# Rectangling Panoramic Images via Warping

Kaiming He, Microsoft Research Asia
Huiwen Chang, ITCS, Tsinghua University
Jian Sun, Microsoft Research Asia

SIGGRAPH 2013

# Motivation

- Limitation of digital camera
- Make the scene more descriptive

# Problem Definition

- How to find the feature points of an image
- Errors when matching the feature points
- Find a better method to warp the image to a rectangle region

# Algorithm

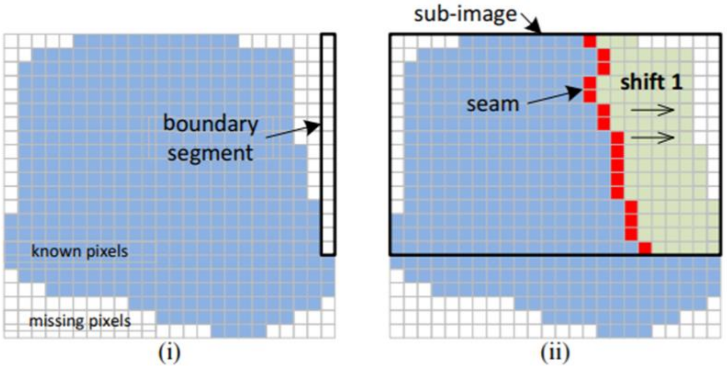
- Part 1. stitching
- Part 2. local warping
- Part 3. global warping

# Algorithm part 1: stitching

- Find the feature points of all input images.
  - If distance < threshold: good key points
  - Good key points -> pair
- Match the feature points of two consecutive images and merge them all into a big panorama.
  - Draw a line between them
  - Match the key point pair
- **sift_{name}.png**

# Algorithm part 2: local warping

- **Seam Carving Algorithm**
  - Vertical: dp_horizontal_segment
    - Down
    - Up
  - Horizontal: dp_vertical_segment
    - Left
    - Right
- **seg_{name}.png**



# Algorithm part 3: global warping (1)

- Draw meshes on seg_{name}.png and warp it back to the original image.
- Use lsd file on github
  - Line Segment Detector
  - https://github.com/theWorldCreator/LSD

# Algorithm part 3: global warping (2)

- energy function $\mathrm{E}(\mathrm{v},\{\theta_\mathrm{m}\}) = E_S(V) + \lambda_L E_L(\mathrm{V},\{\theta_\mathrm{m}\}) + \lambda_B E_B(V)$
- shape preservation
  - 將網格座標代入矩陣，進行一些矩陣計算
    - pseudoinverse
    - 一次微分 ( $\frac{d(x^T A)}{dx} = A$ ，$\frac{d(Ax)}{d(x^T)} = A$ )
    - 二次微分 ( $\frac{d(x^T Ax)}{dx} = (A + A^T)x$ )

$$\begin{array}{ll} \underset{x}{\min} & \frac{1}{2}x^\top Px + q^\top x \\ \text{subject to} & Gx \preceq h \\ & Ax = b \end{array}$$

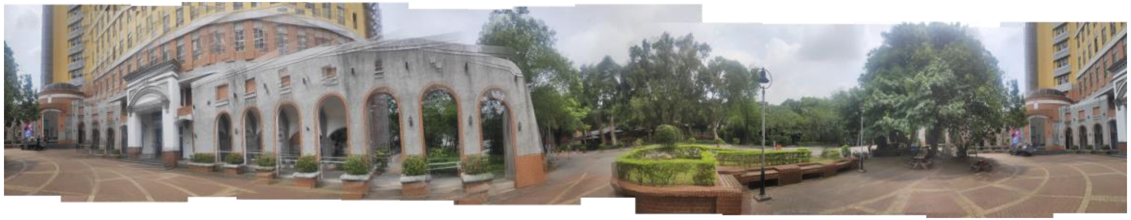
- line preservation: detect line segment (LSD)
- boundary constraints
  - 用 cvxopt (convex optimization 之 package) 限制邊界點的移動方向

# Algorithm part 3: global warping (3)

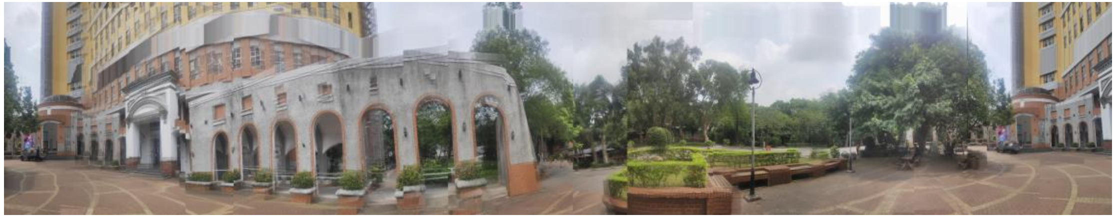
- alternating algorithm: $\min_{\theta_m} \sum_{j\in bin(m)} \left\| C_j(\theta_m e_{q(j)} \right\|^2$
- fix theta and update V:
  - ○ 根據 cvxopt 和矩陣微分，來求更新後的網格座標
- fix V and update theta:
  - ○ 將 theta 量化成 50 個角度
  - ○ 計算 warping 前後，直線之旋轉角度：bilinear quadrilateral interpolation
- repeat 大約 7 次
- 最終將圖片大小rescale
- **./{output_folder}/result.png**

# Expected Results – Image 1

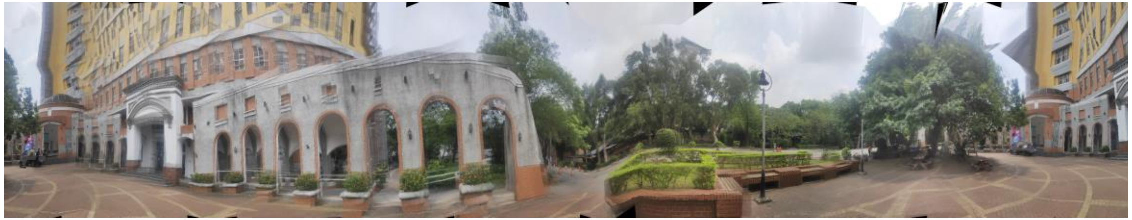
- sift_campus.png



- seg_campus.png

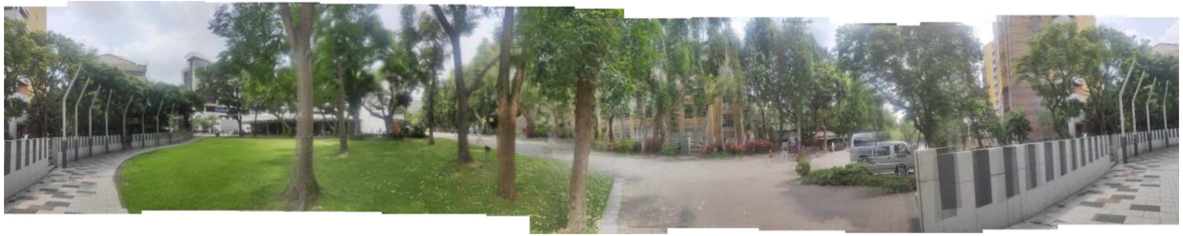


- ./result_campus/result.png

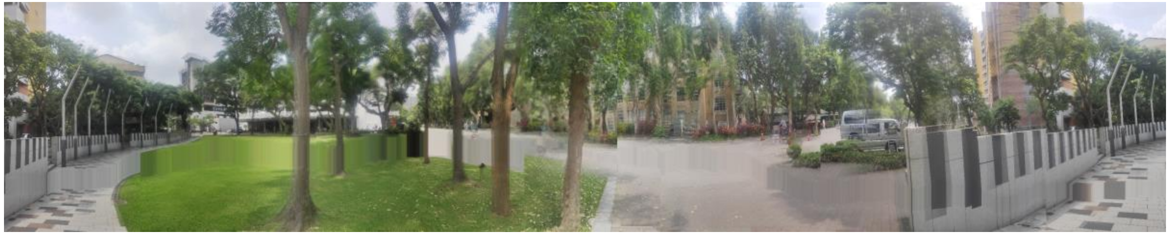


# Expected Results – Image 2

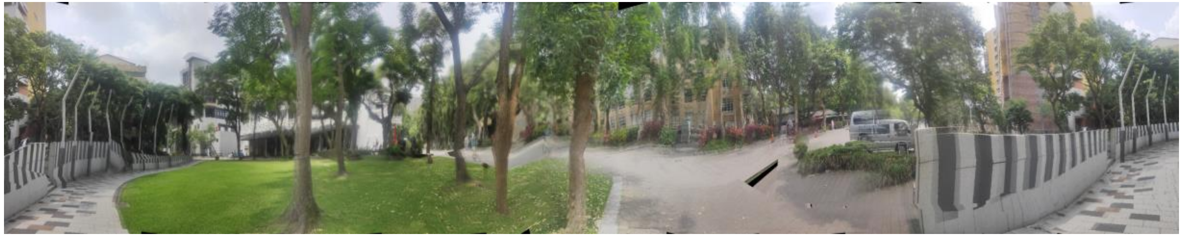
- sift_garden.png



- seg_garden.png

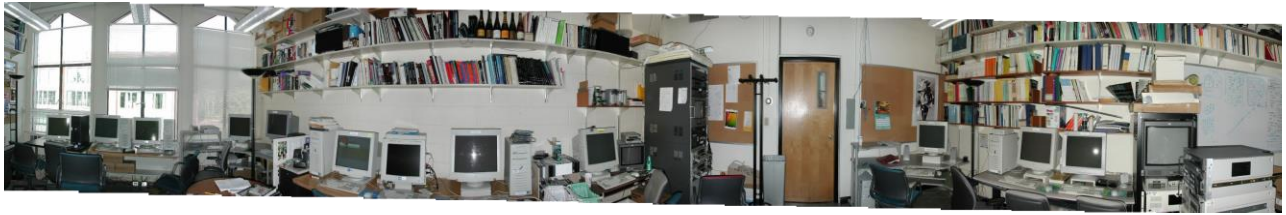


- ./result_garden/result.png

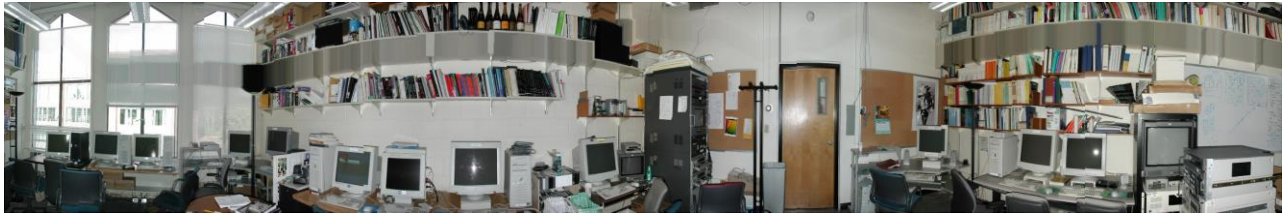


# Expected Results – Image 3

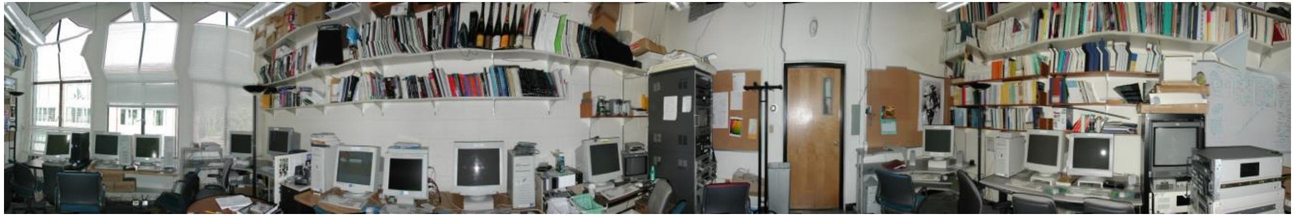
- sift_grail.png



- seg_grail.png



- ./result_grail/result.png



# Discussion

- 執行時間稍長
    - stitching: 3 min
    - local warping: 10 ~ 15 min
    - global warping: 10 ~ 15 min
- global warping的部份，論文的一些細節沒寫清楚，參數的選擇很多，且結果好壞有點主觀，沒找到一項標準來選擇好的參數。
- 有時候global warping反而扭曲了部分原本正常的區域

# References (1)

**Rectangling Panoramic Images via Warping, 2013**
**http://kaiminghe.com/publications/sig13pano.pdf**

**Seam segment carving: retargeting images to irregularly-shaped image domains, 2012**
**https://projet.liris.cnrs.fr/imagine/pub/proceedings/ECCV-2012/papers/7577/75770314.pdf**

**Image Alignment and Stitching: A Tutorial, 2006**
**https://dl.acm.org/doi/10.1561/0600000009**

**Image warps for artistic perspective manipulation, 2010**
**https://dl.acm.org/doi/abs/10.1145/1833349.1778864**

# References (2)

**LSD - Line Segment Detector**

https://github.com/theWorldCreator/LSD

**Python implementation of bilinear quadrilateral interpolation**

https://stackoverflow.com/questions/49071685/python-implementation-of-bilinear-quadrilateral-interpolation

**PIL rotate image colors (BGR -> RGB)**

https://stackoverflow.com/questions/4661557/pil-rotate-image-colors-bgr-rgb

**Mapping a rectangle to a quad with Pillow**

https://stackoverflow.com/questions/65981589/mapping-a-rectangle-to-a-quad-with-pillow

# References (3)

**常用矩陣微分公式**

**https://www.itread01.com/content/1549269003.html**

**How to set up multiple equality constraints in quadratic programming in python?**

**https://stackoverflow.com/questions/58828911/how-to-set-up-multiple-equality-constraints-in-quadratic-programming-in-python**