Scene Summarization for Online Image Collections

Ian Simon Noah Snavely Steven M. Seitz
University of Washington

Presenter: Eileen Bai
Discussant: Armin Samii

Big ideas:

- use multi-user image collections from the Internet
- represent the visual content of a given scene
- most interesting important aspects of the scene with minimal redundancy
- select canonical views to form the scene summary
- analyze user-specified image tag data

10-image summary of the Vatican

tourist
stpeters

michelangelo
genesis

dome
altar

schoolofathens
raphael

galleryofmaps
ceiling

stpeters
hdr

pose
lacoon

blurry
glass

spiral
staircase

stpeters
cathedral

Current problem:

- image collections are unorganized to search through
- want to find a one page visual summary of a scene/city

Three Part Solution:

1. group images together that correspond to different representative views of the scene - *clustering* techniques
2. identify what view is "canonical" - *likelihood* measures
3. compute tag information that represents that scene - *probabilistic reasoning* on histograms

Terminology

- photo/image/view: 2D image
- collection: set of photos
- connected: multiple images containing same object
- scene: set of connected photos
- summary: set of representative photos

Goal:

Given a set of photos **V** of a single scene **S**, compute a summary **C** $\subseteq$ **V** s.t. most of the interesting visual content in **V** is represented in **C.**

Techniques:

Canonical views:

- recurring views in photo-sharing websites

Summarization:

- focus on selecting images as opposed to layout

Textual data:

- only used to enhance summaries
- select tags likely to apply to large clusters

Goal:

Given a set of photos ***V*** of a single scene ***S***, compute a summary ***C*** $\subseteq$ ***V*** s.t. most of the interesting visual content in ***V*** is represented in ***C***.

- Scene ***S*** is a set of visual features $f_1, f_2, \ldots f_{|S|}$ where each feature corresponds to exactly one point in 3D space.
- View V $\in$ ***V*** is represented by the subset of ***S*** corresponding to the features which are visible in the view.
- We have an |***S***| by |***V***| Boolean matrix where entry (i, j) = {T or F} depending on whether is feature is visible in the view.

Algorithm:

1. Compute Feature-Image Matrix
2. Select Summary Views
    1. Image likelihood
    2. Clustering objective for canonical views

Feature Image Matrix:

- find feature points using SIFT detector
- perform feature matching to get candidates
- prune against a fundamental matrix
- split matches into tracks where track = connected component of features
- each track corresponds to a single 3D point in ***S***
- construct |S| x |V| feature-image incidence matrix using the set of tracks

Selecting summary views:

- based on likelihood, where an image should be included if it is similar to many other images in the input set
- similarity between two views:

$$\mathrm{sim}(V_i, V_j) = \frac{|V_i \cap V_j|}{\sqrt{|V_i||V_j|}} \qquad (1)$$

- likelihood is then:

$$\mathrm{sim}(V_i, V_j) = V_i \cdot V_j$$

- which is related to:

$$\mathrm{lik}(V) = \sum_{V_i \in \mathcal{V}} (V_i \cdot V) \qquad (2)$$

$$p(X|\mu, h) = \prod_{x \in X} f(h) e^{h(x \cdot \mu)} \qquad (3)$$

$$\log p(X|\mu, h) = h \sum_{x \in X} (x \cdot \mu) + \log f(h) \qquad (4)$$

Selecting summary views:

- our clustering objective
- quality term for each view $V_i \in \mathbf{V}$
  - similarity between V_i and its closest canonical view $C_{c_{(i)}}$ in $\boldsymbol{C}$ where c contains view->canonical view mapping
  - cost term $\boldsymbol{\alpha}$ to penalize solutions with too many canonical views
  - maximize:

$$Q(\mathcal{C}) = \sum_{V_i \in \mathcal{V}} \left(V_i \cdot C_{c(i)}\right) - \alpha|\mathcal{C}|$$

$$Q(\mathcal{C}) = \sum_{V_i \in \mathcal{V}} \left(V_i \cdot C_{c(i)}\right) - \alpha|\mathcal{C}| - \beta \sum_{C_i \in \mathcal{C}} \sum_{C_{j>i} \in \mathcal{C}} \left(C_i \cdot C_j\right)$$

Greedy algorithm:

1. For each view $V \in \boldsymbol{V} \setminus \boldsymbol{C}$, compute $Q_V = Q(\boldsymbol{C} \cup \{V\}) - Q(\boldsymbol{C})$
2. Find the view V* for which Q_{V^*} is maximal
3. If $Q_{V^*} > 0$, add V* to $\boldsymbol{C}$ and repeat from step 1. Otherwise, stop.

Solution has quality at least (e - 1) / e times the optimal solution.

Image Browsing Application:

Organizing photos:

- construct 3-level hierarchy:
  1. top: scenes {***S***}
  2. middle: canonical views {***C***}
  3. bottom: set of images V $\in$ ***V*** s.t. ***C*** is the most similar canonical view to V

  - find connected components of the image collection
  - for each scene, use algorithm to compute canonical views

Displaying optional tag(s) for each view:

- compute which tag to display based on a function

$$\text{score}(c,t) = \sum_{u \in U} P(c|t,u)P(u)$$

  - measures h luster c
  - conditional probability of the cluster given the tag, independent of user who took the photo

$$P(c|t,u) = \frac{\left|\{V \in \mathcal{V} \mid c(V)=c, t \in T(V), u(V)=u\}\right|}{\left|\{V \in \mathcal{V} \mid t \in T(V), u(V)=u\}\right|}$$

$$P(u) = \frac{\left|\{V \in \mathcal{V} \mid u(V)=u\}\right|}{|\mathcal{V}|}$$

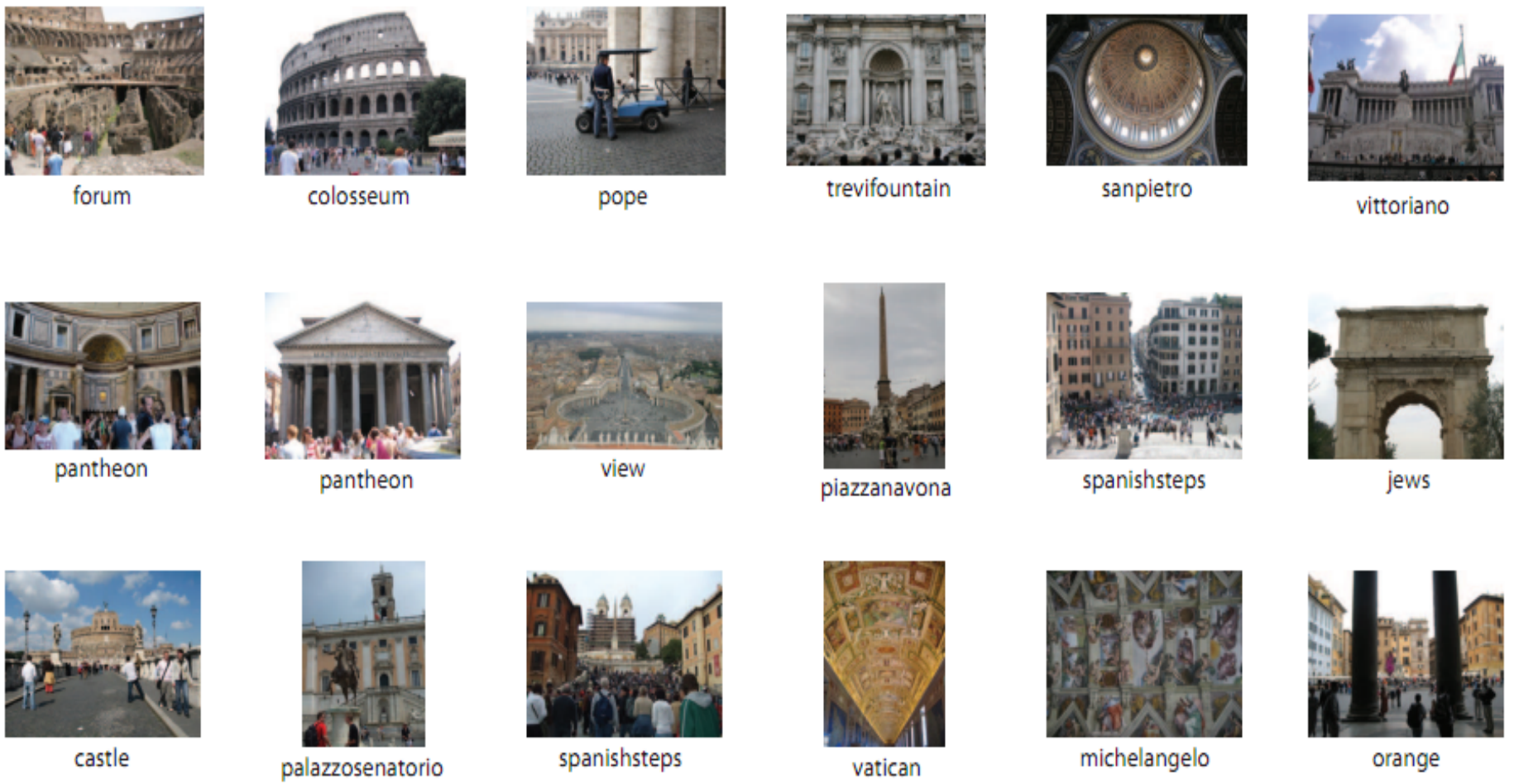

Figure 4. A segmentation of a 20,000 image Rome data set into the 18 largest scenes, with the best tag associated with each scene. The tags are computed according to Equation 5.

(a) Canonical views selected by the spherical k-means algorithm with $k = 6$.

(b) The output of our greedy k-means canonical views algorithm with $\alpha = 8$.

(c) The output of our greedy k-means algorithm with $\alpha = 5.75$ and orthogonality weight $\beta = 100$.

(d) All six photos from the Wikipedia [3] entry for the Pantheon, in order of appearance.

(e) Left to right: one Pantheon photo from the Berlitz [25] and Lonely Planet [18] guidebooks, and three from Fodor's [15]. These are the only images of the Panthon in the three guidebooks.

Results:

Web browser:
http://grail.cs.washington.edu/projects/canonview/

Enhanced 3D browsing:
<Play Video>