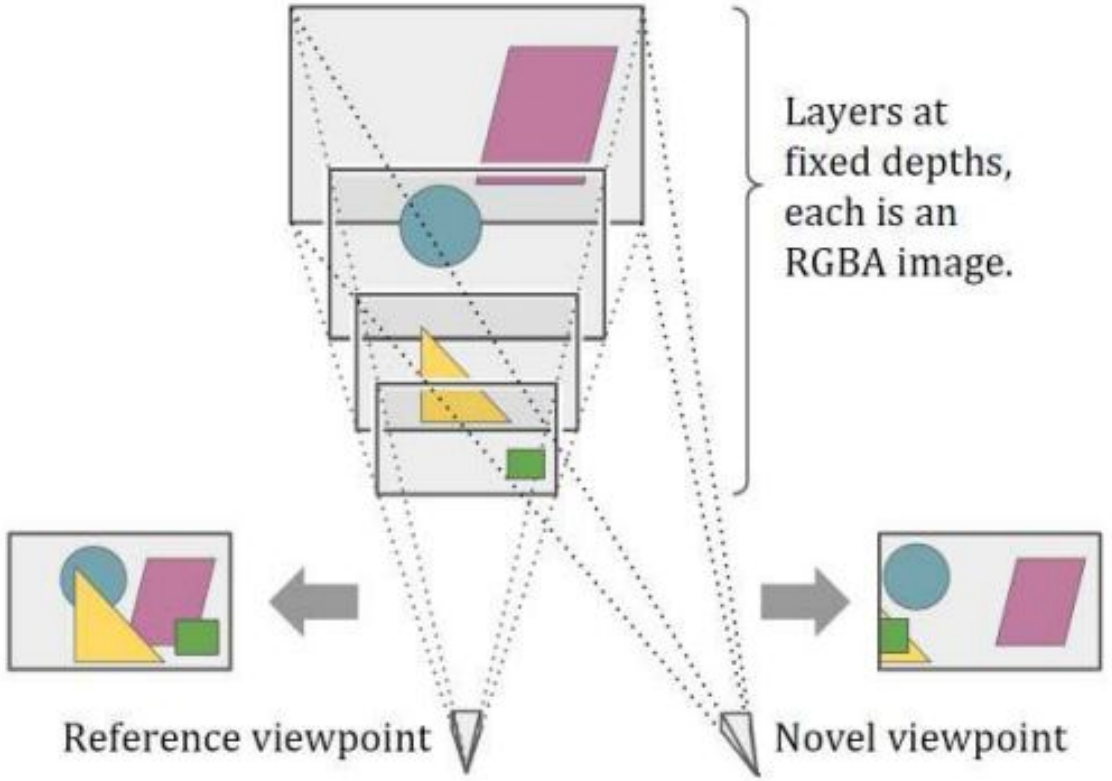
(d) rendering with depth (有做相關深度資訊的渲染) -- note the curved object boundaries.

14.2.1 Impostors, Sprites, and Layers

"Acetates" = transparent plastic sheets



(a)



(b)

Novel viewpoint: new or different ways of looking about something.

14.2.1 Impostors, Sprites, and Layers

- MPI (Multi-Plane Images)



Video Example(0:00~1:00):

https://www.youtube.com/watch?v=gnZT34DYwyE

14.2.1 Impostors, Sprites, and Layers

- Two-layer Decomposition

  Separates an image into a reflected layer and a transmitted (real scene) layer. With MPI, each layer has its own depth map, so they can be warped (transformed) to match a new camera angle.



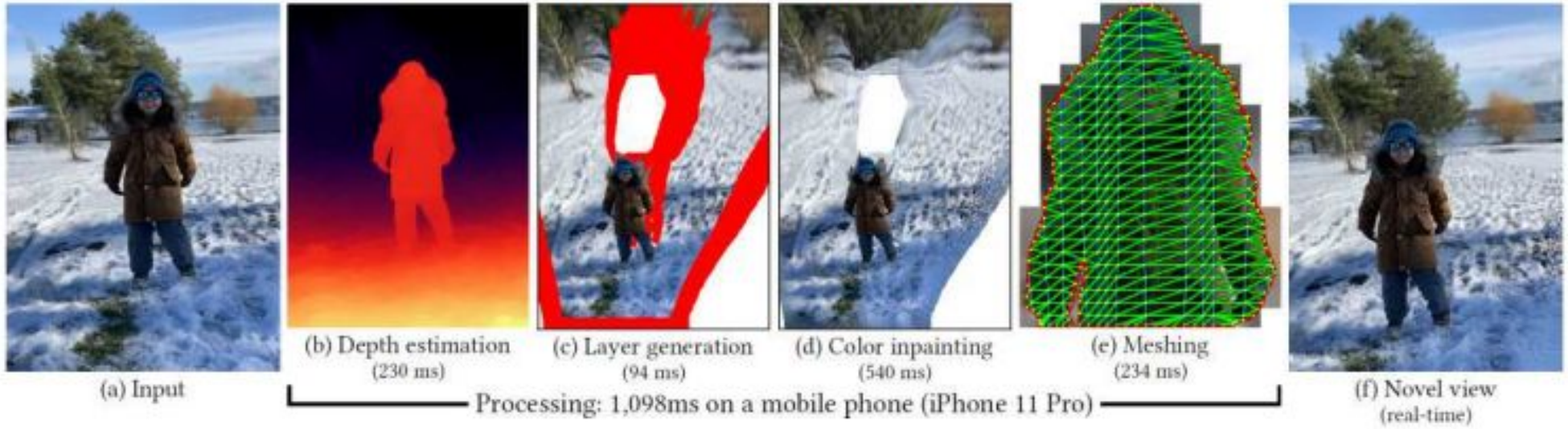
(a) Glass case



(b) Table

14.2.2 *Application* : 3D Photography

- One Shot 3D Photography
  - Starts with a single photograph
  - Performs monocular depth estimation, layer construction and inpainting
  - Mesh and atlas generation
  - Enabling phone-based reconstruction and interactive viewing



Fun Time

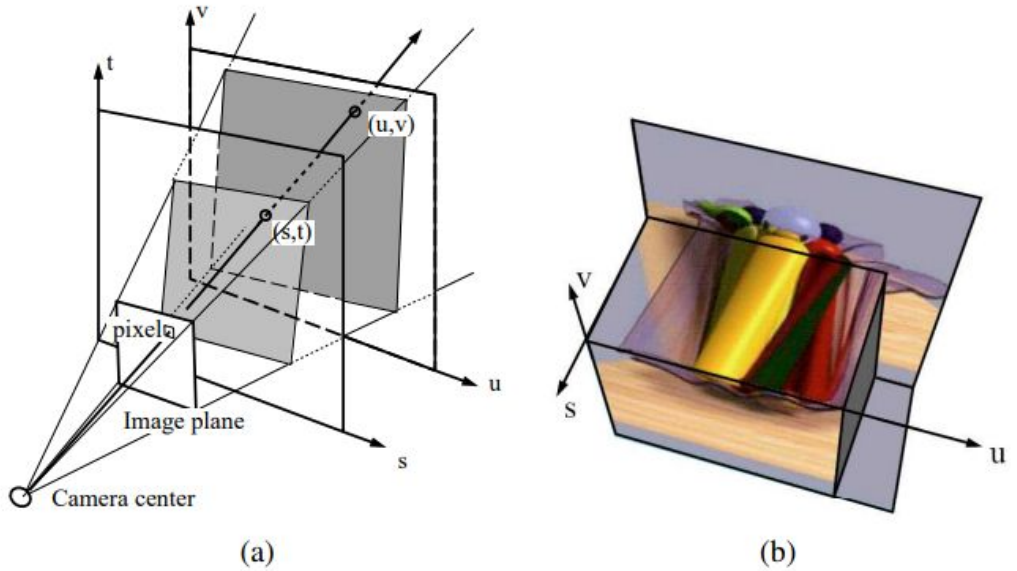
- 14.1 View Interpolation
  - 14.1.1 View-dependent Texture Maps
  - 14.1.2 *Application*: Photo Tourism
- 14.2 Layered Depth Images
  - 14.2.1 Impostors, Sprites, and Layers
  - 14.2.2 *Application*: 3D Photography
- 14.3 Light Fields and Lumigraphs
  - 14.3.1 Lumigraph
  - 14.3.2 Surface Light Fields
  - 14.3.3 *Application*: Concentric Mosaics
- 14.4 Environment mattes
  - 14.4.1 Higher-dimensional Light Fields
  - 14.4.2 The Modeling to Rendering Continuum
- 14.5 Video-based Rendering
  - 14.5.1 Video-based Animation
  - 14.5.2 Video Textures
  - 14.5.3 *Application*: Animating Pictures
  - 14.5.4 3D and Free-viewpoint Video
  - 14.5.5 *Application*: Video-based Walkthroughs
- 14.6 Neural Rendering

14.3 Light Fields and Lumigraphs

- 14.3.1 Lumigraph (光度計)
- 14.3.2 Surface Light Fields (表面光場)
- 14.3.3 *Application*: Concentric Mosaics (應用:同心(圓)馬賽克/拼圖)

14.3.1 Lumigraph

- Taking the cross-product of the three-dimensional space of camera positions with the 2D space of spherical images, we obtain the 5D plenoptic function.
- When no light dispersion in the scene, we can reduce the 5D plenoptic function to the 4D light field.
- Put two planes in the 3D scene roughly bounding the area of interest. Thus all ray terminating at camera can be represented as *(s, t)* and *(u, v).*

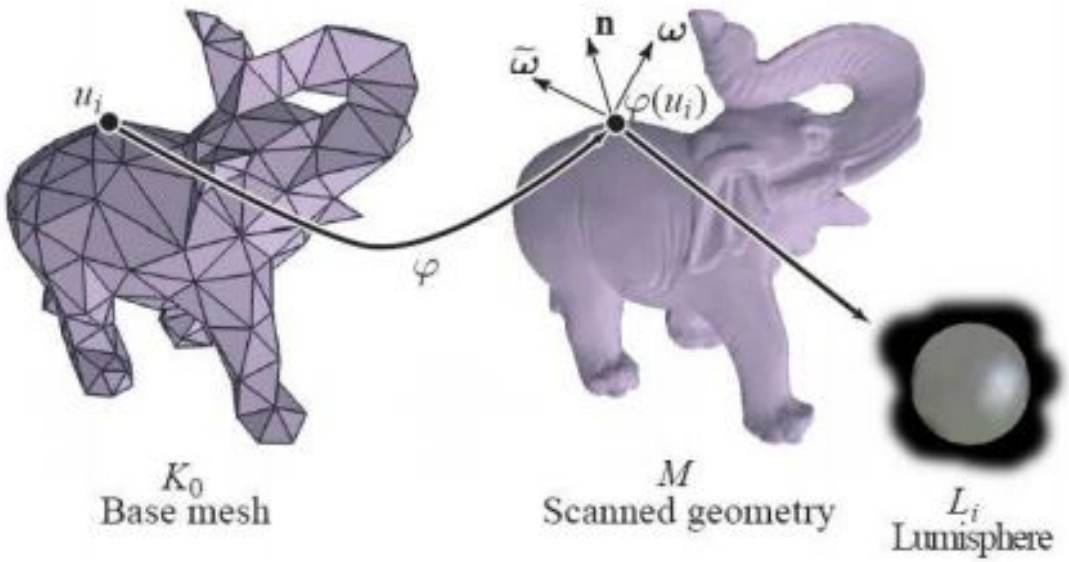


14.3.2 Surface Light Fields

- Good representation to add realism to scanned 3D object models by **modeling their specular properties.**
- Avoiding the **“cardboard” (matte)** appearance of such models when their reflections are ignored.



(a)



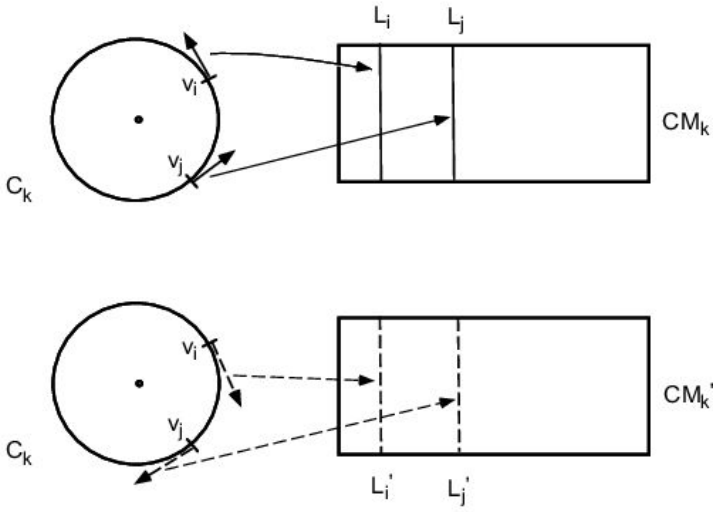
(b)

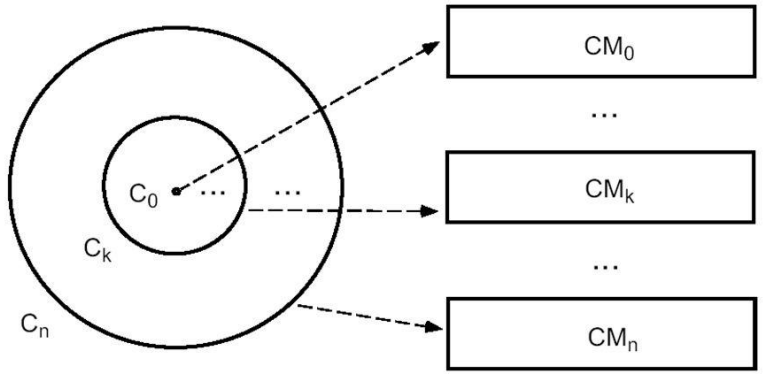
Video Example(0:00~1:50):

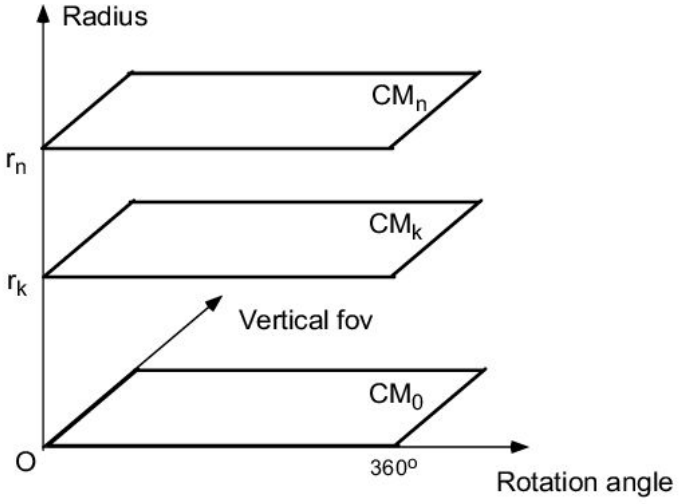
https://www.youtube.com/watch?v=OhJTjkukARc

14.3.3 *Application* : Concentric Mosaics..

- Concentric Mosaics are captured by moving the camera on a (roughly) circular arc.
- Holding a camera at arm's length while rotating your body around a fixed axis.
- Also called **Layered Depth Panorama**.







- 14.1 View Interpolation
  - 14.1.1 View-dependent Texture Maps
  - 14.1.2 *Application*: Photo Tourism
- 14.2 Layered Depth Images
  - 14.2.1 Impostors, Sprites, and Layers
  - 14.2.2 *Application*: 3D Photography
- 14.3 Light Fields and Lumigraphs
  - 14.3.1 Lumigraph
  - 14.3.2 Surface Light Fields
  - 14.3.3 *Application*: Concentric Mosaics
- 14.4 Environment mattes
  - 14.4.1 Higher-dimensional Light Fields
  - 14.4.2 The Modeling to Rendering Continuum
- 14.5 Video-based Rendering
  - 14.5.1 Video-based Animation
  - 14.5.2 Video Textures
  - 14.5.3 *Application*: Animating Pictures
  - 14.5.4 3D and Free-viewpoint Video
  - 14.5.5 *Application*: Video-based Walkthroughs
- 14.6 Neural Rendering

14.4 Environment Mattes

$$C_i = \alpha_i F_i + (1 - \alpha_i) B_i \qquad (1)$$

$$I_i = R_i \int A_i(x) B(x) dx \qquad (2)$$

$$I_i = \sum_j R_{ij} \int G_{ij}(x) B(x) dx \qquad (3)$$

$$G_{ij}(x) = G_{2D}(x;\, c_{ij}, \sigma_{ij}, \theta_{ij}) \qquad (4)$$



(a) (b) (c) (d)

14.4 Environment Mattes

$$C_i = \alpha_i F_i + (1 - \alpha_i) B_i \qquad (1)$$

$$I_i = R_i \int A_i(x) B(x) dx \qquad (2)$$

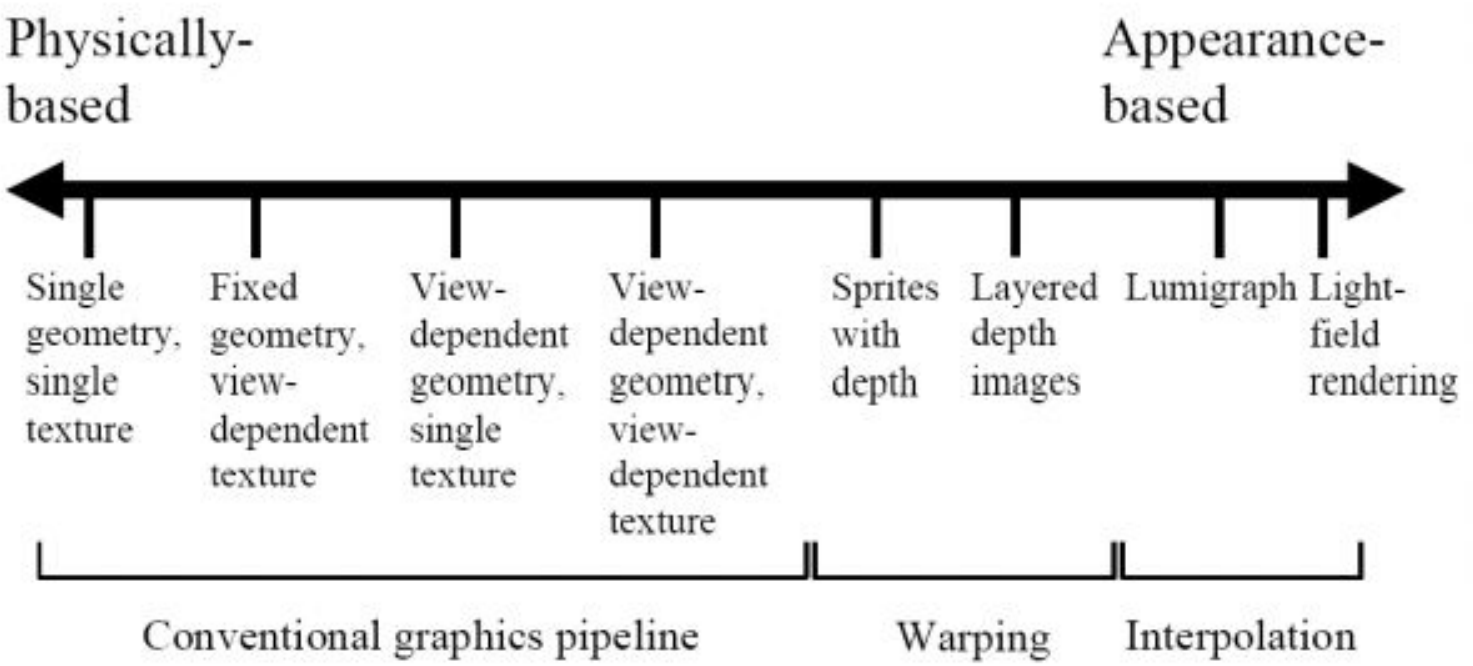
$$I_i = \sum_j R_{ij} \int G_{ij}(x) B(x) dx \qquad (3)$$

$$G_{ij}(x) = G_{2D}(x;\, c_{ij}, \sigma_{ij}, \theta_{ij}) \qquad (4)$$

(αF, α) : premultiplied colors and opacities
Bi : new background
***I*i** : Intensity
Ai : the rectangular area of support for that pixel
Ri : the colored reflectance or transmittance (for colored glossy surfaces or glass)
Gij (x) : 2D Gaussian
Cij : center
***σ*ij** : unrotated widths
Θij : orientation

14.4.2 The Modeling to Rendering Continuum

- The left: use more detailed geometry and simpler image representations.
- The right: use more images and less geometry.



Fun Time

- 14.1 View Interpolation
  - 14.1.1 View-dependent Texture Maps
  - 14.1.2 *Application*: Photo Tourism
- 14.2 Layered Depth Images
  - 14.2.1 Impostors, Sprites, and Layers
  - 14.2.2 *Application*: 3D Photography
- 14.3 Light Fields and Lumigraphs
  - 14.3.1 Lumigraph
  - 14.3.2 Surface Light Fields
  - 14.3.3 *Application*: Concentric Mosaics
- 14.4 Environment mattes
  - 14.4.1 Higher-dimensional Light Fields
  - 14.4.2 The Modeling to Rendering Continuum
- 14.5 Video-based Rendering
  - 14.5.1 Video-based Animation
  - 14.5.2 Video Textures
  - 14.5.3 *Application*: Animating Pictures
  - 14.5.4 3D and Free-viewpoint Video
  - 14.5.5 *Application*: Video-based Walkthroughs
- 14.6 Neural Rendering

14.5 Video-based Rendering

- 14.5.1 Video-based Animation (視訊基礎動畫)
- 14.5.2 Video Textures (視訊材質)
- 14.5.3 *Application*: Animating Pictures（動畫圖片）
- 14.5.4 3D and Free-viewpoint Video（自由視角影片）
- 14.5.5 *Application*: Video-based Walkthroughs（視訊基礎虛擬漫遊）

14.5.1 Video-based Animation

- Frames from original video footage are rearranged to match them to novel spoken utterances.
  - e.g., for **movie dubbing**
- Similar to how *concatenative speech synthesis* systems work.
- Use speech recognition to extract phonemes from both the source video material and the novel audio stream. -> ***triphones*** **(triplets of phonemes)**



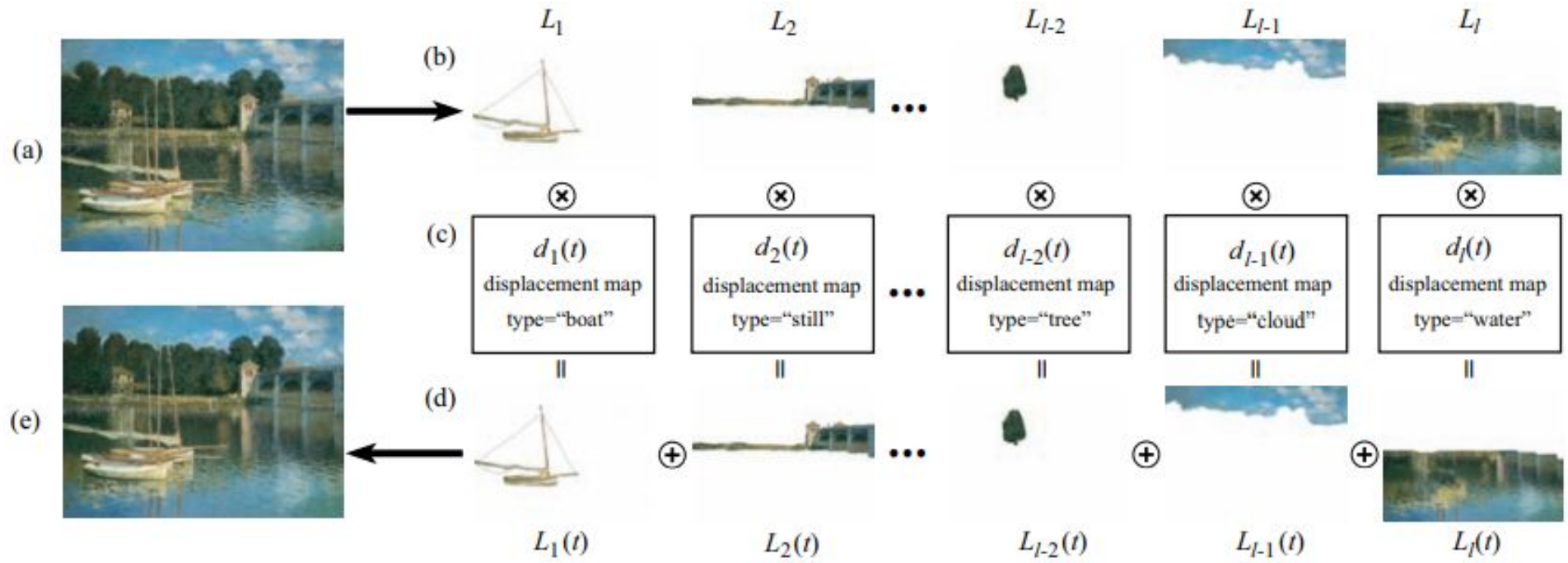
14.5.2 Video Textures

- A static image or a video clip has a discontinuity when it loops.
- A short video clip can be arbitrarily extended by re-arranging video frames while preserving visual continuity.
- The simplest approach: match frames by visual similarity (e.g., **L2 distance**) and jump between frames that appear similar. -> may have problem like (a).
- Constructed an *n*×*n* similarity matrix between all video frames (where *n* is the number of frames), a simple **Finite Impulse Response (FIR) filtering** of each match sequence can be used to emphasize subsequences that match well.



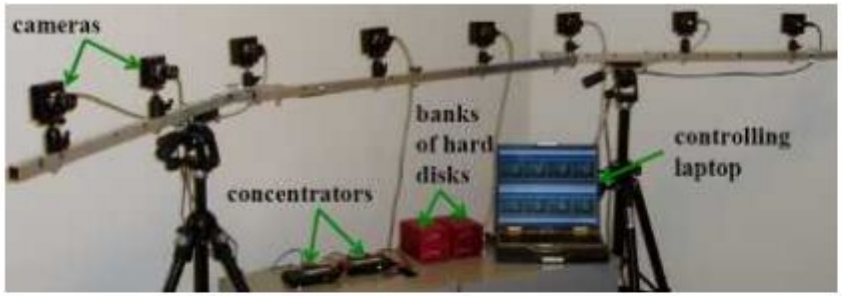
14.5.3 *Application* : Animating Pictures

- Animating still pictures

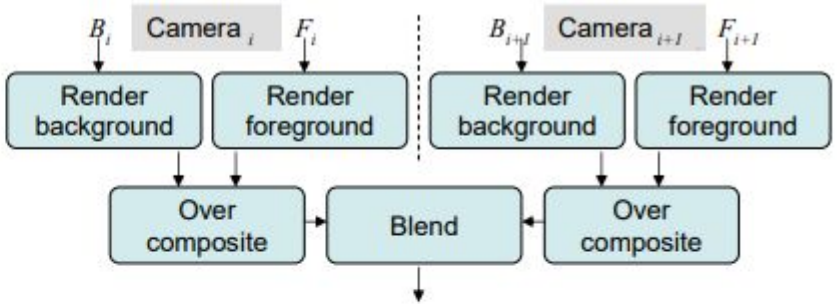


14.5.4 3D and Free-viewpoint Video

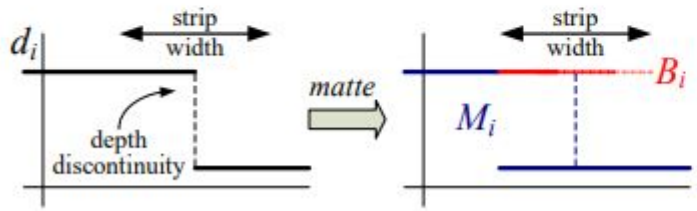
- Video View Interpolation



(a)



(b)



(c)



(d)

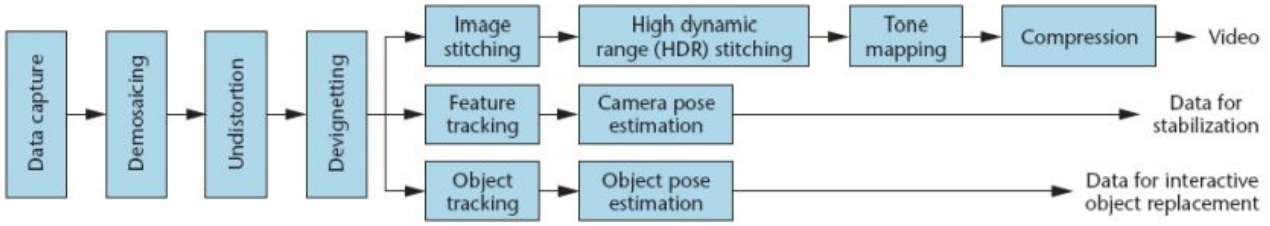


(e)



(f)

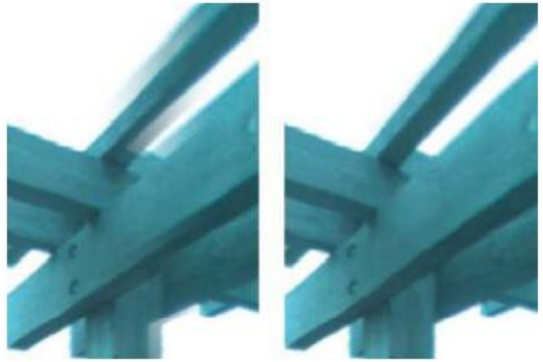
14.5.5 *Application* : Video-based Walkthroughs



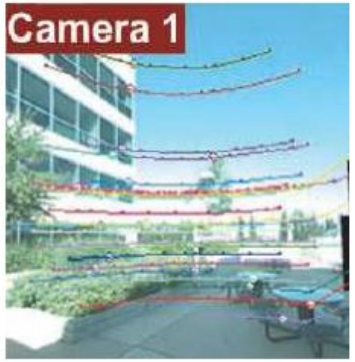
(a)



(b)



(c)



(d)



(e)



(f)



(g)

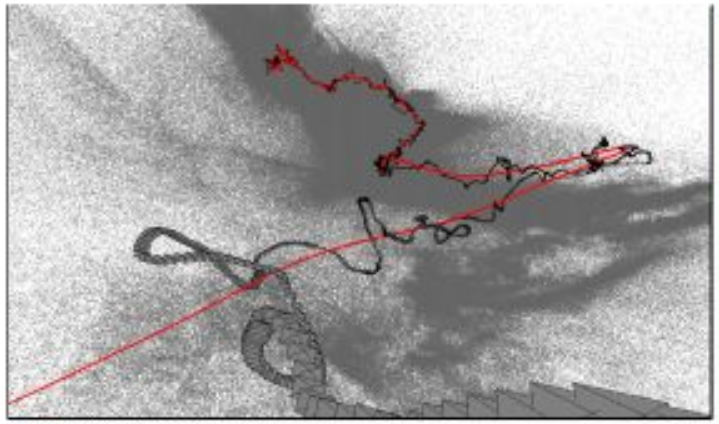
Video Example(0:00~1:50):

https://www.youtube.com/watch?v=lX9cq-RjMdk

https://www.youtube.com/watch?v=FQaKwYRouyI

14.5.5 *Application* : Video-based Walkthroughs

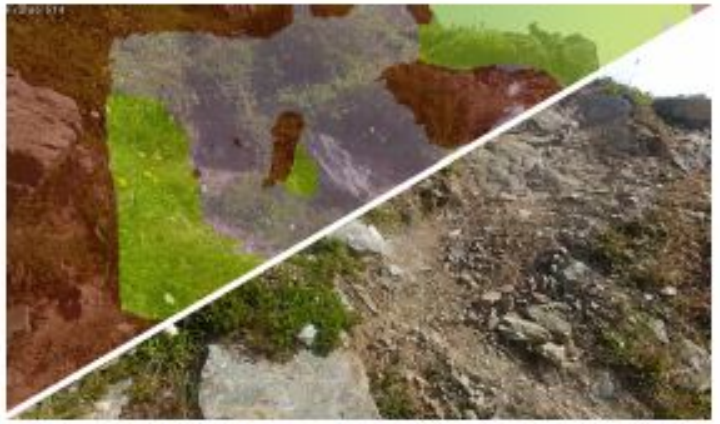
- First-person hyperlapse video creation



(a) Scene reconstruction



(b) Proxy geometry



(c) Stitched & blended

Video Example(0:28~1:35):

https://www.youtube.com/watch?v=sA4Za3Hv6ng

Fun Time

- 14.1 View Interpolation
  - 14.1.1 View-dependent Texture Maps
  - 14.1.2 *Application*: Photo Tourism
- 14.2 Layered Depth Images
  - 14.2.1 Impostors, Sprites, and Layers
  - 14.2.2 *Application*: 3D Photography
- 14.3 Light Fields and Lumigraphs
  - 14.3.1 Lumigraph
  - 14.3.2 Surface Light Fields
  - 14.3.3 *Application*: Concentric Mosaics
- 14.4 Environment mattes
  - 14.4.1 Higher-dimensional Light Fields
  - 14.4.2 The Modeling to Rendering Continuum
- 14.5 Video-based Rendering
  - 14.5.1 Video-based Animation
  - 14.5.2 Video Textures
  - 14.5.3 *Application*: Animating Pictures
  - 14.5.4 3D and Free-viewpoint Video
  - 14.5.5 *Application*: Video-based Walkthroughs
- 14.6 Neural Rendering

ACM: Association for Computing Machinery
IEEE: Institute of Electrical and Electronics Engineers

14.6 Neural Rendering

- **Texture-mapped meshes**

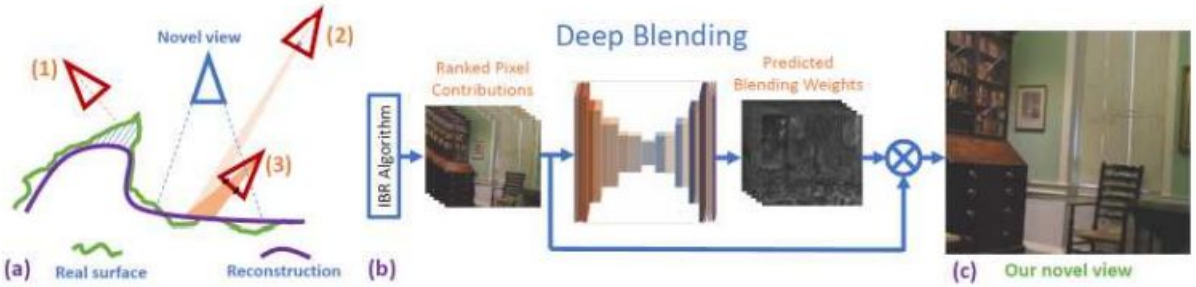
(a) deep blending of depth-warped source images © 2018 ACM

- **Depth images and layers**

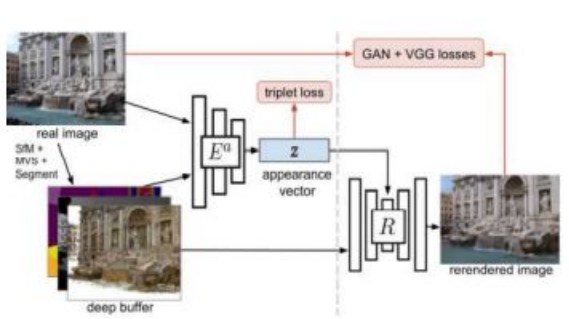
(b) neural re-rendering in the wild with controllable view and lighting © 2019 IEEE

(c) crowdsampling the plenoptic function with a deep MPI © 2020 Springer

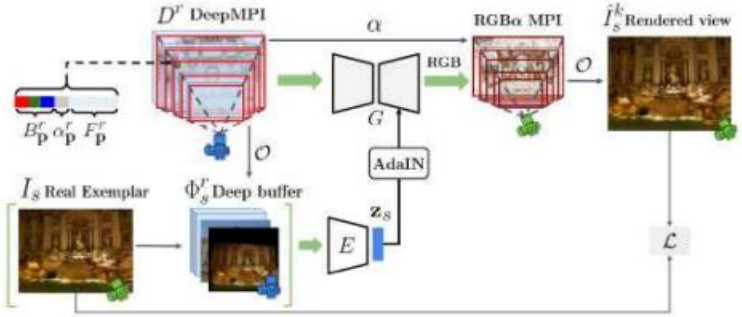
(d) SynSin: novel view synthesis from a single image © 2020 IEEE.



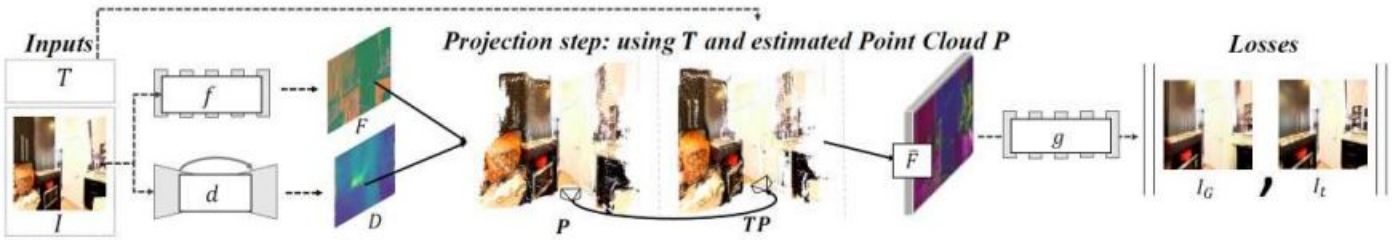
(a)



(b)



(c)



(d)

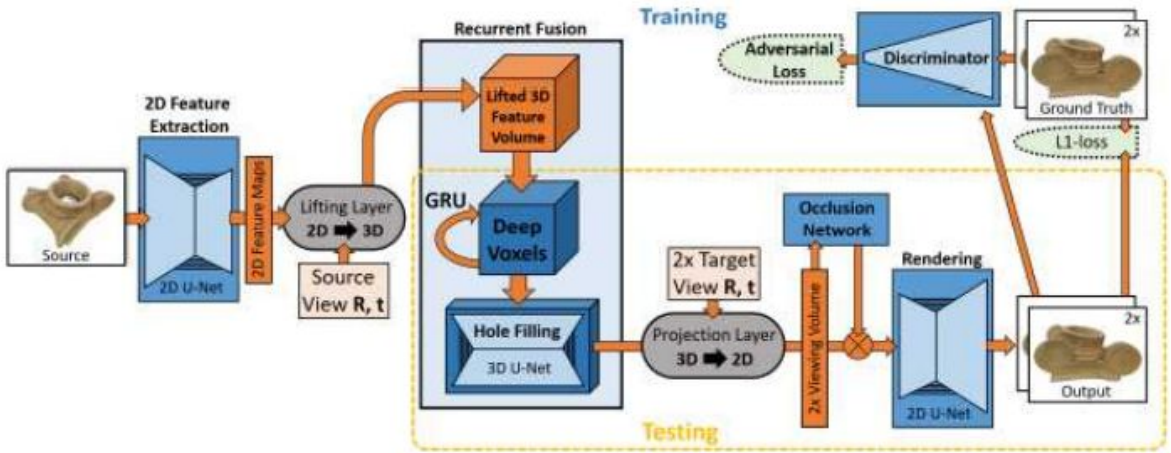
ACM: Association for Computing Machinery
IEEE: Institute of Electrical and Electronics Engineers

14.6 Neural Rendering

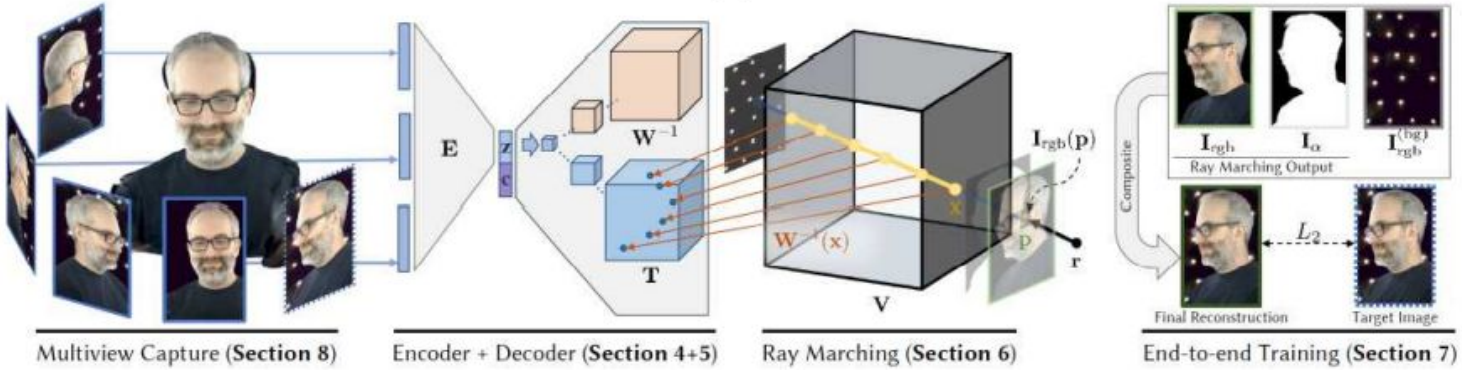
- **Voxel representations**

(a) DeepVoxels © 2019 IEEE

(b) Neural Volumes © 2019 ACM.



(a)



(b)

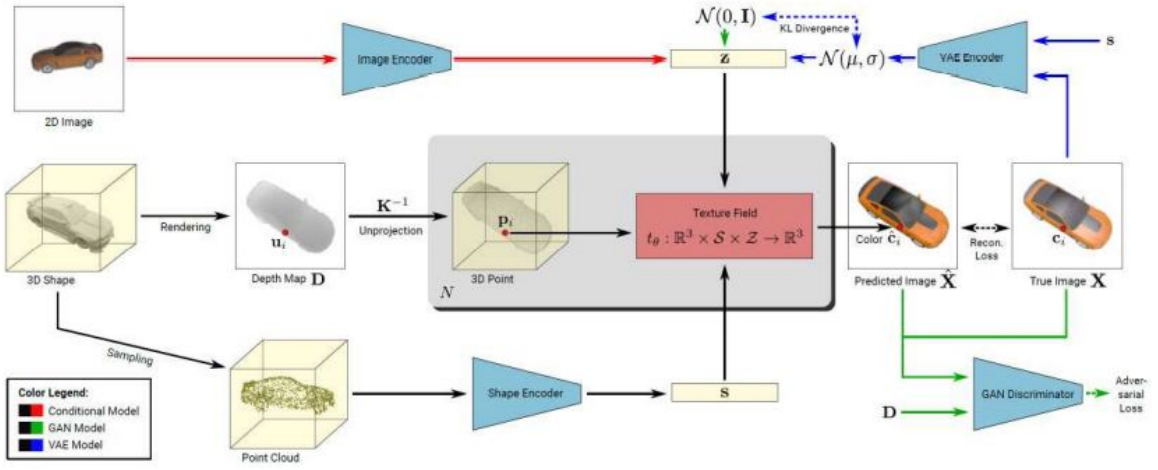
MLP: Multi-Layer Perceptron

14.6 Neural Rendering

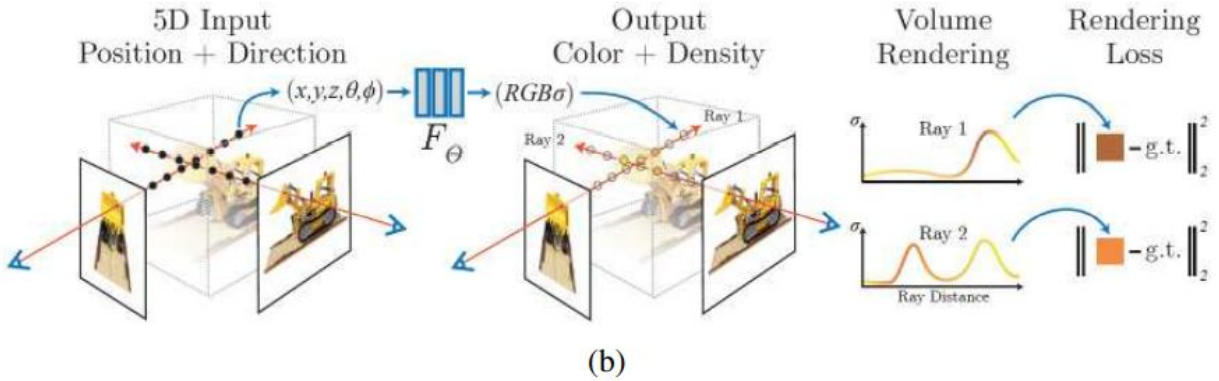
- **Implicit functions implemented using MLPs.**

(a) Texture Fields © 2019 IEEE

(b) Neural Radiance Fields © 2020 Springer



(a)



(b)

ACM: Association for Computing Machinery

IEEE: Institute of Electrical and Electronics Engineers

Fun Time

References

- Richard Szeliski , “Computer Vision: Algorithms and Applications 2nd Edition, ”

  https://szeliski.org/Book/, 2021.