Image Data Similarity Search
Image Data Similarity Search
Obligatory “Moore's Law” Slide:

Magnetic storage capacity has been increasing at a rate faster than Moore's Law (about 2x per year)

The increasing availability (and decreasing cost) of storage has allowed for a huge amount of rich media (images, sound, video) to be archived

Efficient access to this data will become an increasingly difficult problem as archives grow in size
Example:

Suppose you have a 1 Terra-byte disk (approximately 2 years in the future). This disk could hold approximately 80,000 4MP images (uncompressed). A human-powered linear scan through this archive (4 images per second) would require ~ 6 hours.
Conclusion:

*Use computers to solve the problem!*
Conclusion:

*Use computers to solve the problem!*

...easier said than done.
Goal:

Given a target image (query), find all images in the database that are “similar” to the query.
Goal:

Given a target image (query), find all images in the database that are “similar” to the query.
Image Data Similarity Search

Goal:

Given a target image (query), find all images in the database that are “similar” to the query.
Annotation:

For each image, manually associate keywords that describe the image

Use traditional text-based retrieval mechanisms to search for similarity

(“tree”)
Annotation:

Text-based retrieval has received a large amount of research and innovation in precision, recall, and efficiency.
Annotation:

("tree")
Image Data Similarity Search

Annotation:

("tree")

("tree")

Figure: Tree data structure

("tree")
Annotation:

Labels of images will always be both imprecise and subjective, due to the differences in perception between various users.

Additionally, annotating a large amount of images requires many hours of tedious labor. With large image sets, this may even be a near-intractable task.
Content-based:

Index images based upon their data

Automated

Objective

Useful?
Image Data Similarity Search

Sensory Gap:

cultural

transitional

geometric

perceptual

literal

physical

categorical
Image Data Similarity Search

Sensory Gap:

literal
Sensory Gap:

- perceptual
- literal
Sensory Gap:

- geometric
- perceptual
- literal
Sensory Gap:

- Physical
- Perceptual
- Literal
Sensory Gap:

- Categorical
- Geometric
- Perceptual
- Literal
- Physical
Image Data Similarity Search

Sensory Gap:

- Cultural
- Geometric
- Perceptual
- Literal
- Physical
- Categorical
Color:

Tristimulus Theory of color perception gives a natural representation for color:

\[ C_x = (R_x, G_x, B_x) \]

This representation is derived from the fact that the human eye has cells receptive to specific wavelengths:

- 580 nm (red)
- 545 nm (green)
- 440 nm (blue)
Image Data Similarity Search

Color:
Color:

However, RGB color description is far from ideal:

RGB distances between colors is not perceptually uniform metric
Color:

However, RGB color description is far from ideal:

RGB distances between colors is not perceptually uniform metric

(127,0,0)  \rightarrow  128*(1,0,0)  \rightarrow  (255,0,0)
Color:

However, RGB color description is far from ideal:

RGB distances between colors is not perceptually uniform metric

- (127,0,0) → 128*(1,0,0) → (255,0,0)
- (127,0,0) → 128*(-√2,0,√2) → (37,0,90)
Color:

However, RGB color description is far from ideal:

RGB distances between colors is not perceptually uniform metric

RGB is heavily dependent upon lighting and viewing conditions
Image Data Similarity Search

Color:

CIE L*a*b: Luminance, Green-Red, Blue-Yellow

Frank Battaglia
Color:

CIE L*a*b: Luminance, Green-Red, Blue-Yellow

Perceptually uniform (distances in L*a*b are linear with perceived difference in color)
Color:

HSV: Hue, Saturation, Value
Image Data Similarity Search

Color:

HSV: Hue, Saturation, Value

The axes of HSV map to a more natural set of parameters

Hue is invariant relative to object orientation (for most objects)
Color Histograms:

Histograms express the distribution of color over a collection of pixels (image or region)
Color Histograms:

- Histograms express the distribution of color over a collection of pixels (image or region).

- Histograms from different sources can be compared for similarity using the $L_2$ difference of each channel.

- However, quantization error can cause histograms of similar images to have a larger $L_2$ distance than is perceptually meaningful.
Color Moments:

The histogram can be described by statistical “moments”, where the $n^{th}$ moment is expressed as

$$\mu_n(a) = \langle (x-a)^n \rangle$$

$$\mu_n(a) = \frac{1}{N} \sum_i (x_i - a)^n$$
Color Moments:

The histogram can be described by statistical “moments”, where the $n^{th}$ moment is expressed as

$$\mu_n(a) = \langle (x - a)^n \rangle$$

$$\mu_n(a) = \frac{1}{N} \sum_i (x_i - a)^n$$

- $\mu_1(0) = \mu_1' := \text{mean}$
- $\mu_2(\mu_1') := \text{variance}$
- $\mu_3(\mu_1') := \text{skew}$
Color Moments:

Compact representation of histograms (3 numbers per color channel)

More robust against quantization error

Simple dissimilarity metric:

\[ D(h_a, h_b) = w_1 |\mu_1'_{a} - \mu_1'_{b}| + w_2 |\mu_2_{a} - \mu_2_{b}| + w_3 |\mu_3_{a} - \mu_3_{b}| \]
Texture:

“Visual patterns the have properties of homogeneity that do not result from the presence of only a single color or intensity” (Rui, Huang, Chang 1999)
Image Data Similarity Search

Texture:

- Grass
- Herringbone pattern
- Leopard print
Image Data Similarity Search

Texture:

Psychologically meaningful parameters:

- Coarseness
- Contrast
- Directionality
- Line-like
- Regularity
- Roughness
Texture:

Texture can also be analyzed with wavelets

Similar textures possess similarities in the wavelet subbands
Segmentation:

For a given object, it is assumed that color and texture properties will conform to a certain degree of homogeneity.

Using this assumption, the image can be divided into a set of homogeneous regions such that each region corresponds to a single object.

A single object may correspond to several regions.
Image Data Similarity Search

Segmentation:

1) partition has to cover the whole image
2) each region has to be homogeneous
3) two adjacent region cannot be merged into a single homogeneous region

(Lucchese, Mitra 2001)
Image Data Similarity Search

Segmentation:
Image Data Similarity Search

Segmentation:
Image Data Similarity Search

Segmentation:

Techniques employ:

- Clustering pixels (K-means, etc)
- Region Growing
- Edge Detection

Varying degrees of success
Image Data Similarity Search

Shape:

Rotation invariant?

Translation invariant?

Scaling invariant?
Fourier Descriptors:

Express the shape as a parametric curve

\[(x(l), y(l)) = Z(l), 0 \leq l \leq L\]

Denote the angular direction at point \(l\) be \(\theta(l)\)
Fourier Descriptors:

Let $\phi(l)$ be the net angular difference between $\theta(l)$ and $\theta(0)$.
Fourier Descriptors:

\[ \phi^* (t) = \phi \left( \frac{L t}{2 \pi} \right) + t, \quad 0 \leq t \leq 2 \pi \]
Fourier Descriptors:

\[ \phi(l) \]

\[ \phi^*(l) \]
Fourier Descriptors:

$$
\phi^* (t) = \mu_0 + \sum_{k=1}^{\infty} A_k \cos (kt - a_k)
$$
Fourier Descriptors:

- Compact representation for shape
- Rotation can be factored out (phase angles)
- Scale can be factored out ($L$)
- Translation is not included in this representation
Image Data Similarity Search

Implementation:
Implementation:
Image Data Similarity Search

Segmentation:

Image $\rightarrow$ Segmentation $\rightarrow$ Segmented image
Feature Extraction:

Region $l$ -> Feature extraction -> 14-D


Region $n$ -> Feature extraction -> 14-D
Feature Extraction:

For each region in the image, calculate:

- 3 Color Moments for each channel (H, S, V)
  - Mean
  - Variance
  - Skew
Feature Extraction:

For each region in the image, calculate:

3 Color Moments for each channel (H, S, V)

Geometric information
  width
  height
  # pixels
  aspect ratio
  centroid \((c_x, c_y)\)
Feature Extraction:

For each region in the image, calculate:

3 Color Moments for each channel (H, S, V)

Geometric information

log(aspect ratio)
log(width*height)

# pixels / (width*height)

\( c_X \)
\( c_Y \)
Feature Extraction:

For each region in the image, calculate:

3 Color Moments for each channel (H, S, V)

Geometric information

...14-dimensional vector for each region
Bit Vector Conversion:

Bit vector conversion \( \rightarrow \) Rgn. bit vec.

Weight \( w_1 \)

\( \cdots \)

\( \cdots \)

\( \cdots \)

Bit vector conversion \( \rightarrow \) Rgn. bit vec.

Weight \( w_n \)
Bit Vector Conversion:

Use the 14-dimensional, real-valued vector to create a N-bit vector

Hamming distance of bit vector approximates $L_1$ distance between real-valued vectors

Significant savings in storage space as well as computation speed
Bit Vector Conversion:

Let the $i^{th}$ dimension be in the range $[l_i, h_i]$ and have weight $w_i$

$$T = \sum_i w_i \times (h_i - l_i)$$

$$p_i = \frac{w_i \times (h_i - l_i)}{T}$$

Pick $i : [0, d-1]$ with probability $p_i$
Pick $t : [l_i, h_i]$
Bit Vector Conversion:

Pick $i : [0, d-1]$ with probability $p_i$
Pick $t : [l_i, h_i]$

\[
bit = \begin{cases} 
  0 & \text{if } v_i < t \\
  1 & \text{if } v_i \geq t
\end{cases}
\]
Bit Vector Conversion:

Lemma 1: \( \| u - v \|_{L,w} = x \Rightarrow Pr(\text{bit}(u) \neq \text{bit}(v)) = x/T \)

Proof:

\[
Pr(\text{bit}(u) \neq \text{bit}(v)|C_i) = \frac{\| u_i - v_i \|_{L_i}}{r_i}
\]

\[
Pr(\text{bit}(u) \neq \text{bit}(v)) = \sum_{i=0}^{d-1} Pr(\text{bit}(u) \neq \text{bit}(v)|C_i) \times Pr(C_i)
\]

\[
Pr(\text{bit}(u) \neq \text{bit}(v)) = \sum_{i=0}^{d-1} \frac{\| u_i - v_i \|_{L_i}}{r_i} \times \frac{w_i \times r_i}{T}
\]

\[
Pr(\text{bit}(u) \neq \text{bit}(v)) = \sum_{i=0}^{d-1} w_i \times \| u_i - v_i \|_{L_i}/T
\]

\[
Pr(\text{bit}(u) \neq \text{bit}(v)) = x/T
\]
Bit Vector Conversion:

XOR groups of K bits to produce a single bit

101010 → 1
100100 → 0
111010 → 0
100101 → 1
Bit Vector Conversion:

Lemma 2: \( \|u - v\|_{L_1} = x \Rightarrow Pr \left( h_K(u) \neq h_K(v) \right) = 0.5 \left( 1 - \left( 1 - \frac{2x}{T} \right)^K \right) = q \)

Proof:

\[
q = \sum_{\text{odd } j} \binom{K}{j} p^j (1-p)^{K-j}
\]

\[
q = \frac{1}{2} \sum_j \binom{K}{j} p^j (1-p)^{K-j} - \frac{1}{2} \sum_j (-1)^j \binom{K}{j} p^j (1-p)^{K-j}
\]

\[
q = \frac{1}{2} \left( 1 - \left( 1 - 2p \right)^K \right) = 0.5 \left( 1 - \left( 1 - \frac{2x}{T} \right)^K \right)
\]
Image Data Similarity Search

Embedding:

[Diagram of the embedding process with boxes labeled Embedding, Bit vector conversion, Filtering, and Top K.

Store is connected to the Filtering block, and the Image bit vector DB is connected to Filtering.

Frank Battaglia]
Embedding:

Even further!

In large databases, it would be beneficial to filter images during the search so comparisons are not performed on images that are very dissimilar.

A fast, compact representation for *images* that approximates EMD.
**Embedding:**

An image is a set of regions (bit vectors)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>( w_r )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( r_1 )</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.1</td>
</tr>
<tr>
<td>( r_2 )</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.6</td>
</tr>
<tr>
<td>( r_3 )</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.3</td>
</tr>
</tbody>
</table>
Embedding:

Select a random set of positions \( \{p_1, ..., p_n\} \) as well as a random set of bits \( \{b_1, ..., b_n\} \). Together these for pattern \( P = \{(p_1, b_1), ..., (p_n, b_n)\} \)

A given region \( r \) matches pattern \( P \) if

\[
    r_{p_j} = b_j \quad \text{for } j = 1, 2, ..., h
\]

with \( r_{p_i} \) signifying the \( p_j^{th} \) bit in the bit vector representing \( r \)
Embedding:

For each region in an image, determine if the region matches $P$

If so, add the region's weight to the matched weight for the image
Image Data Similarity Search

Embedding:

Let $P = \{(3, 0), (5, 1), (7, 1)\}$

\[
\begin{array}{cccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & w_i \\
\hline
r_1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0.1 \\
r_2 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0.6 \\
r_3 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0.3 \\
\end{array}
\]

$MW(p)$

0.4
Repeat this process with several distinct patterns \( \{P_1, P_2, \ldots, P_m\} \). If the regions in two images are highly similar, the two images will tend to receive the same \( MW(P_i) \) for various \( P \).

Any images with very different MW vectors are likely dissimilar and can be filtered out before the EMD* matching.