Photographing Long Scenes with Multi-viewpoint Panoramas

A. Agarwala, M. Agrawala, M. Cohen, D. Salesin, R. Szeliski

Presenter: Stacy Hsueh

Discussant: Vasily Volkov

Motivation

- Want an image that shows an elongated scene
- Single image not sufficient
  - Small part of street
  - Wider field of view: distortions towards the edges of image
  - Far away: loss of details (perspective depth cues)
- Capture images from different points of view
- Needs a way to stitch images together
- Should resemble what humans would see

Some definitions

- Multi-viewpoint
  - Many single viewpoint photos rendered in one picture naturally
- Long scenes
  - Back of river, aisle of grocery store

Strip Panorama

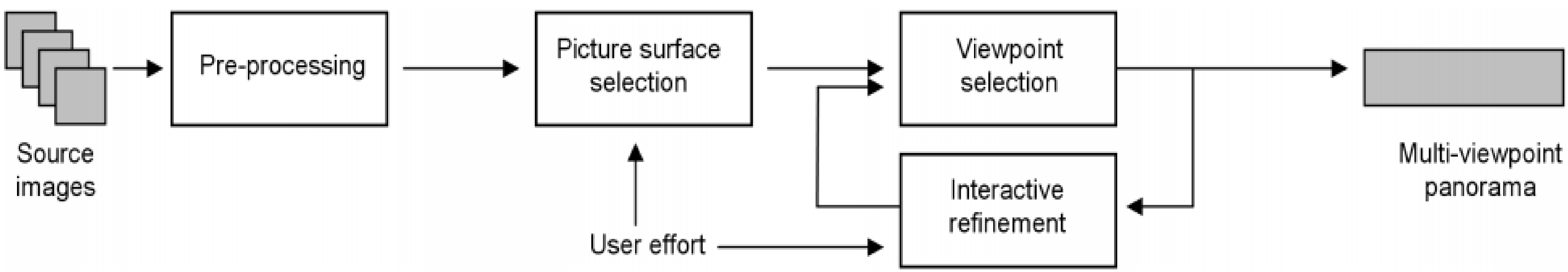
- Also known as slit-scan panorama
- Past: created by sliding slit-shaped aperture across film
- Now: extract thin, vertical strips of pixels from frames of a video sequence

Disadvantages of Strip Panorama

- Objects further from camera, horizontally stretched
- Closer objects, squashed
- For automatic system – complex capture setup
- Bad quality
- Do not preserve depth cues

System Overview

- Goal: reduce disadvantages of strip panoramas
- Stitch together arbitrary regions of source images
- Use Markov Random Field optimization to solve objective function
- Allows interactive refinement



What constitutes a good panorama image?

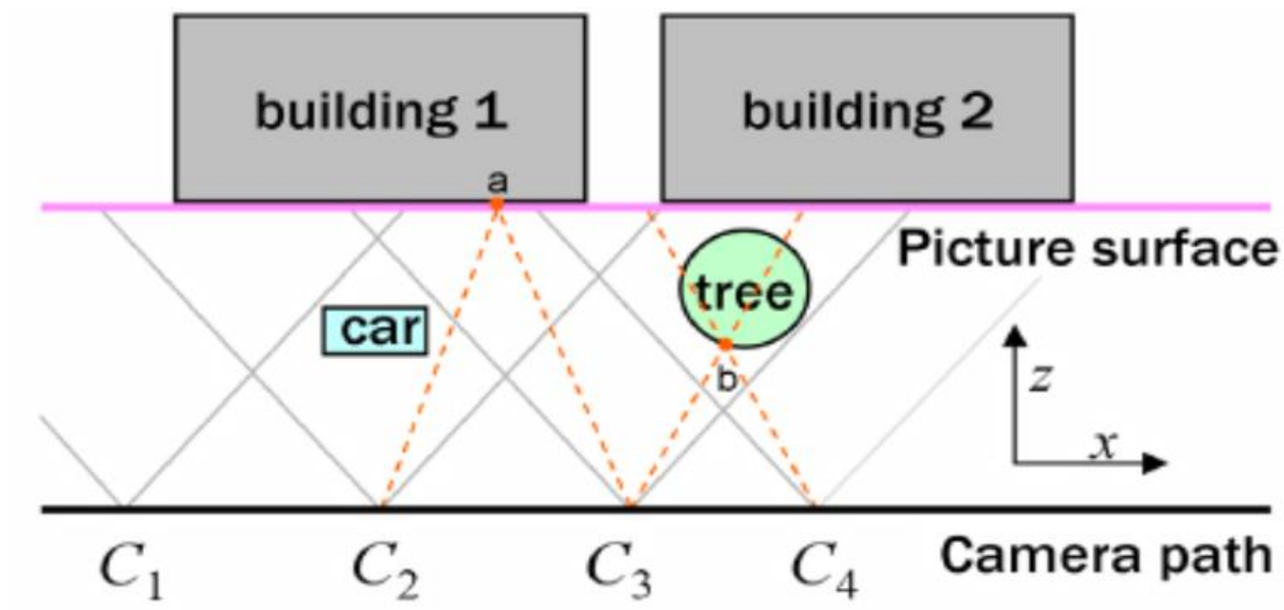
- Inspired by work of artist Michael Koller
- Each object in the scene is rendered from a viewpoint in front of it (avoid perspective distortion)
- Panoramas composed of large regions of linear perspective seen from a viewpoint where a person would naturally stand (city block viewed from across street, not far away)
- Local perspective effects are evident (closer objects larger than farther objects)
- Seams between these perspective regions do not draw attention (natural/continuous)

Image Types

- Those too long to effectively image from single viewpoint
- Those whose geometry predominantly lies along large, dominant plane
- 3D images are less likely to work well (turn around street corners, four sides of buildings, etc.)

Key Observation

- Images projected onto the picture surface from their original 3D viewpoints will agree in areas depicting scene geometry lying on the dominant plane (point a will project from each camera to same pixel on picture surface, while point b will project to different places)



Key Observation

- This agreement can be visualized by averaging the projections of all of the cameras onto the picture surface
- The resulting image is sharp for geometry near the dominant plane because these projections are consistent and blurry for objects at other depths

Choose the best seam

1. Capture images

Capture lots of images (40 min)

- e.g. 107 for this road

1. Capture images

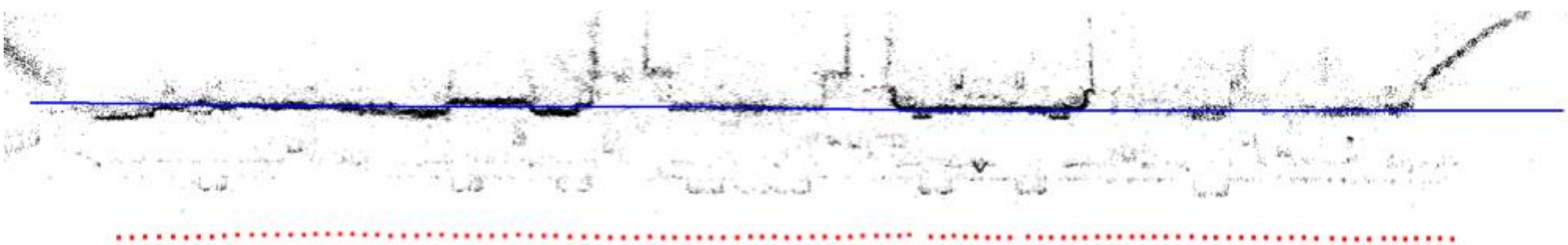
- Photographs taken with hand-held camera
  - From multiple viewpoints along scene
  - Intervals of one large step (~1m)
  - Auto focus
  - Manual exposure
- Fisheye lens for some scenes
  - Cover more scene content in one picture to avoid frequent "viewpoint transition"

2. Preprocess

- Remove radial distortions (e.g. fisheye lens)
- Build projection matrices for each camera i
  - 3D rotation matrix R_i
  - 3D translation matrix t_i
  - Focal length f_i
  - Camera location in world coordinates: $C_i = -R_i^T t_i$
- Recover parameters using structure-from-motion system
  - Match SIFT features between pairs of inputs
- Compensate exposures

3. Picture Surface Selection

- Picture surface selected by user
- View of recovered 3D points
- Automatic definition of coordinate system
  - Fit plane to camera viewpoints using PCA
- Blue line: picture surface selected by user
- Red line: extracted camera locations



3. Picture Surface Selection

- Project source image onto picture surface

  - S(i,j): 3D location of sample (i,j) on picture surface
  - S(i,j) projected into source

3. Picture Surface Selection

Average image

After warping + cropping

4. Viewpoint Selection

- Each image I_i represents i'th viewpoint

- Task: choose color for each pixel p = (px,py) in panorama from one source image: $I_i(p)$
- In essence, a pixel labeling problem

4. Viewpoint Selection

- Objective function
- For every point p of result find best source image L(p) = i if pixel p of the panorama is assigned color $I_i(p)$
- Best = minimizing energy
- Minimize using MRF optimization
- 3 terms

$$\sum_{p}(\alpha D(p,L(p))+\beta H(p,L(p)))+\sum_{p,q}V(p,L(p),q,L(q)),$$

4. Viewpoint Selection

- Term I
- D: an object in the scene should be imaged from a view point roughly in front of it
- Approximation of a more direct notion
  - Vector starting at S(p) of picture surface
  - Extend in direction normal to picture surface
  - Angle between C_i – S(p) and above vector
  - The higher the angle the less in front of object

$$D(p, L(p)) = |p - p_{L(p)}|$$

4. Viewpoint Selection

- p_i here (i.e. $p_{L(p)}$) = pixel in i-th image closest to camera (~center of the image) in the composite coordinates
- Find p_i
- Pixel p chooses its color from I_i
- Minimize 2D distance from p to $p_{L(p)}$

$$D(p, L(p)) = |p - p_{L(p)}|$$

4. Viewpoint Selection

- Term II
- H: cost function that encourages panorama to resemble average image in areas where scene geometry intersects picture surface
- Will occur naturally except in outliers resulted from motion, occlusions, etc.
- Want to discount outliers

$$H(p, L(p)) = \begin{cases} |M(p) - I_{L(p)}(p)| & \text{if } \sigma(p) < 10 \\ 0 & \text{otherwise} \end{cases}$$

4. Viewpoint Selection

- Median image, M(x,y)
  - Vector median filter computed across three color channels
- MAD, σ(x,y)
  - Median absolute deviation
- Minimize difference between median image and image defined by current labeling for pixels whose variance is low; 0 if variance is too large

$$H(p, L(p)) = \begin{cases} |M(p) - I_{L(p)}(p)| & \text{if } \sigma(p) < 10 \\ 0 & \text{otherwise} \end{cases}$$

4. Viewpoint Selection

$$\sum_{p}(\alpha D(p,L(p))+\beta H(p,L(p)))+\sum_{p,q}V(p,L(p),q,L(q)),$$

- Term III
- V: encourage seamless transition between different regions of linear perspective
- p and q are neighboring pixels

$$V(p,L(p),q,L(q))=|I_{L(p)}(p)-I_{L(q)}(p)|^2+|I_{L(p)}(q)-I_{L(q)}(q)|^2$$

4. Viewpoint Selection

$$\sum_{p}(\alpha D(p,L(p))+\beta H(p,L(p)))+\sum_{p,q}V(p,L(p),q,L(q)),$$

- Parameters α, β
- Determined experimentally
- α - typically 100
- β - typically 0.25
- Higher α = more straight views and more noticeable seams
- Lower both α and β = more likely remove objects off of the dominant plane

4. Viewpoint Selection

- The solver

$$\sum_{p}(\alpha D(p,L(p))+\beta H(p,L(p)))+\sum_{p,q}V(p,L(p),q,L(q)),$$

- Constraint: pixels in image I_i to which the I'th camera does not project are set as null -- > the black holes
  - L(p) = I is not possible if I_i = null
- Wish to compute panorama that minimizes overall cost function
- Resembles Markov Random Field optimization
- Minimize using min-cut optimization in a series of alpha-expansion moves
- Takes typically ~20 minutes
- Still, some artifacts remain
  - Fix them manually

5. Interactive Refinement: View Selection

Supply the solution L(p) manually for some pixels p

- Selects source image, draws stroke where source should appear in panorama

5. Interactive Refinement: Seam Suppression

- MRF optimization try to route seams around objects that lie off the dominant plane
- Such objects don’t always exist

Shortened car

Mark source

5. Interactive Refinement: Seam Suppression

Mark original images, propagate to projected image

- Allows indication of objects in scene across which seams should not be placed

Keep whole region as much as possible

Example Result