Scene Completion of Selfie Photos based on Image Search

Woong Jick Youn

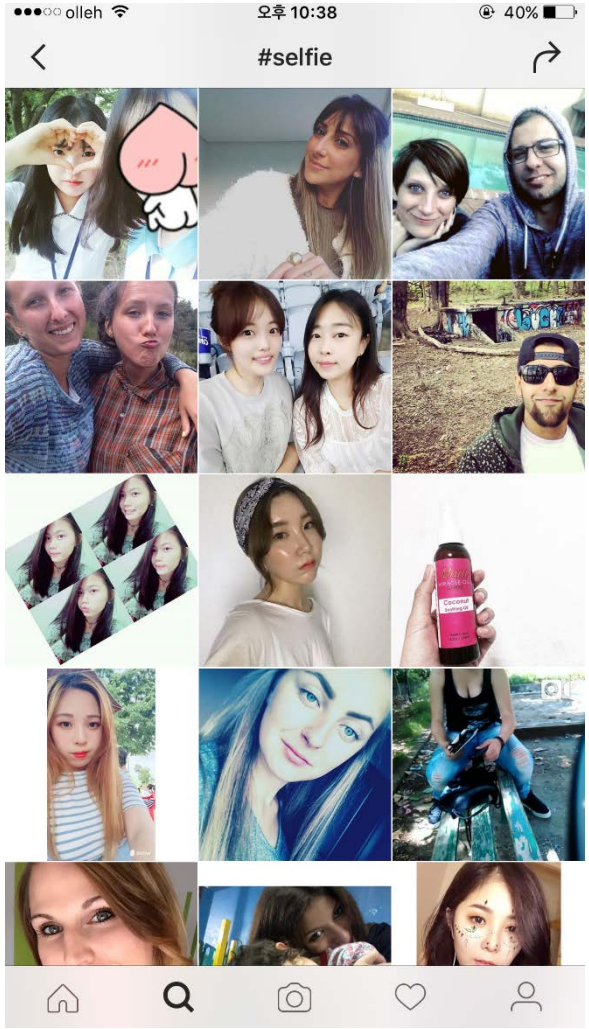
2016.06.16

Advisor: Sung-Eui Yoon

Images on Social Network Service

- Selfies are focused on their faces
- How can we expand the background?
- Can we make a larger selfie

  with scene completion?



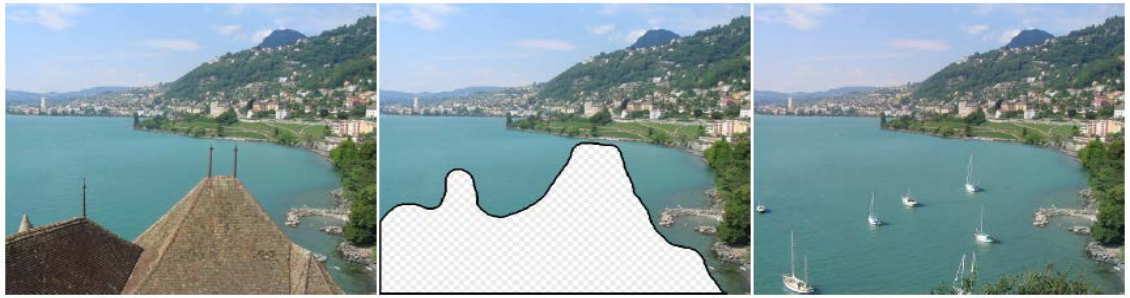
Images from Instagram and geekfence

Scene Completion on Large-scale Images

- Scene Completion Using Millions of Photographs

  [Hays and Efros, *SIGGRAPH* 07]

- Data-driven approach with millions of images



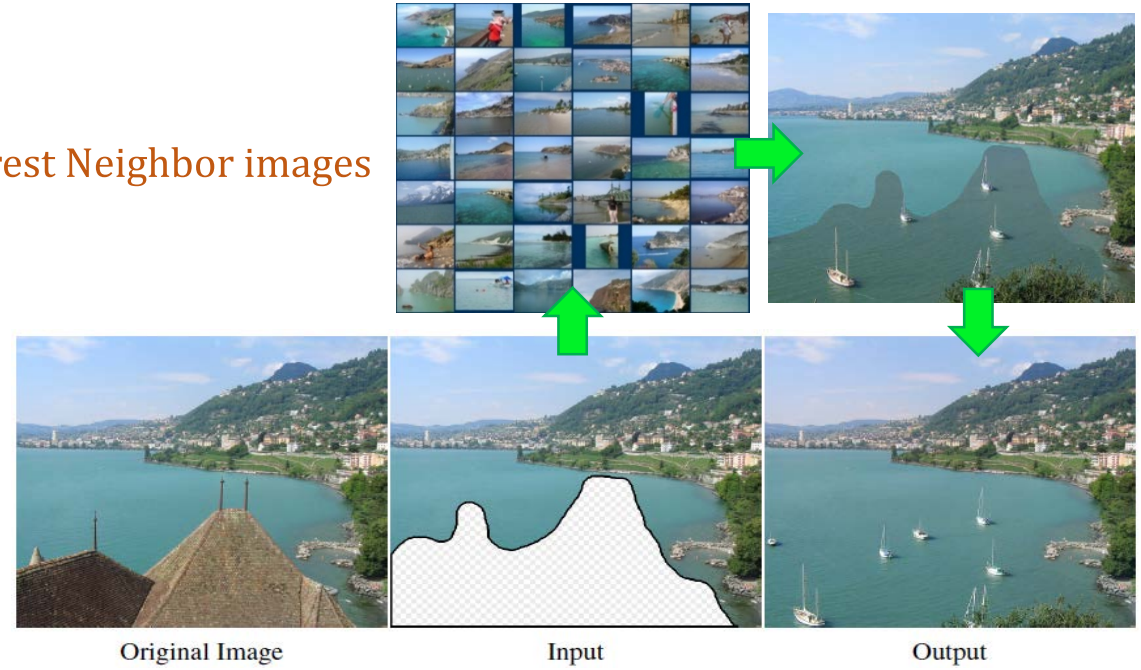
Original Image | Input | Output

Images from Scene Completion Using Millions of Photographs

Scene Completion on Large-scale Images

- Finding visually similar images with nearest neighbor search
- But, expensive for a large image collection
- To solve the scalability issue, we can consider ANN search

Nearest Neighbor images



Images from Scene Completion Using Millions of Photographs

Approximate Nearest Neighbor (ANN) Search

- Characterizes the neighborhood rather than identifying the exact neighborhood themselves
- Finds the NN with high probability
- Can be much faster, but missing some nearest matches

Our Approach

- Enlarges a selfie with scene completion through image search
- Expands upward and both sides with image patches
- Reduces memory footprint by ANN search

Images from Instagram

Overview of Our Approach

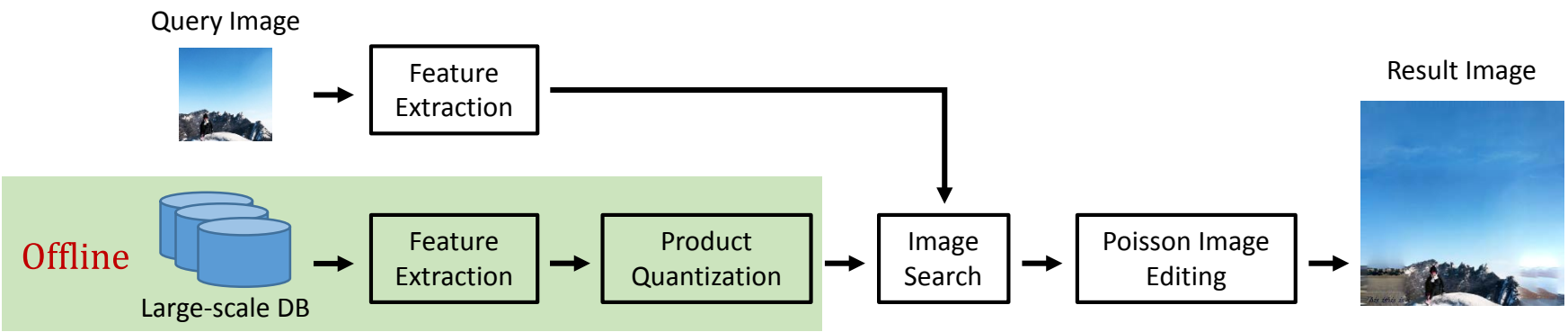
① Extracts GIST and LBP-HSV descriptors

② Product quantization

③ Patch-based image search

④ Poisson image editing for image completion

⑤ Iterates ③ and ④ for all the patches



Overview of our system

Overview of Our Approach

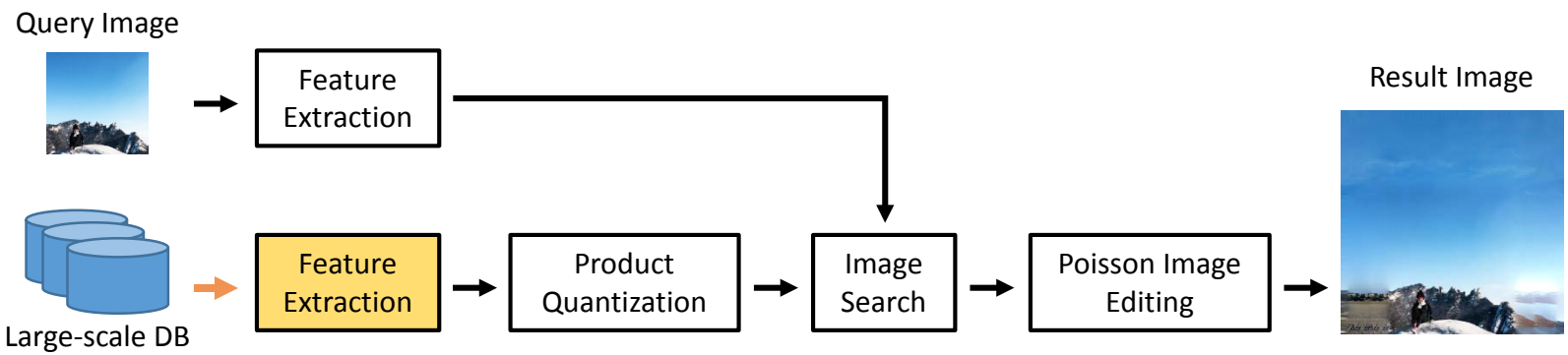
① Extracts GIST and LBP-HSV descriptors

② Product quantization

③ Patch-based image search

④ Poisson image editing for image completion

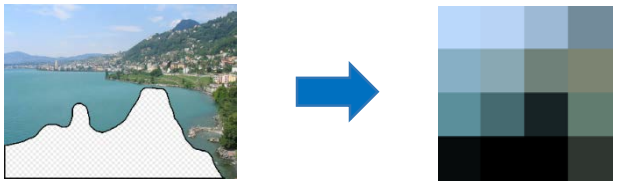
⑤ Iterates ③ and ④ for all the patches



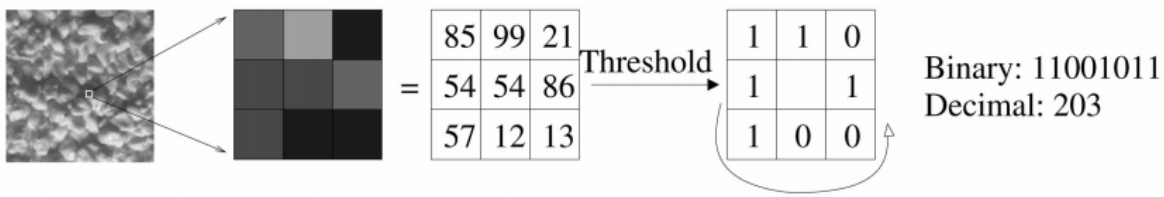
Overview of our system

① Descriptors

- GIST
  - Extracts in 8 x 8 blocks with 4 scales (4,608 dim)



- LBP-HSV
  - HSV color space for texture information
  - Extracts in 8 x 8 blocks (3,776 dim on each HSV channel)



GIST Images from Scene Completion using Millions of Photographs slides
LBP Images from Face Description with Local Binary Patterns: Application to Face Recognition

① Descriptors

- GIST
  - Extracts in 8 x 8 blocks with 4 scales (4,608 dim)
  - 32GB from 1M images
- LBP-HSV
  - Extracts in 8 x 8 blocks (3,776 dim on each HSV channel)
  - 78GB (26GB on each HSV channel) from 1M images
- Over 100GB for two features
- Scalability issue
- Solve this issue with product quantization

Overview of Our Approach

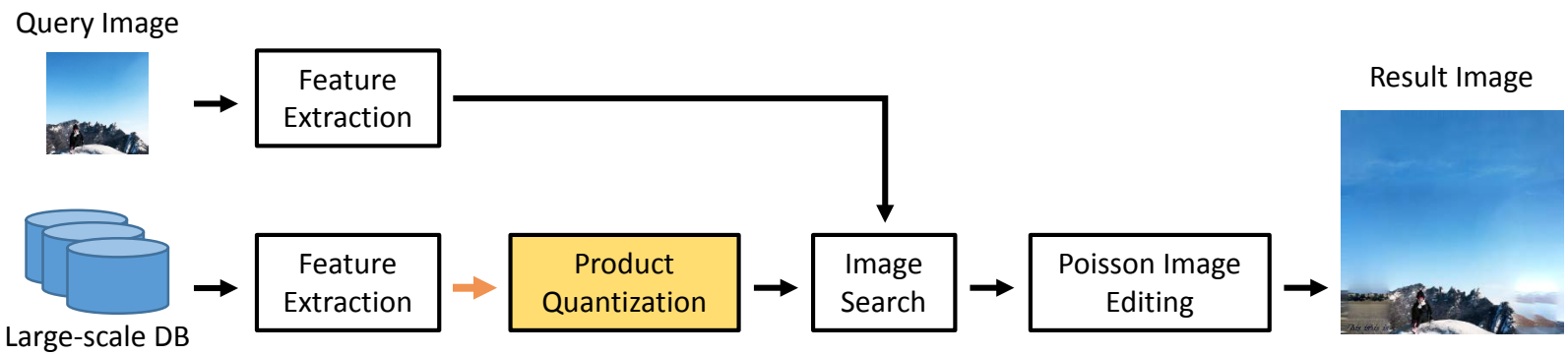
① Extracts GIST and LBP-HSV descriptors

② Product quantization

③ Patch-based image search

④ Poisson image editing for image completion

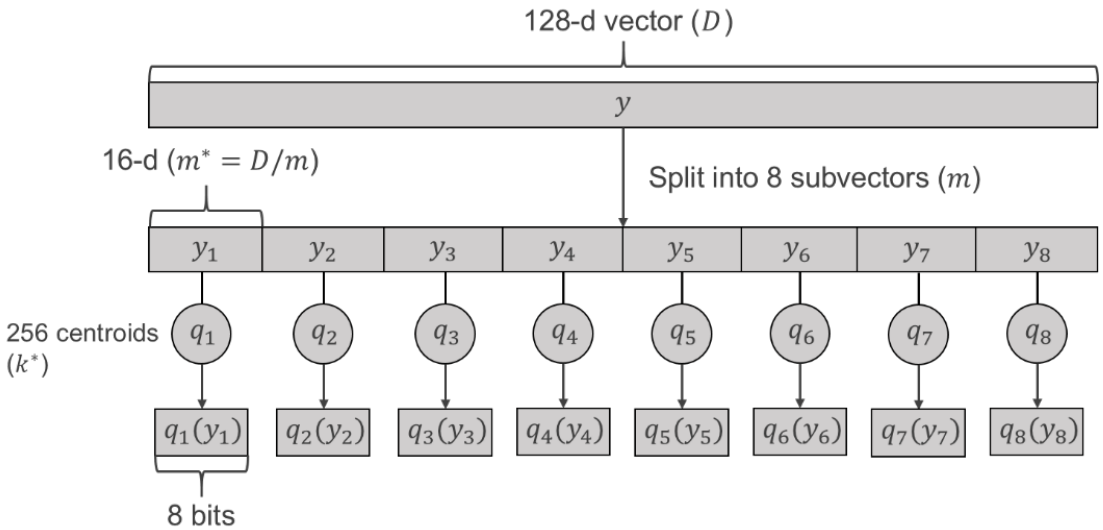
⑤ Iterates ③ and ④ for all the patches



Overview of our system

② Product Quantization (PQ)

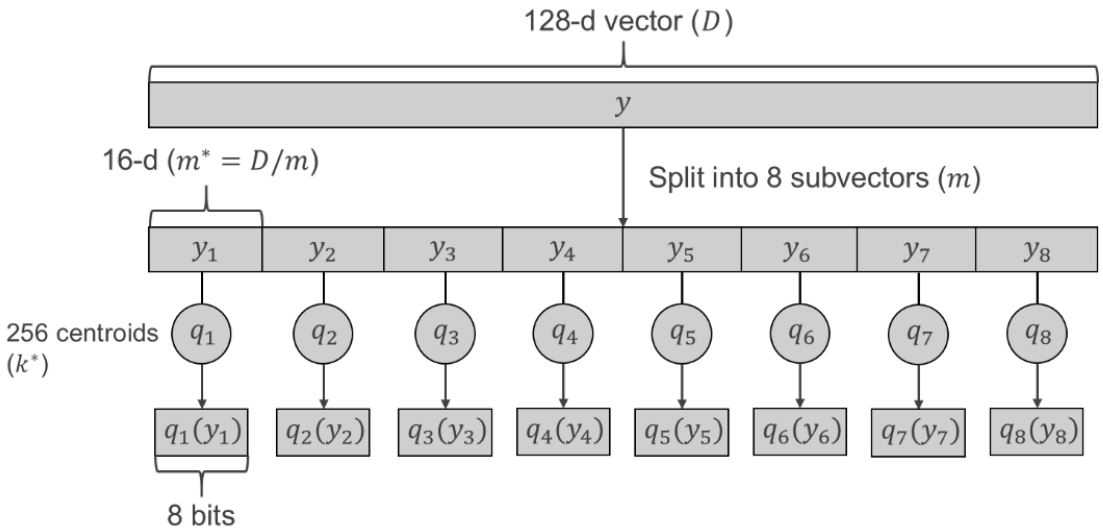
- One of the ANN search techniques
- Vector quantization with block-wise Cartesian product
- Decomposes the original vector space into disjoint subspaces
- Effectively generates clusters in each subspace



Images from Product quantization for nearest neighbor search

② Product Quantization (PQ)

- Product quantization on both descriptors
  - Generates 4 codebooks
  - 64 subspaces (Utilizing 8 x 8 spatial bins of descriptors)
  - Each subspace is quantized to 256 codewords



Images from Product quantization for nearest neighbor search

Overview of Our Approach

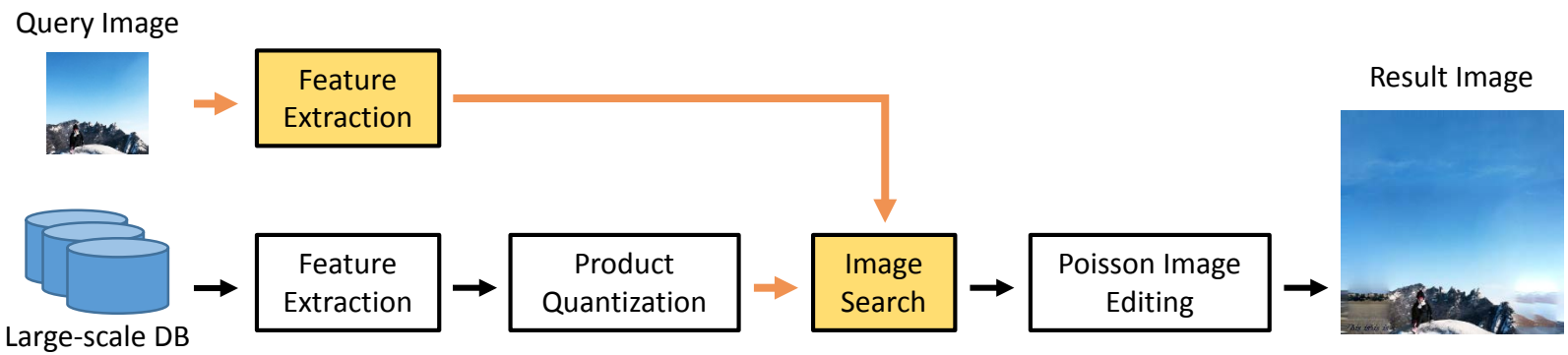
① Extracts GIST and LBP-HSV descriptors

② Product quantization

③ Patch-based image search

④ Poisson image editing for image completion

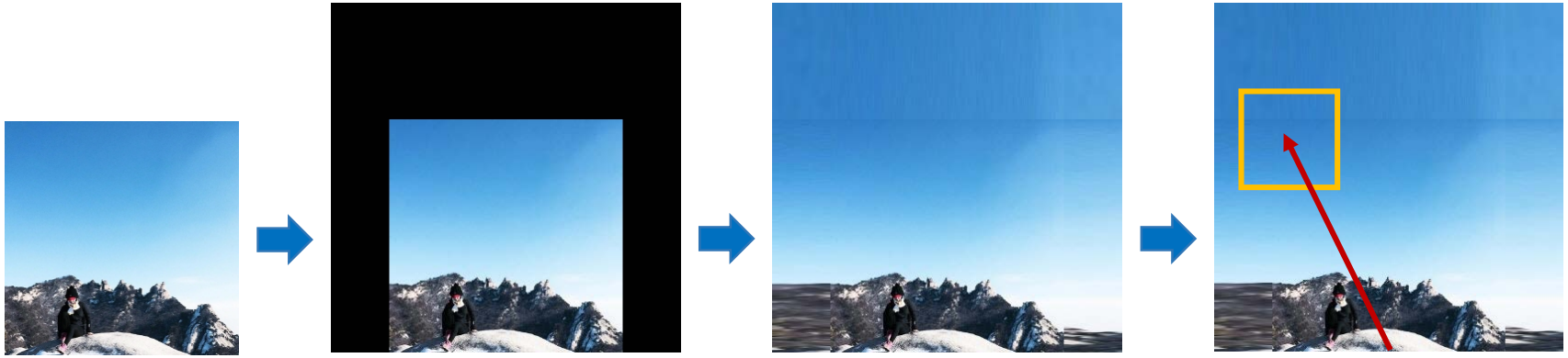
⑤ Iterates ③ and ④ for all the patches



Overview of our system

③ Patch-based Image Search

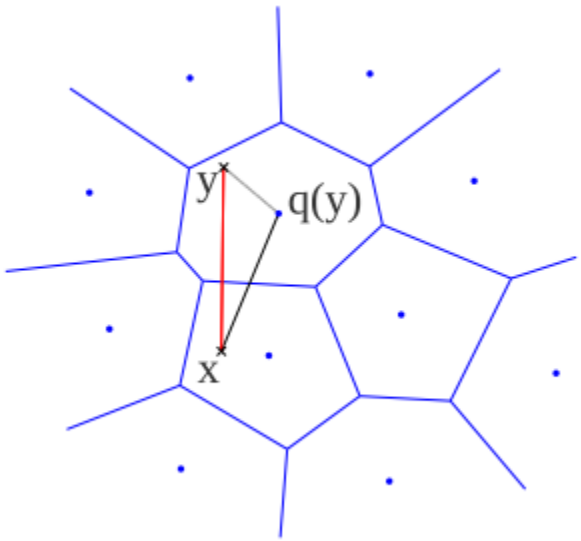
- Generates a target region by a user input
- Expand the boundary regions to marginal regions
  - Reduces drastic color changes
- A small patch searches a similar image by PQ



Query image → Target region → Filling marginal regions → Searching similar image

③ Patch-based Image Search

- Distance computation for search by PQ
- The distance of GIST descriptor
  - $d_{gist}(x, y) = \sqrt{\sum_{i=1}^{m} \| x - q_i(y) \|^2}$
- The distance of LBP-HSV descriptors
  - $d_h(x, y) = \sqrt{\sum_{i=1}^{m} \| x - q_i(y) \|^2}$
  - $d_s(x, y) = \sqrt{\sum_{i=1}^{m} \| x - q_i(y) \|^2}$
  - $d_v(x, y) = \sqrt{\sum_{i=1}^{m} \| x - q_i(y) \|^2}$
- The distance the query and image collection
  - $dist(x, y) = d_{gist}(x, y) + d_h(x, y) + d_s(x, y) + d_v(x, y)$



Overview of Our Approach

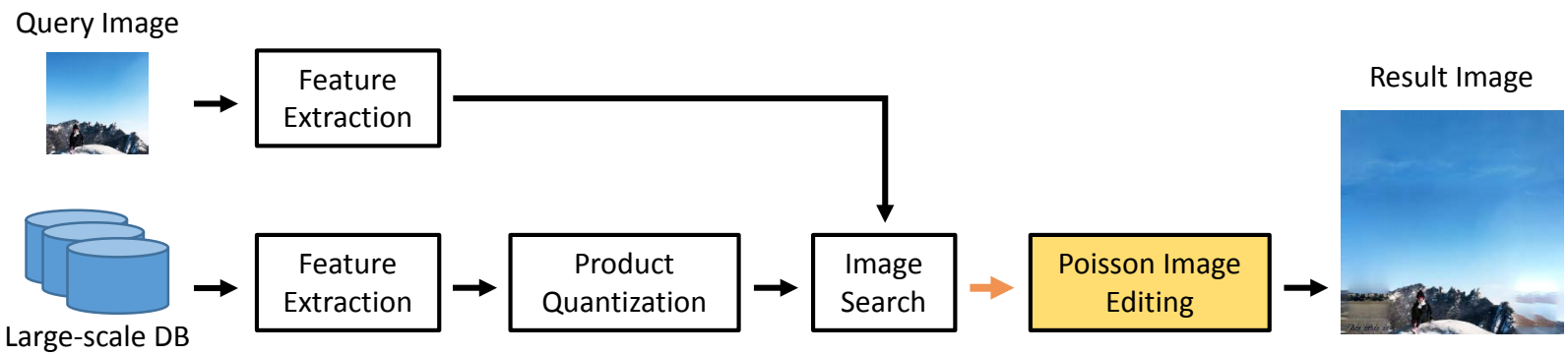
① Extracts GIST and LBP-HSV descriptors

② Product quantization

③ Patch-based image search

④ Poisson image editing for image completion

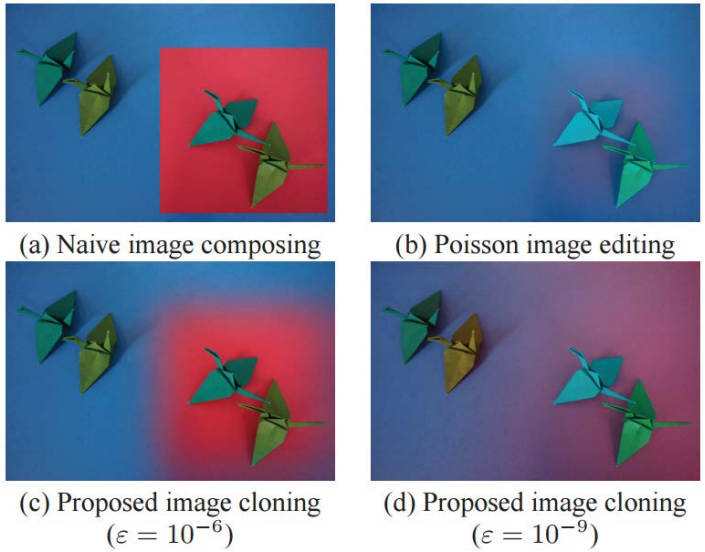
⑤ Iterates ③ and ④ for all the patches



Overview of our system

④ Poisson Image Editing

- Seamless Image Cloning by a Closed Form Solution of a Modified Poisson Problem [SIGGRAPH Asia 12]
  - Can control the level of color adaptation
  - Suppresses drastic color changes



Images from Seamless Image Cloning by a Closed Form Solution of a Modified Poisson Problem

④ Poisson Image Editing

- Blends similar image on the boundary regions only
- Preserves the color in blended regions
- Can keep the visual information on the rest of the region

Patch image

Similar image from dataset

Result of image completion

Overview of Our Approach

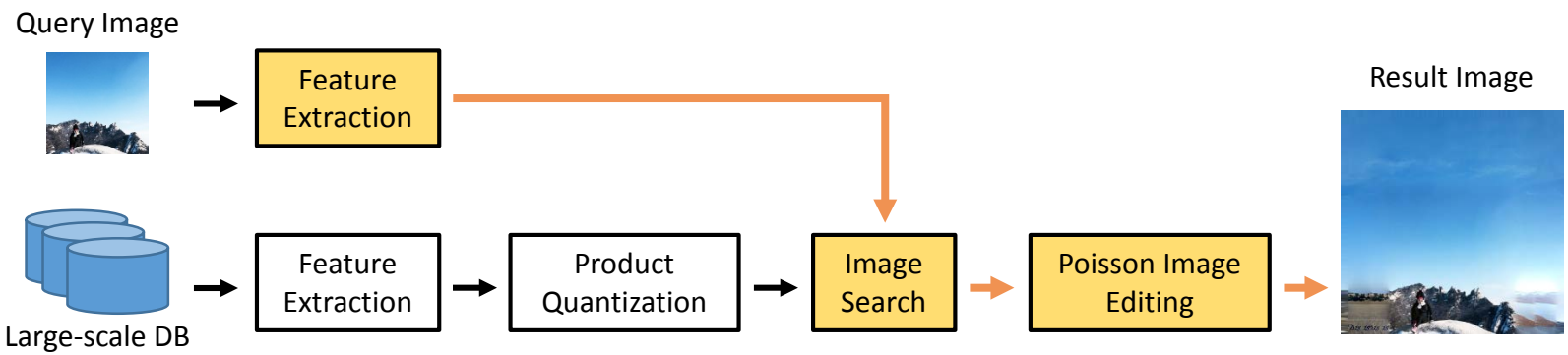
① Extracts GIST and LBP-HSV descriptors

② Product quantization

③ Patch-based image search

④ Poisson image editing for image completion

⑤ Iterates ③ and ④ for all the patches



Overview of our system

⑤ Iteration

- Steps for patch-based image completion

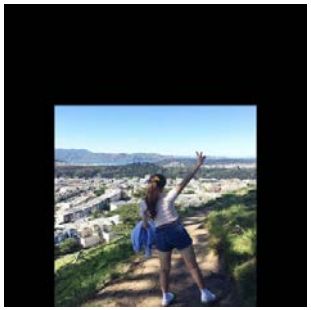
Dataset

- MIRFlickr-1M
- One million Flickr images
- Excludes grayscale images for the dimension consistency
- After excluding, 936,800 images are used

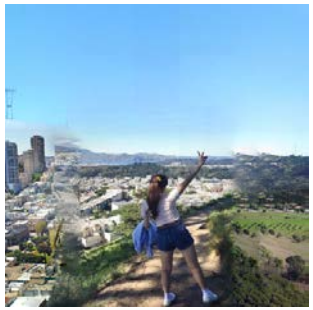
Images of MIRFlickr-1M dataset

Result

- Memory efficient
  - Descriptors are compressed to 250MB from 110GB (450x)
  - Runnable on a typical computer
- Contains a creative content
- Avoids the duplication problem caused by a copy and paste manner



Target image



Final result

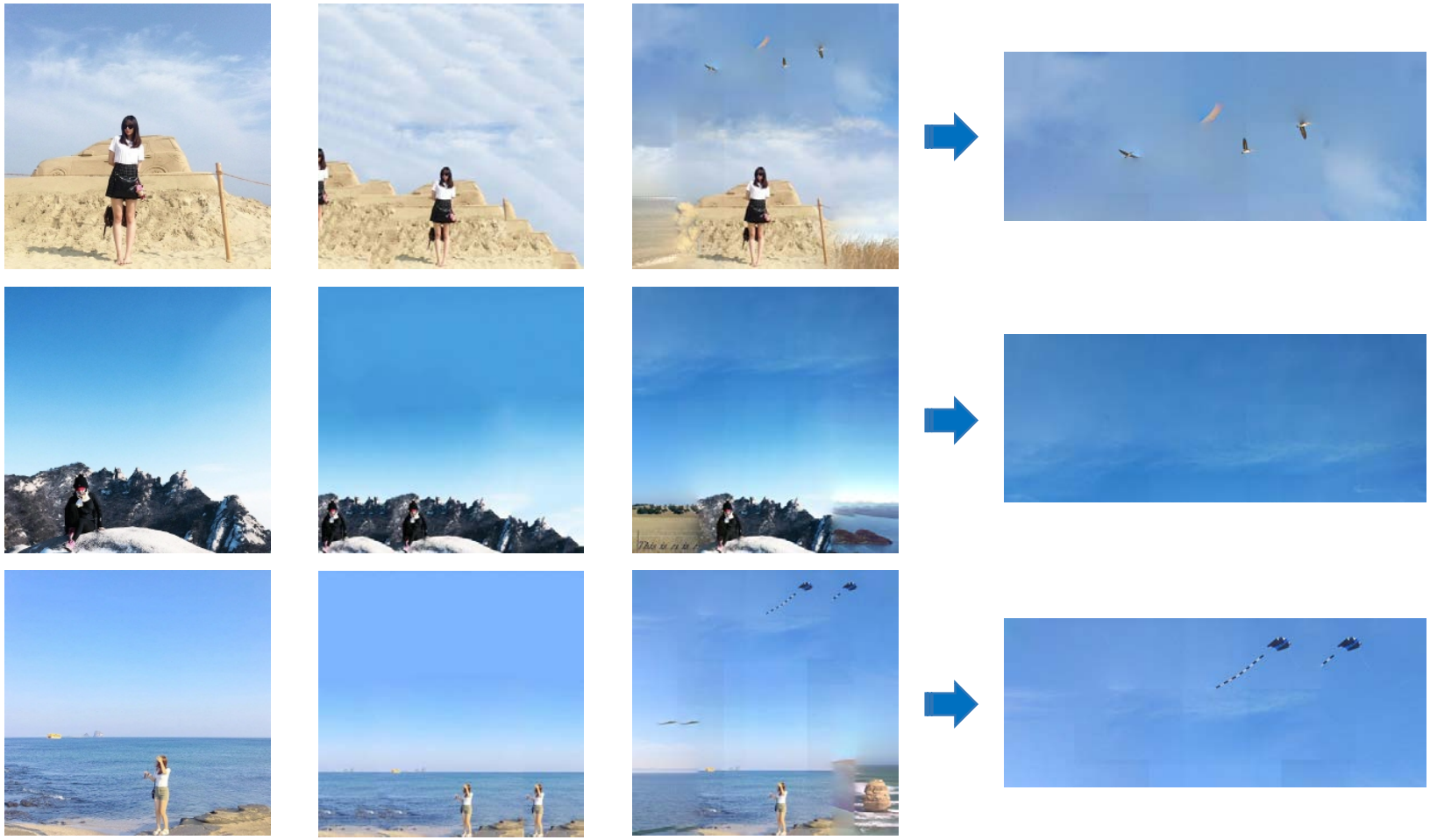
Result

- Compared our result with content-aware fill in Photoshop CC

Query image | Content-aware fill | Ours

Result

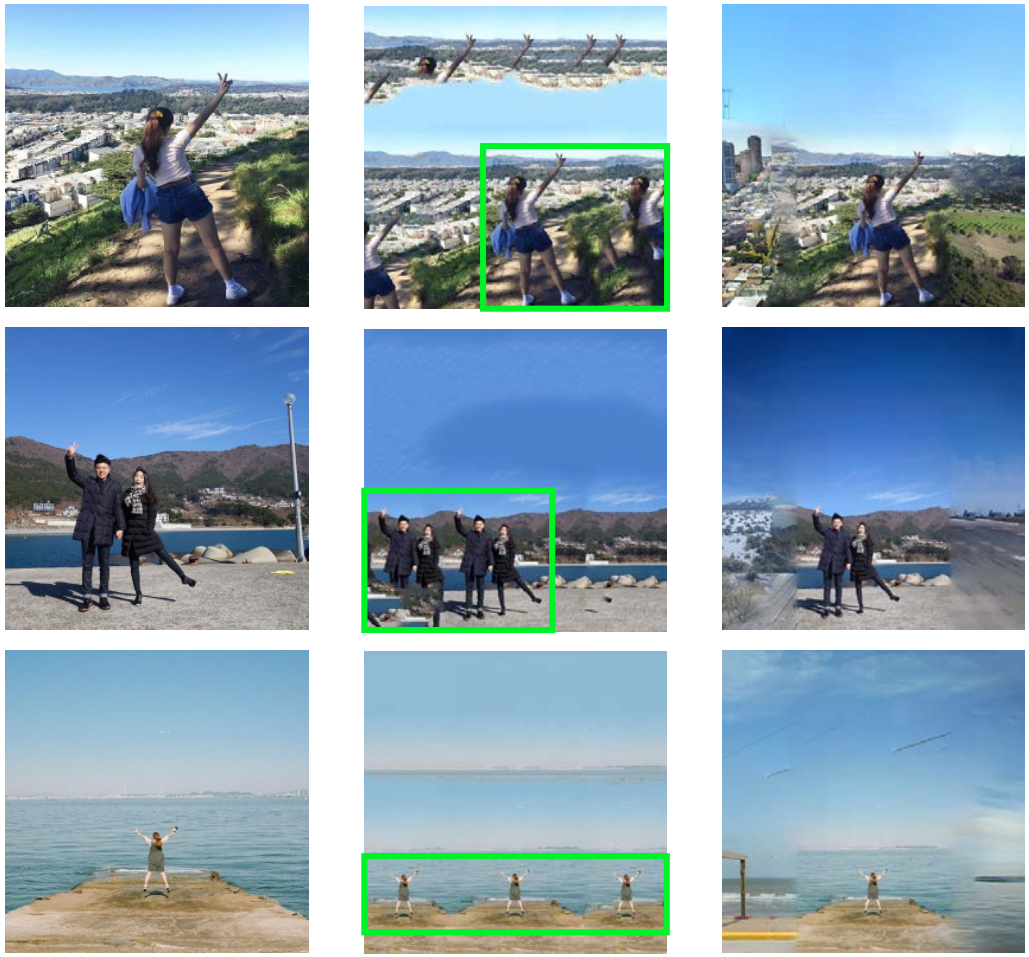
- Contains a creative content



Query image | Content-aware fill | Ours

Result

- Avoids the duplication problem



Query image | Content-aware fill | Ours

Conclusion

- Scene completion based on product quantization
- Memory efficiency of image search
- Extrapolates the background where visual information does not exist
- Generates a creative image

Future work

- A larger image collection with tags
- Graph matching algorithm
- Fine matching for improving side patches

Publication

- 이미지 검색을 활용한 자가촬영 사진의 배경 합성

  윤웅직, 윤성의

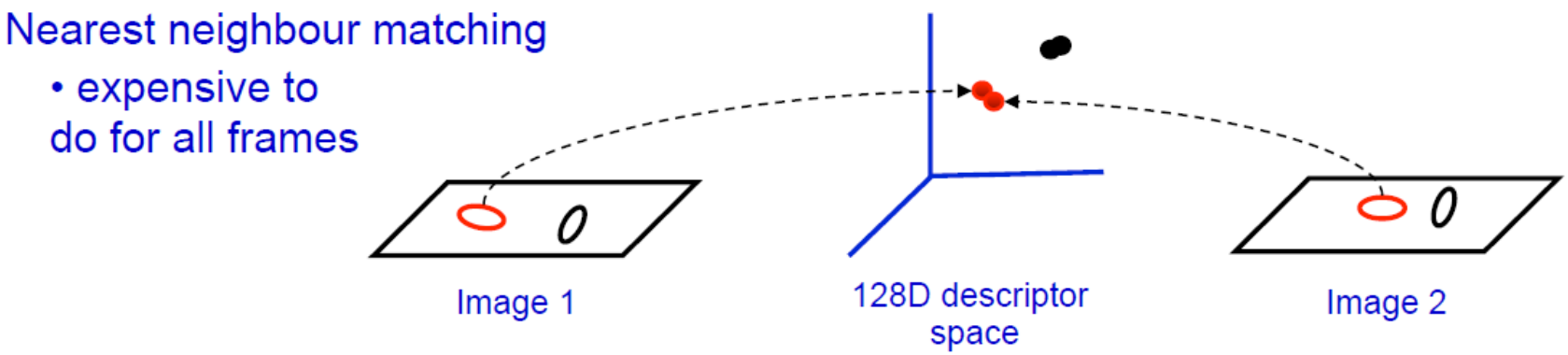
  IPIU 2016

Thank you

Any questions or comments?

Nearest Neighbor Search

- Extracts image descriptors
- Compares the distance between descriptors
- Finds nearest neighbor features from all the database images
- $NN(x) = argmin_y \parallel x - y \parallel$



Slide from Visual Recognition and Machine Learning Summer School 2013

Nearest Neighbor Search

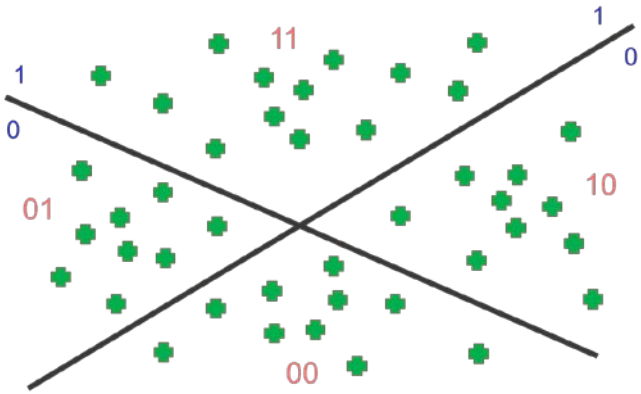
- NN search is expensive for a large image collection
- Matching two images (1k 128-d SIFT descriptors)

  NN search in 0.4 seconds
- To solve the scalability issue, we can consider ANN search

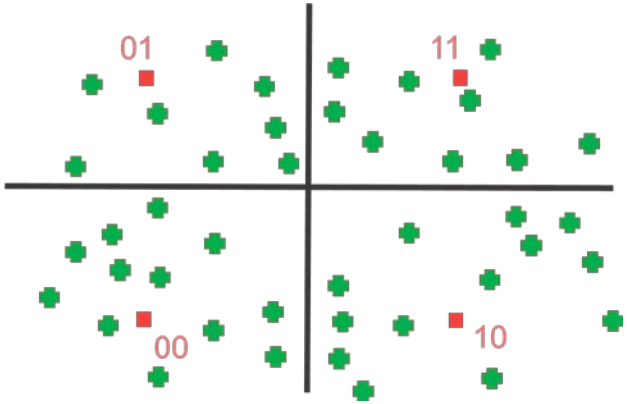
# of images	Time
1000	~ 7 mins
10,000	~ 1h 10 mins
10 million	~ 115 days

Approximate Nearest Neighbor (ANN) Search

- Characterizes the neighborhood rather than identifying the exact neighborhood themselves
- Finds the NN with high probability
- Can be much faster, but missing some nearest matches
- E.g. LSH, ITQ, PQ …



LSH



ITQ

Scene Completion

- BiggerPicture: Data-Driven Image Extrapolation Using Graph Matching [Wang et al., *SIGGRAPH Asia* 14]
- Hierarchically segmented to build graphs in multi levels
- Graph matching algorithm

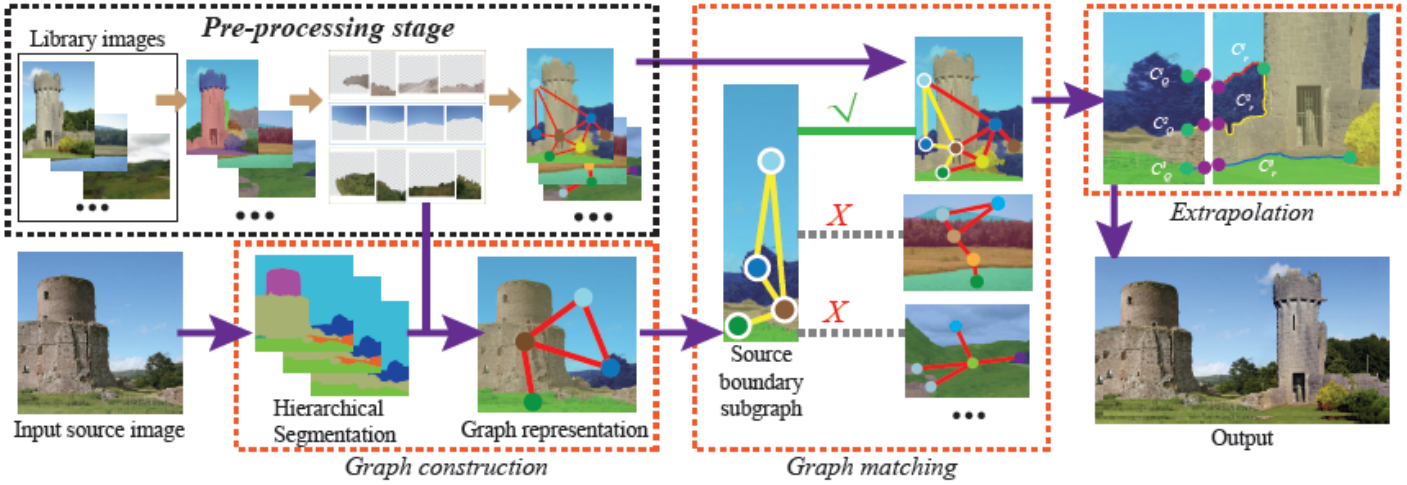
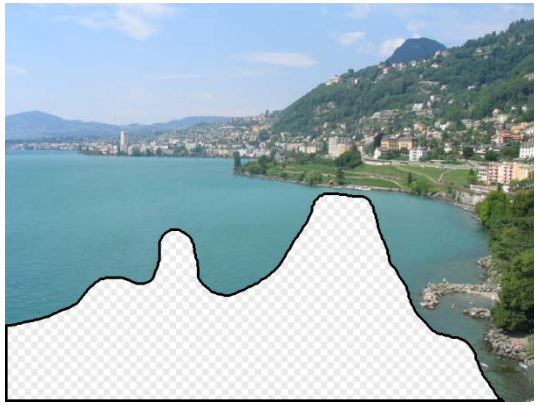


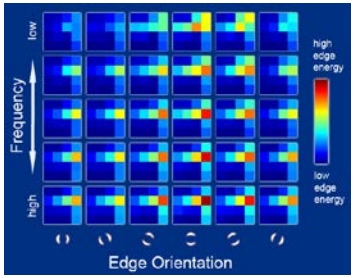
Figure 2: *System pipeline.*

Images from BiggerPicture: Data-Driven Image Extrapolation Using Graph Matching

Scene Completion



Input image



Scene Descriptor

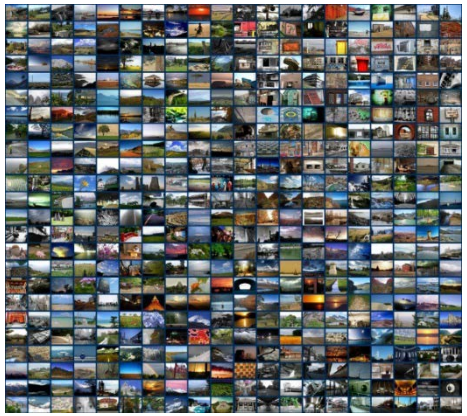


Image Collection

200 matches

Context matching
+ blending

20 completions

Slides from Scene Completion Using Millions of Photographs

Result

- Compare with extrapolation method





Query image | Wang et al. | Ours

- BiggerPicture: 5mins for segmentation, 5secs for graph matching, 3secs for stitching and alignment
- Ours: 13secs for extrapolation with low memory footprint