An Interactive Approach for CBIR Using a Network of Radial Basis Functions

by P. Muneesawang and L. Guan
IEEE Transactions on Multimedia, 2004

Joseph Lee
Contents

1. Background
2. General Framework
3. Previous Models
4. Proposed RBF Model
5. Learning Strategy
6. Experimental Results
7. Conclusion
1. Background

- Content-based Image Retrieval (CBIR)
  - use colors, shapes, textures → rely on image itself
  - without image content → rely on metadata
1. Background

- **Content-based Image Retrieval (CBIR)**
  - use **colors, shapes, textures** → rely on image itself
  - without image content → rely on metadata

- **Relevance Feedback**
  - User refines the result by marking images as **relevant** or **non-relevant** and repeats search.
1. Background

- Image Similarity
1. Background

- Image Similarity
1. Background

- Examples of Image Similarity Problems

MARS-1

RBF
2. General Framework

- Ideal Case
2. General Framework

- Ideal Case – User Query
2. General Framework

- Ideal Case – Similarity Measure
2. General Framework

- Real Case
2. General Framework

- Real Case – Inaccurate Query
2. General Framework

- Real Case – Inaccurate Query
2. General Framework

- We should refine Query
2. General Framework

- We should refine Query & Metric
3. Previous Models

- Query Reformulation Model
  - Relevance feedback → learn query representation

- Adaptive Metric Model
  - Relevance feedback → learn similarity function
3. Previous Models

- Learning user perception

\[ y_s = f(x) \]
3. Previous Models

- Learning user perception
  
  \[ y_s = f(x) \]

- MARS-1
  
  \[ y_s = f_{\text{cosine}}(x, x_q) \]

  \[ x_q = \alpha x_q + \gamma (\text{mean}_{i=1}^l \{x_i\}) - \varepsilon (\text{mean}_{i=0}^l \{x_i\}) \]
3. Previous Models

- Learning user perception
  \[ y_s = f(x) \]
  - MARS–1
    \[ y_s = f_{\text{cosine}}(x, x_\hat{q}) \]
    \[ x_\hat{q} = \alpha x_q + \gamma (\text{mean}\{x_i\}) - \varepsilon (\text{mean}\{x_i\}) \]
  - MARS–2
    \[ y_s = f(x, x_q) = (x - x_q)^T W (x - x_q) \]
3. Previous Models

- Learning user perception
  
  \[ y_s = f(x) \]

  \[ y_s = f_{\text{cosine}}(x, x_q) \]
  \[ x_q = \alpha x_q + \gamma (\text{mean}\{x_i\}) - \varepsilon (\text{mean}\{x_i\}) \]

  \[ y_s = f(x, x_q) = (x - x_q)^T W (x - x_q) \]

  \[ y_s = f(x, x_q) = (x - x_q)^T W (x - x_q) \]
  \[ x_q = \frac{X^T v}{\sum_{i=1}^{N} v_i} \]
3. Previous Models

- Learning user perception
  \[ y_s = f(x) \]

  - MARS-1
    \[ y_s = f_{\text{cosine}}(x, x_{\hat{q}}) \]
    \[ x_{\hat{q}} = \alpha x_q + \gamma (\text{mean}\{x_i\}) - \varepsilon (\text{mean}\{x_i\}) \]

  - MARS-2
    \[ y_s = f(x, x_q) = (x - x_q)^T W (x - x_q) \]

  - OPT-RF
    \[ y_s = f(x, x_q) = (x - x_q)^T W (x - x_q) \]
    \[ x_{\hat{q}} = \frac{X^Tv}{\sum_{i=1}^{N} v_i} \]

- Assumption
  - same distance gives same degree of similarity
4. Proposed RBF Model

- CBIR
  - online learning
  - two-class problem

\[ f(x) = \sum_{j=1}^{N} w_j G(x, z_j) \]
\[ = \sum_{j=1}^{N} w_j \exp \left( -\frac{1}{2\sigma_j^2} \sum_{i=1}^{P} (x_i - z_{ji})^2 \right) \]
4. Proposed RBF Model

- CBIR
  - online learning
  - two-class problem

\[ f(x) = \sum_{i=1}^{P} G_i(x_i, z_i) \]
\[ = \sum_{i=1}^{P} \exp \left( -\frac{(x_i - z_i)^2}{2\sigma_i^2} \right) \]
5. Learning Strategy

- Example Vectors
5. Learning Strategy

- Voronoi Vectors
5. Learning Strategy

- Voronoi Cells
5. Learning Strategy

- Learning Vector Quantization
5. Learning Strategy

- Learning Vector Quantization
5. Learning Strategy

- Learning Vector Quantization

\[ z_c(n + 1) = z_c(n) + \alpha_n [x(n) - z_c(n)] \]
5. Learning Strategy

- Learning Vector Quantization

\[ z_c(n + 1) = z_c(n) + \alpha_n [x_i(n) - z_c(n)] \]
\[ z_c(n + 1) = z_c(n) - \alpha_n [x_i(n) - z_c(n)] \]
5. Learning Strategy

- Modified LVQ (Model 1)

\[
\begin{align*}
  z_c(n + 1) &= z_c(n) + \alpha_n [x_i(n) - z_c(n)] \\
  z_c(n + 1) &= z_c(n) - \alpha_n [x_i(n) - z_c(n)]
\end{align*}
\]
5. Learning Strategy

- **Modified LVQ (Model 1)**

\[
\begin{align*}
z_c(n+1) &= z_c(n) + \alpha_n [x_i(n) - z_c(n)] \\
z_c(n+1) &= z_c(n) - \alpha_n [x_i(n) - z_c(n)] \\
z_q(t+1) &= z_q(t) + \alpha_R (\bar{x}' - z_q(t)) - \alpha_N (\bar{x}'' - z_q(t))
\end{align*}
\]
5. Learning Strategy

- Modified LVQ (Model 1)

\[ z_c(n+1) = z_c(n) + \alpha_n [x_i(n) - z_c(n)] \]
\[ z_c(n+1) = z_c(n) - \alpha_n [x_i(n) - z_c(n)] \]
\[ z_q(t+1) = z_q(t) + \alpha_R (\bar{x}' - z_q(t)) - \alpha_N (\bar{x}'' - z_q(t)) \]
5. Learning Strategy

- Modified LVQ (Model 1)

\[ z_c(n+1) = z_c(n) + \alpha_n[x_i(n) - z_c(n)] \]
\[ z_c(n+1) = z_c(n) - \alpha_n[x_i(n) - z_c(n)] \]
\[ z_q(t+1) = z_q(t) + \alpha_R(\bar{x}' - z_q(t)) - \alpha_N(\bar{x}'' - z_q(t)) \]
5. Learning Strategy

- **Modified LVQ (Model 2)**

\[
\begin{align*}
\mathbf{z}_c(n + 1) &= \mathbf{z}_c(n) + \alpha_n [\mathbf{x}_i(n) - \mathbf{z}_c(n)] \\
\mathbf{z}_c(n + 1) &= \mathbf{z}_c(n) - \alpha_n [\mathbf{x}_i(n) - \mathbf{z}_c(n)] \\
\mathbf{z}_q(t + 1) &= \mathbf{z}_q(t) + \alpha_R (\bar{\mathbf{x}}' - \mathbf{z}_q(t)) - \alpha_N (\bar{\mathbf{x}}'' - \mathbf{z}_q(t)) \\
\mathbf{z}_q(t + 1) &= \bar{\mathbf{x}}' - \alpha_N (\bar{\mathbf{x}}'' - \mathbf{z}_q(t))
\end{align*}
\]
5. Learning Strategy

- Effects of Positive and Negative Learning
  - Relevant samples → common interest
  - Non-relevant samples → specific interest
5. Learning Strategy

- Effects of Positive and Negative Learning
  - Relevant samples → common interest
  - Non-relevant samples → specific interest
5. Learning Strategy

- Effects of Positive and Negative Learning
  - Relevant samples → common interest
  - Non-relevant samples → specific interest
5. Learning Strategy

- Effects of Positive and Negative Learning
  - Relevant samples → common interest
  - Non-relevant samples → specific interest
5. Learning Strategy

- Selection of RBF Width
5. Learning Strategy

- Selection of RBF Width
5. Learning Strategy

- Selection of RBF Width

\[ \sigma_i = \eta \max_{\text{all}} |x'_m - z_i| \]

\[ \sigma_i = \exp(\beta \cdot \text{Std}_i) \]
6. Experimental Results

<table>
<thead>
<tr>
<th>Method</th>
<th>t=0</th>
<th>t=1</th>
<th>t=2</th>
<th>t=3</th>
<th>CPU time (second per iteration)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBF</td>
<td>44.82</td>
<td>79.82</td>
<td>88.75</td>
<td>91.79</td>
<td>2.34</td>
</tr>
<tr>
<td>MARS-2</td>
<td>44.82</td>
<td>60.18</td>
<td>61.61</td>
<td>61.96</td>
<td>1.26</td>
</tr>
<tr>
<td>OPT-RF</td>
<td>44.82</td>
<td>72.14</td>
<td>79.64</td>
<td>80.54</td>
<td>1.27</td>
</tr>
<tr>
<td>Simple CBIR</td>
<td>44.82</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.90</td>
</tr>
</tbody>
</table>

Average Precision Rate (%)
6. Conclusion

- **Non-linear** model for similarity evaluation
- Learning from **positive and negative** samples
References

- http://videolectures.net/minh_hoai_nguyen/
- http://www.mqasem.net/vectorquantization/vq.html