GEOMETRIC CONSISTENT VISUAL PHRASE MATCHING FOR PARTIAL DUPLICATE IMAGE DISCOVERY

Qian Zhang
Supervisor: Guoping Qiu
Outline

• Introduction
• TMH tree partition model
• Problem
• Geometric consistent visual phrase
• Experiment / Result
Introduction

**Partially duplicated image:** Image of the same scenes, buildings or objects taken from different angles, distance and vantage points.

**Image Discovery:** Find image groups that contain the same objects in image dataset. Different from object-based image retrieval

- Search all groups of partially duplicated images in image dataset.
- Don’t require any particular query image.
- More challenge.

**Application:** Manage large image repositories, Identify popular images, image clustering, Unsupervised object discovery
TMH tree partition model

Image Representation: bag of visual word

Classification Method: Min-hash hash been used to facilitate fast indexing and matching of images.

TMH tree partition: instead of hashing the whole image each time, we first cut image into numbers of partitions and then hashing each partition individually.
Step 1: Using Tree structure to hierarchically divide image into irregular partitions based on the density and appearance information of its interest points.

Step 2: Apply min-hash technique to each partition separately.

Step 3: Use all of the partitions as the processing primitives to vote in the similarity measure procedure.

\[
\text{SpreadRatio} = \frac{\text{Max}(\sigma_{\text{leafset}}, \sigma_{\text{rightset}})}{\text{Min}(\sigma_{\text{leafset}}, \sigma_{\text{rightset}})}
\]
Problem

Since min-hash technique measures the similarity between two images by counting the number of times they share an identical hash value, mismatching often occurs due to randomly returned hash value and inappropriate total number of times of hashing.

They share high similarity generated by min-hash function

True Matching
The straightforward way to enhance the accuracy of matching is to include geometric constraints and locality of visual words.

Problem: How to encode the spatial relationship among each visual word? Invariant to different transformation.

Observation: Under different transformations, although the value of visual words’ coordinates and the spatial relationship between pair of visual word may vary accordingly, the spatial distribution of locally neighbouring visual word set will be unchanged.
Our method—Loose circle spatial encoding

Circle Geometric Order Mapping:

We encode the geometric relationships between visual words by projecting them onto the perimeter of a circle centred on the geometric centroid of the whole visual word set. The $x$ and $y$ coordinates projected onto a circle can be encoded as scalar numbers.
To simplify the calculation, instead of calculating the coordinates of projected words on circle, we proposed to use angles to indicate visual words’ positions on the circle. The geometric order of the projected points equals to the geometric order of their angles.

\[
\theta_i = \begin{cases} 
\arccos \frac{\mathbf{v} \cdot \mathbf{p}_i}{\|\mathbf{v}\| \|\mathbf{p}_i\|} & \text{if } x_i \geq 0 \text{ and } y_i \geq 0 \\
\frac{\pi}{2} + \arccos \frac{\mathbf{v} \cdot \mathbf{p}_i}{\|\mathbf{v}\| \|\mathbf{p}_i\|} & \text{if } x_i < 0 \text{ and } y_i \geq 0 \\
\pi + \arccos \frac{\mathbf{v} \cdot \mathbf{p}_i}{\|\mathbf{v}\| \|\mathbf{p}_i\|} & \text{if } x_i < 0 \text{ and } y_i < 0 \\
\frac{3\pi}{2} + \arccos \frac{\mathbf{v} \cdot \mathbf{p}_i}{\|\mathbf{v}\| \|\mathbf{p}_i\|} & \text{if } x_i \geq 0 \text{ and } y_i < 0 
\end{cases}
\]

\[S = \{s_i | s_m \leq s_{m+1}, s_i \in \{\Theta\}, i = 1, 2, ..., m, m + 1, ..., n\}\]
Our method - Geometric consistent matching

We use order consistency $G$ to measure the consistency of geometric order of two image partitions. We denote query image as $I_Q$ and matching image as $I_M$, which share $d$ pairs of visual words. Their geometric order could be encoded as $SQ = \{sq_i\}$ and $SM = \{sm_i\}$, where $i = 1, 2, 3, \ldots, d$. Their geometric consistent could be calculated with:

$$g_i = \sum_j \delta_{i,j}, i = 1, 2, 3, \ldots, d, j = 1, 2, 3, \ldots, d$$

Where $\delta_{i,j}$ is the indicator function that measure the order consistency between each ordered pair item form SQ and SM

$$\delta_{i,j} = \begin{cases} 
0, & \text{if ordered pair } sq_i \text{ and } sq_j \text{ is consistent with ordered pair } sm_i \text{ and } sm_j \\
1, & \text{otherwise}
\end{cases}$$
Our method - Geometric consistent matching (cont.)

Order consistency before \( \mathbf{G} = \{g_c, g_b, g_a, g_d, g_f, g_e\} = \{0, 0, 3, 1, 1, 1\} \)

Remove false matching Visual word

Order consistency after \( \mathbf{G} = \{g_c, g_b, g_d, g_f, g_e\} = \{0, 0, 0, 0, 0\} \)

\[
\begin{align*}
g_c &= \delta_{cb} + \delta_{ca} + \delta_{cd} + \delta_{cf} + \delta_{ce} \\
&= 0 + 0 + 0 + 0 + 0 = 0 \\
g_b &= \delta_{bc} + \delta_{ba} + \delta_{bd} + \delta_{bf} + \delta_{be} \\
&= 0 + 0 + 0 + 0 + 0 = 0 \\
g_a &= \delta_{ac} + \delta_{ab} + \delta_{ad} + \delta_{af} + \delta_{ae} \\
&= 0 + 0 + 1 + 1 + 1 = 3 \\
g_d &= \delta_{dc} + \delta_{db} + \delta_{da} + \delta_{df} + \delta_{de} \\
&= 0 + 0 + 1 + 0 + 0 = 1 \\
g_f &= \delta_{fc} + \delta_{fb} + \delta_{fa} + \delta_{fd} + \delta_{fe} \\
&= 0 + 0 + 1 + 0 + 0 = 1 \\
g_e &= \delta_{ec} + \delta_{eb} + \delta_{ea} + \delta_{ed} + \delta_{ef} \\
&= 0 + 0 + 1 + 0 + 0 = 1
\]

\[
\begin{align*}
g_c &= \delta_{cb} + \delta_{cd} + \delta_{cf} + \delta_{ce} \\
&= 0 + 0 + 0 + 0 = 0 \\
g_b &= \delta_{bc} + \delta_{bd} + \delta_{bf} + \delta_{be} \\
&= 0 + 0 + 0 + 0 = 0 \\
g_d &= \delta_{dc} + \delta_{db} + \delta_{df} + \delta_{de} \\
&= 0 + 0 + 0 + 0 = 0 \\
g_f &= \delta_{fc} + \delta_{fb} + \delta_{fd} + \delta_{fe} \\
&= 0 + 0 + 0 + 0 = 0 \\
g_e &= \delta_{ec} + \delta_{eb} + \delta_{ed} + \delta_{ef} \\
&= 0 + 0 + 0 + 0 = 0
\]
Our method – applied to TmH

Output of each step:

- Divide Image into Partitions
- Apply Min-hash functions to Partitions
- Geometric consistent matching
- Vote in similarity measure
- Re-rank image list

Geometric consistent matching

<table>
<thead>
<tr>
<th>Image Partition</th>
<th>Image ID</th>
<th>Partition ID</th>
<th>Returned hash values (visual words)</th>
<th>Coordinates of returned hash values</th>
<th>Geometric Order</th>
</tr>
</thead>
</table>
Our method – applied to TmH (cont.)

Detail of how we match the geometric order between pair of image patches/partitions:

1. For two image patches/partitions Q and M, find their shared visual words from returned hash value list, and denote them as \( P = \{p_i, i = 1, 2, 3, \ldots, d\} \), \( d \) is the number of matching pairs.

2. Index geometric order of \( P \) from their corresponding geometric order SQ and SM to \( SQ_p = \{sq_{p_i}\} \) and \( SM_p = \{sm_{p_i}\} \). Note, there may exist rotation between two similar image partitions. To make their geometric orders comparable, we adjust them by first randomly picking a shared visual word as the geometric order start point, and then shifting all the ordered items in front of the start point to the end of the order list. As the geometric order is generated from a circle, the adjustment won’t influence the result of order consistency.

3. Compute their order consistency \( G \) as introduced in above. Continuously identify and remove the index of false matching visual words from \( SQ_p \) and \( SM_p \).
We give each image partition a weight to represent its importance. Let $W(q;m)$ be the similarity voting weight of image partition pair $q$ and $m$, it could be computed by combining both visual weight and geometric weight:

$$W(q;m) = f(W_v(q;m); W_g(q;m))$$

$$f(W_v(q;m); W_g(q;m)) = N_v \cdot \sqrt{N_g}$$

Where $N_v$ is the number of shared visual word returned from min-hash and the $N_g$ is the number of visual words remained after matching the spatial code. Hence, the similarity of two images could be calculated as

$$Sim(I_1, I_2) = \sum_{i,j} w(q_i; m_j),$$
Experiment and Result

(a) 
(b) 
(c)
### Experiment and Result (cont.)

|               | Standard Min-hash | RANSAC | Our Method | SP $r = 1$ | SP $r = 2$ | SP $r = 3$ | SP $r = 4$
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>mAP</td>
<td>0.46</td>
<td>0.54</td>
<td>0.57</td>
<td>0.54</td>
<td>0.55</td>
<td>0.55</td>
<td>0.54</td>
</tr>
<tr>
<td>mAP*</td>
<td>0.46</td>
<td>0.54</td>
<td>0.56</td>
<td>0.20</td>
<td>0.28</td>
<td>0.32</td>
<td>0.37</td>
</tr>
<tr>
<td>T(s)</td>
<td>0.08</td>
<td>7.08</td>
<td>0.76</td>
<td>0.86</td>
<td>0.84</td>
<td>1.22</td>
<td>1.45</td>
</tr>
<tr>
<td>T*(s)</td>
<td>0.08</td>
<td>7.10</td>
<td>0.85</td>
<td>1.04</td>
<td>1.27</td>
<td>1.85</td>
<td>2.45</td>
</tr>
</tbody>
</table>

### (a) Oxford dataset

|               | Standard Min-hash | RANSAC | Our Method | SP $r = 1$ | SP $r = 2$ | SP $r = 3$ | SP $r = 4$
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>mAP</td>
<td>0.64</td>
<td>0.70</td>
<td>0.66</td>
<td>0.68</td>
<td>0.66</td>
<td>0.65</td>
<td>0.63</td>
</tr>
<tr>
<td>mAP*</td>
<td>0.64</td>
<td>0.70</td>
<td>0.65</td>
<td>0.17</td>
<td>0.27</td>
<td>0.33</td>
<td>0.45</td>
</tr>
<tr>
<td>T(s)</td>
<td>0.06</td>
<td>6.06</td>
<td>0.38</td>
<td>0.21</td>
<td>0.35</td>
<td>0.53</td>
<td>0.66</td>
</tr>
<tr>
<td>T*(s)</td>
<td>0.06</td>
<td>6.40</td>
<td>0.43</td>
<td>0.44</td>
<td>0.72</td>
<td>1.07</td>
<td>1.42</td>
</tr>
</tbody>
</table>

### (b) Dup dataset

|               | Standard Min-hash | Our Method | SP $r = 1$ | SP $r = 2$ | SP $r = 3$ | SP $r = 4$
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>top 4 score</strong></td>
<td>3.07</td>
<td>3.16</td>
<td>3.01</td>
<td>3.11</td>
<td>3.13</td>
<td>3.14</td>
</tr>
<tr>
<td><strong>(top 4 score)</strong></td>
<td>3.07</td>
<td>3.14</td>
<td>1.98</td>
<td>2.53</td>
<td>2.70</td>
<td>2.88</td>
</tr>
<tr>
<td>T(s)</td>
<td>0.04</td>
<td>0.17</td>
<td>0.19</td>
<td>0.20</td>
<td>0.24</td>
<td>0.29</td>
</tr>
<tr>
<td>T*(s)</td>
<td>0.04</td>
<td>0.18</td>
<td>0.19</td>
<td>0.21</td>
<td>0.25</td>
<td>0.32</td>
</tr>
</tbody>
</table>

### (c) Kentucky dataset