Image Data Similarity Search

Frank Battaglia

Image Data Similarity Search

Frank Battaglia

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

Image Data Similarity Search

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.

Image Data Similarity Search

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.

Image Data Similarity Search

Goal:

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

Frank Battaglia

Image Data Similarity Search

Annotation:

For each image, manually associate keywords that describe the image

Use traditional text-based retrieval mechanisms to search for similarity

("tree")

Image Data Similarity Search

Annotation:

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

("tree")

Image Data Similarity Search

Annotation:

("tree")

Image Data Similarity Search

Annotation:

("tree")

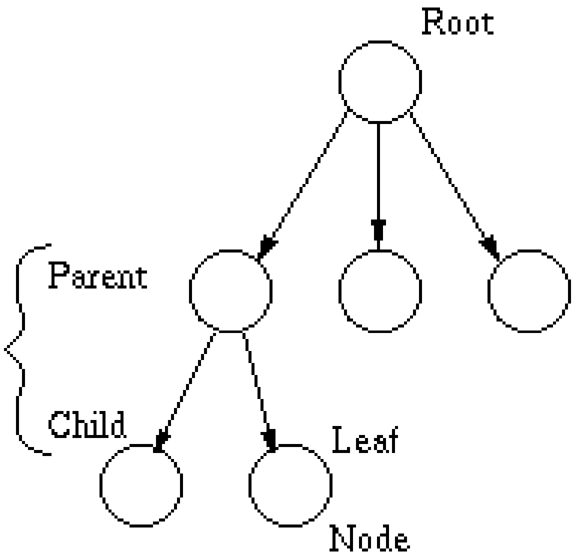


Figure: Tree data structure

("tree")

Image Data Similarity Search

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.

Image Data Similarity Search

Content-based:

Index images based upon their data

Automated

Objective

Useful?

Image Data Similarity Search

Sensory Gap:

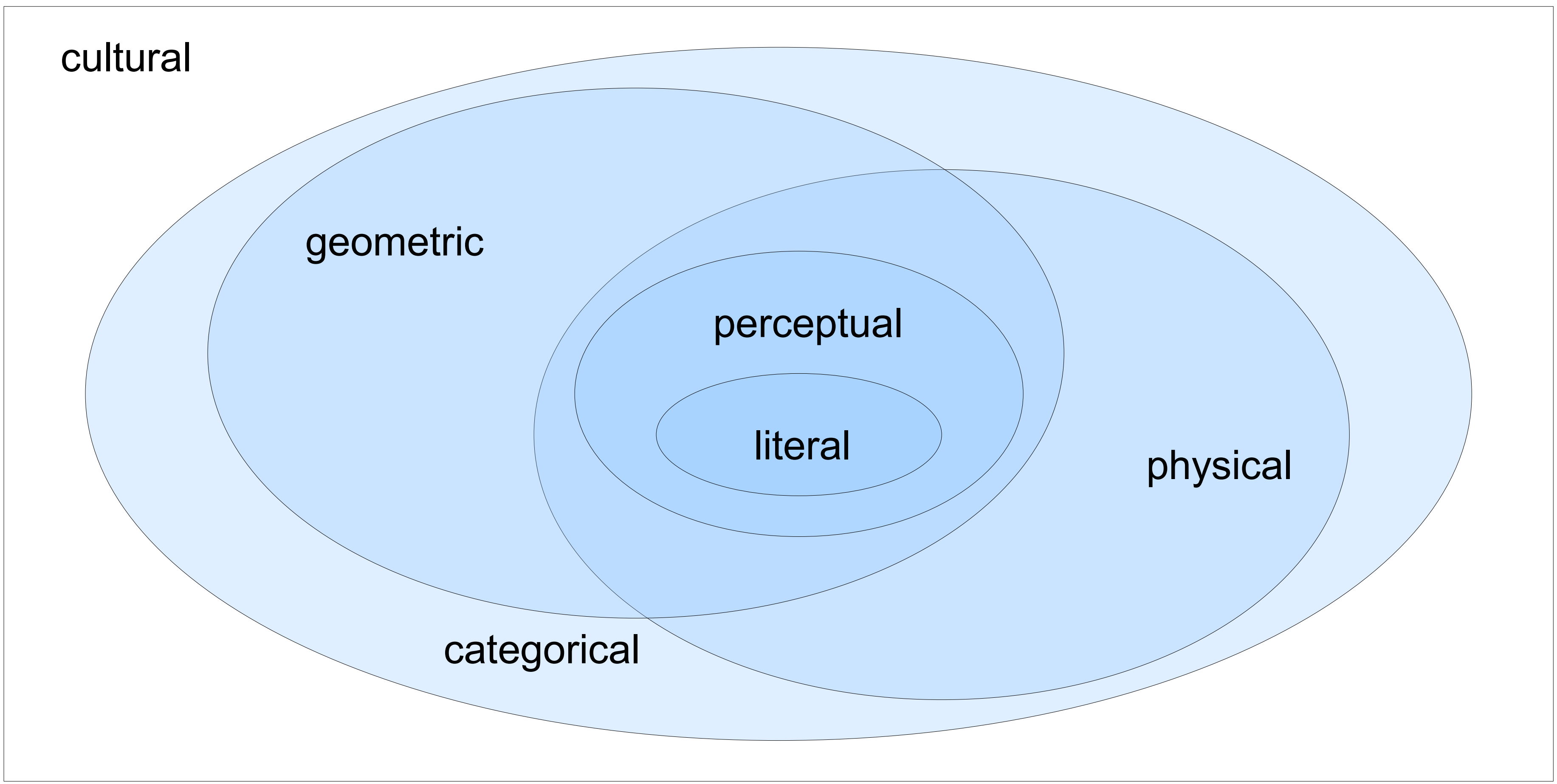


Image Data Similarity Search

Sensory Gap:

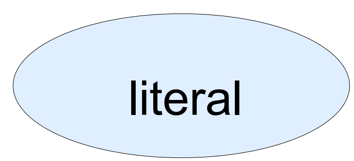
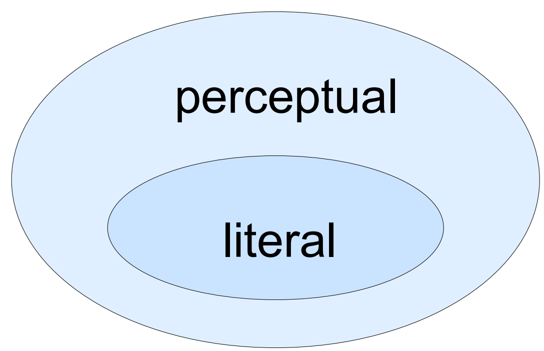


Image Data Similarity Search

Sensory Gap:



Frank Battaglia

Image Data Similarity Search

Sensory Gap:

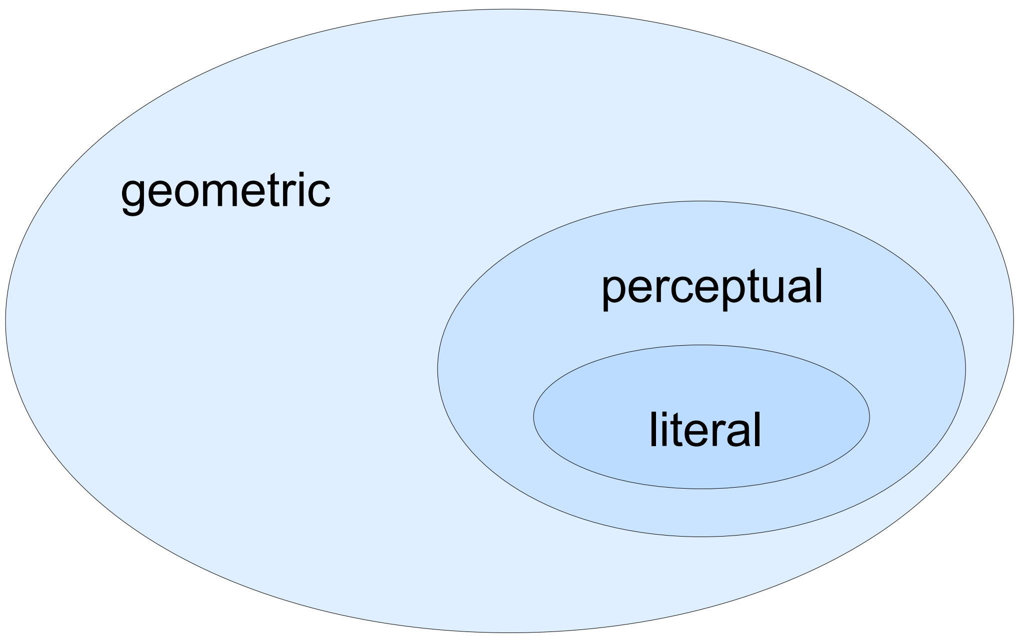


Image Data Similarity Search

Sensory Gap:

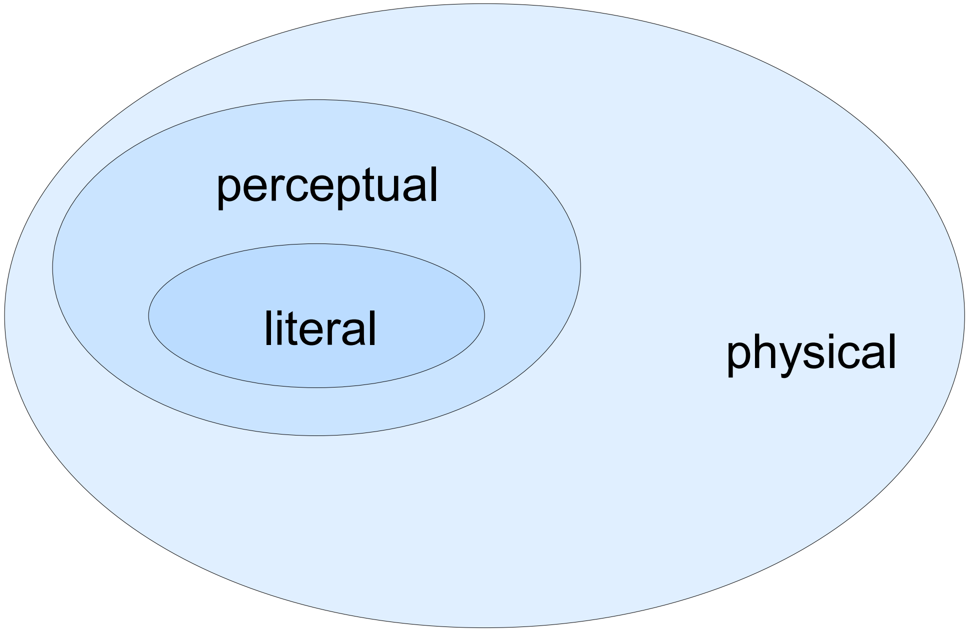


Image Data Similarity Search

Sensory Gap:

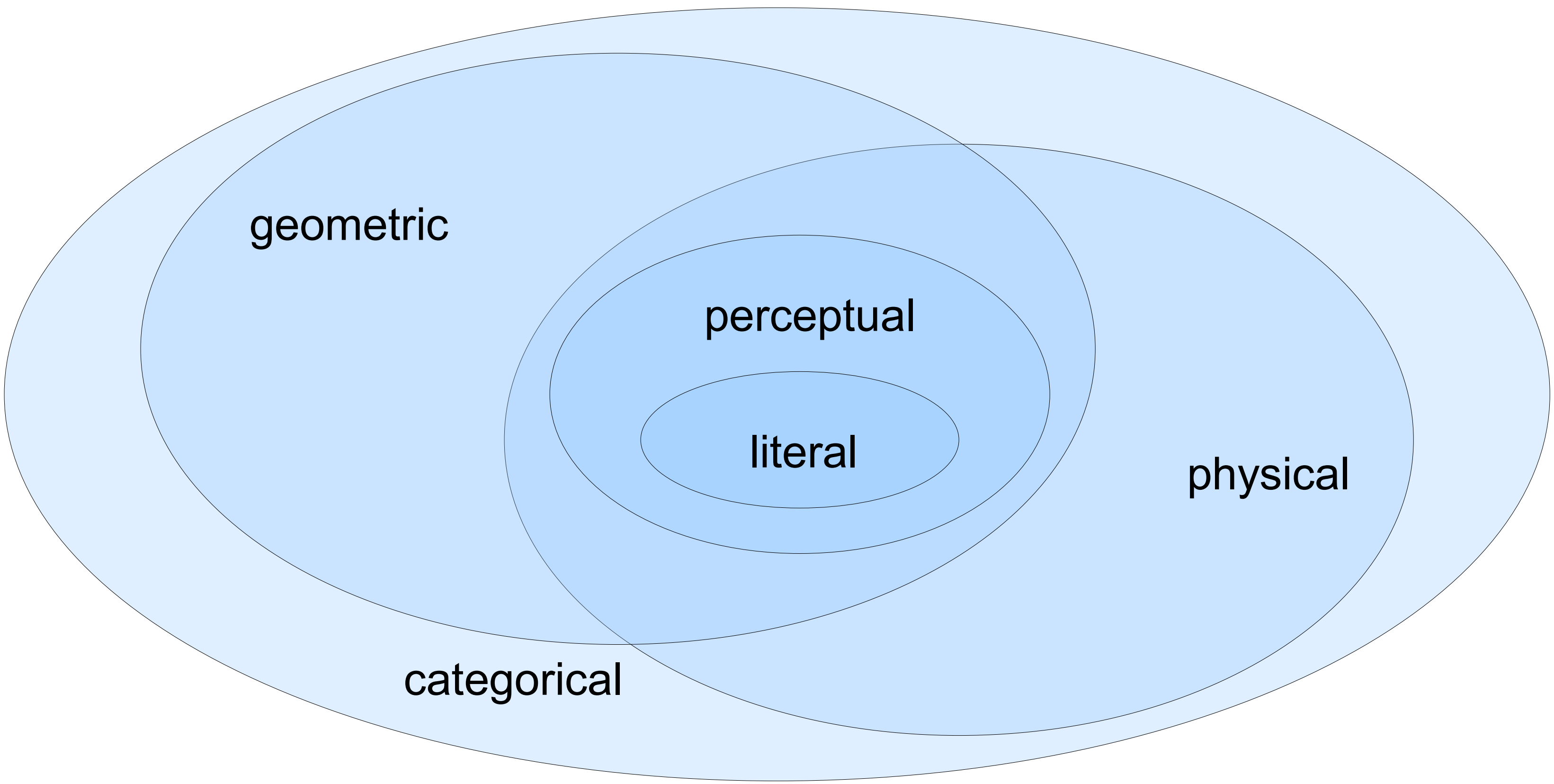
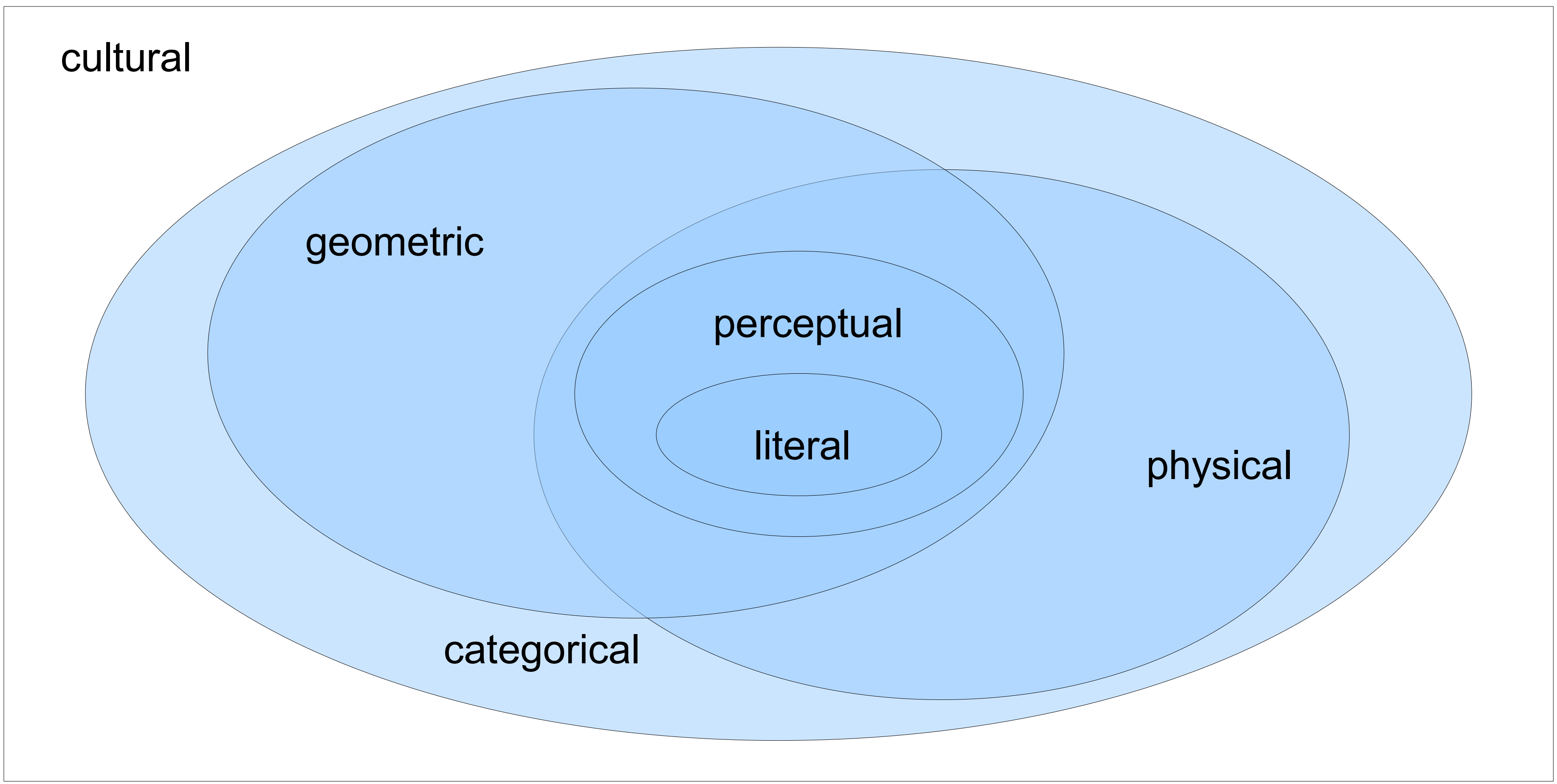


Image Data Similarity Search

Sensory Gap:



Frank Battaglia

Image Data Similarity Search

Color:

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

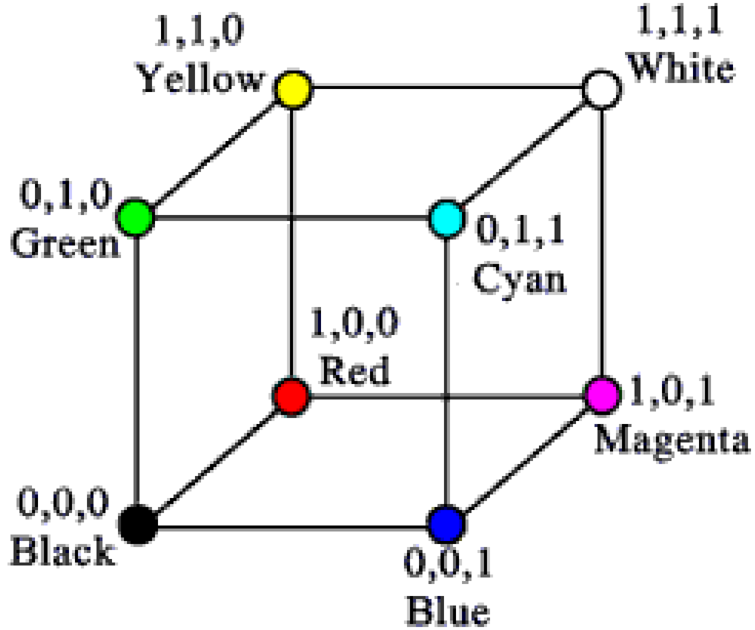
$$C_x = (R_x, G_x, B_x)$$

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:



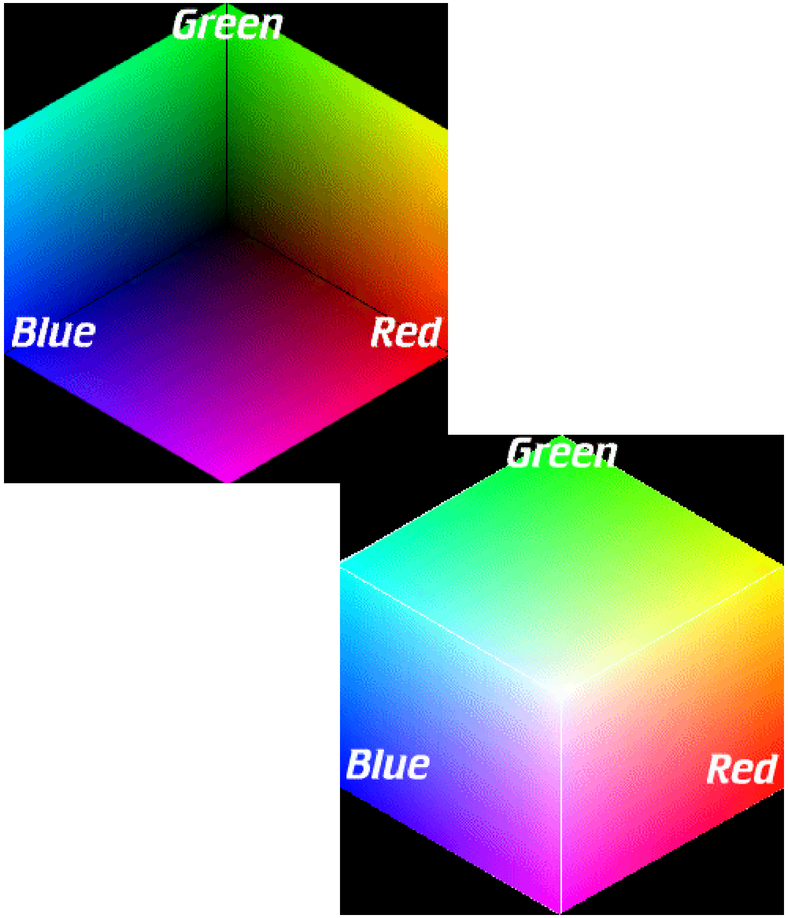


Image Data Similarity Search

Color:

However, RGB color description is far from ideal:

RGB distances between colors is not perceptually uniform metric

Image Data Similarity Search

Color:

However, RGB color description is far from ideal:

RGB distances between colors is not perceptually uniform metric

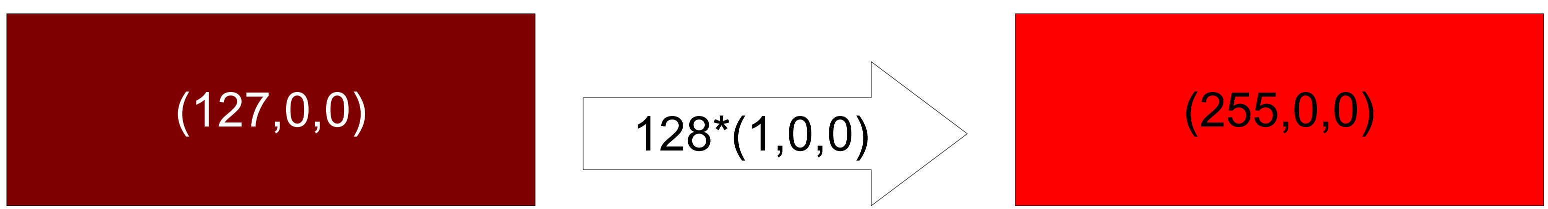


Image Data Similarity Search

Color:

However, RGB color description is far from ideal:

RGB distances between colors is not perceptually uniform metric

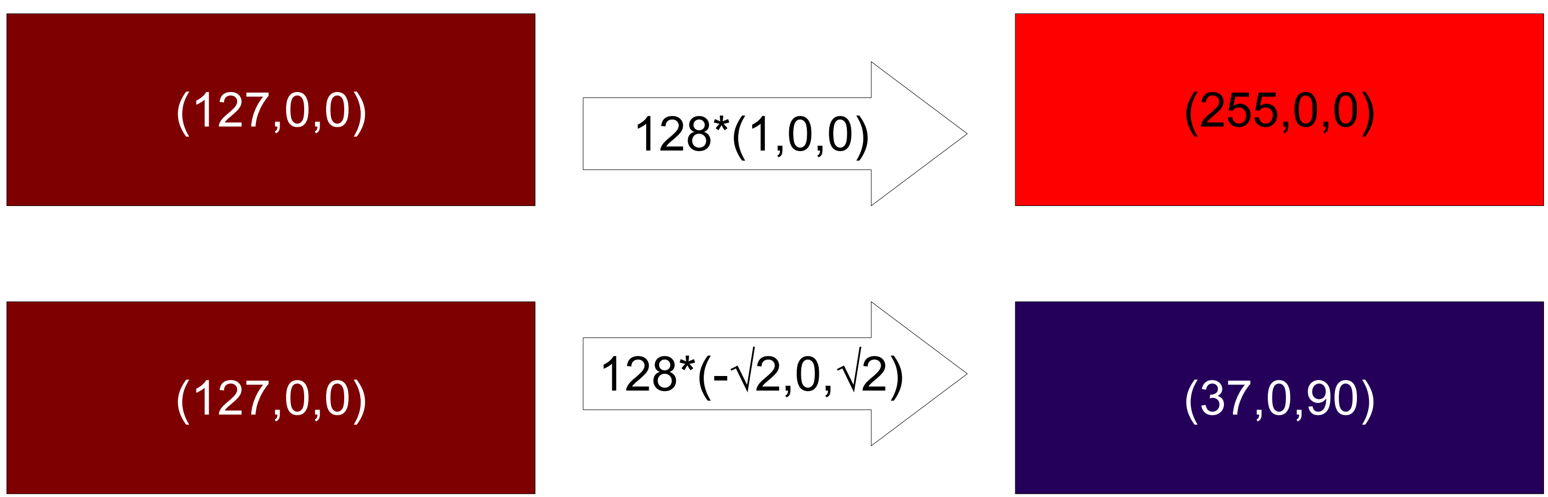


Image Data Similarity Search

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

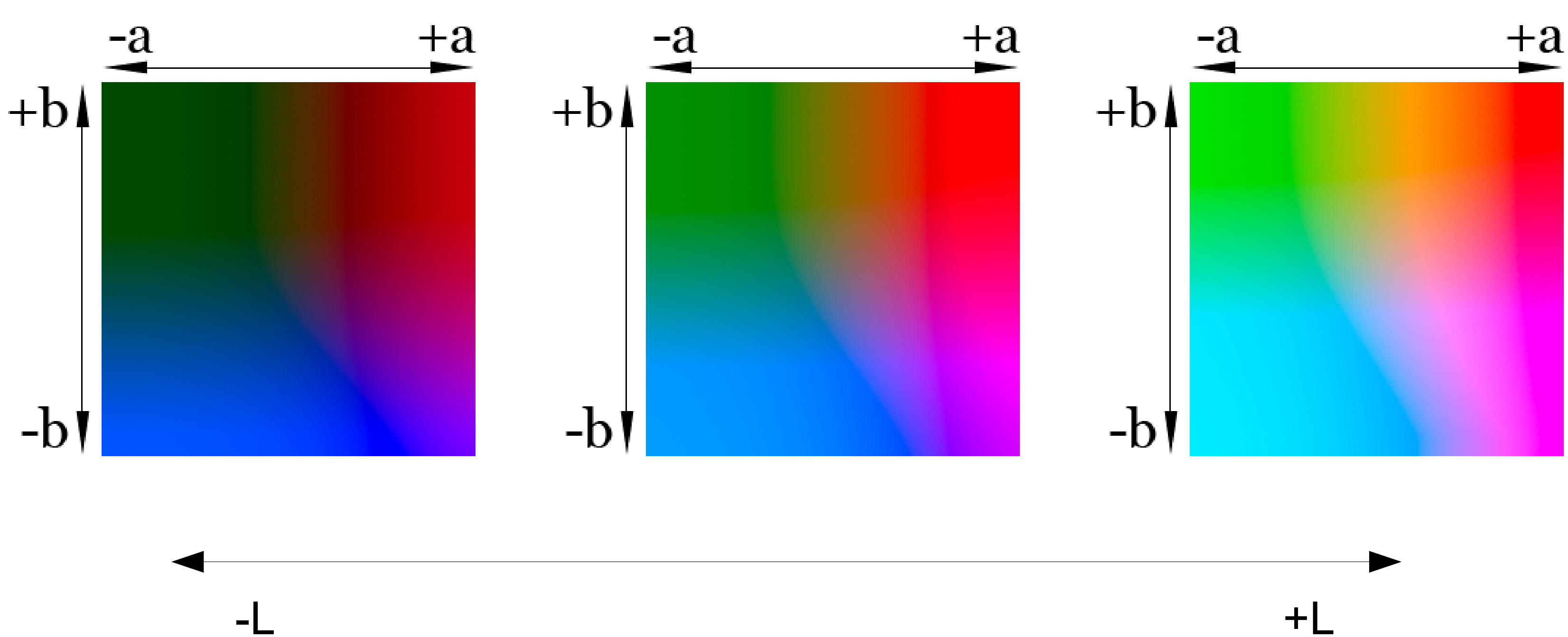


Image Data Similarity Search

Color:

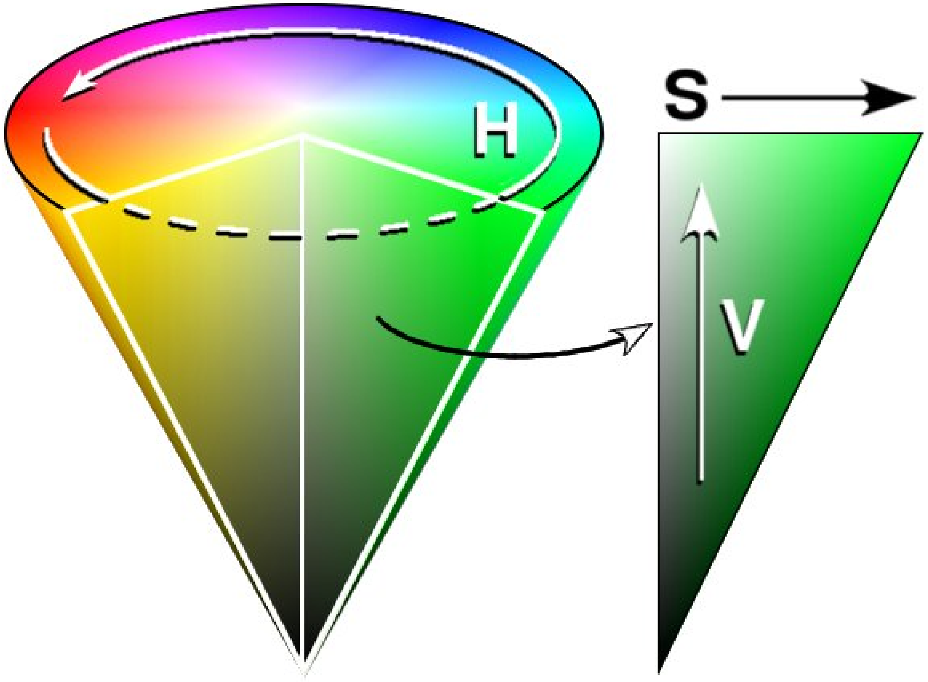
CIE L*a*b: Luminance, Green-Red, Blue-Yellow

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

Image Data Similarity Search

Color:

HSV: Hue, Saturation, Value



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)

Image Data Similarity Search

Color Histograms:

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

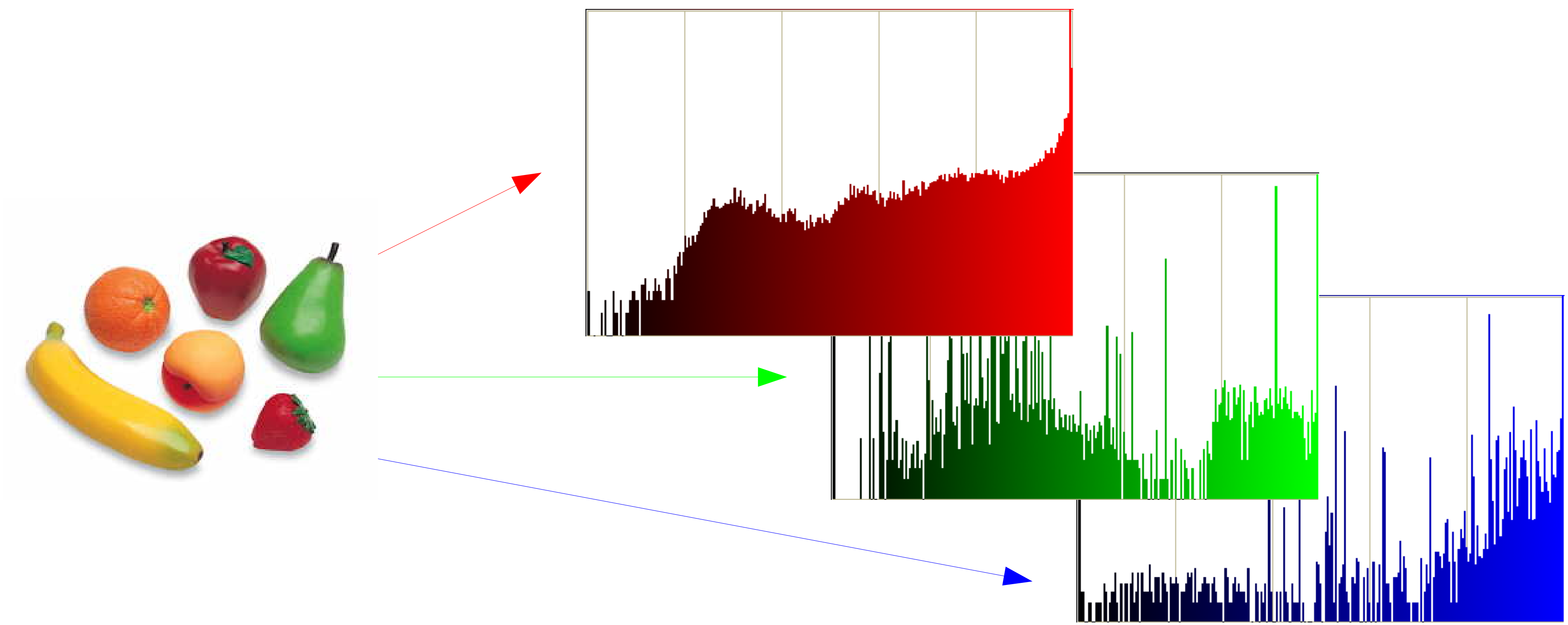


Image Data Similarity Search

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 a larger L_2 distance than is perceptually meaningful.

Image Data Similarity Search

Color Moments:

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

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

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

Image Data Similarity Search

Color Moments:

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

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

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

$$\mu_1(0) = \mu_1{}' := mean$$

$$\mu_2(\mu_1{}') := variance$$

$$\mu_3(\mu_1{}') := skew$$

Image Data Similarity Search

Color Moments:

Compact representation of histograms (3 numbers per color channel)

More robust against quantization error

Simple dissimilarity metric:

$$D(h_a, h_b) = w_1 |\mu_1{'}_a - \mu_1{'}_b| + w_2 |\mu_{2a} - \mu_{2b}| + w_3 |\mu_{3a} - \mu_{3b}|$$

Image Data Similarity Search

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:

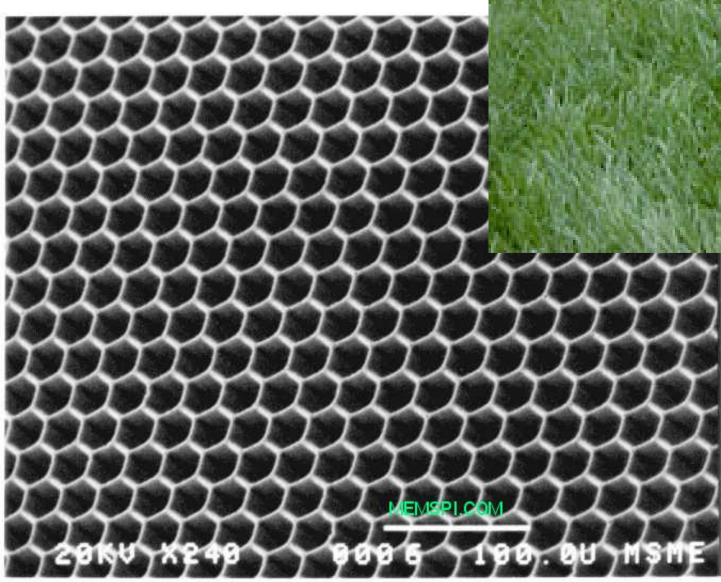


Image Data Similarity Search

Texture:

Psychologically meaningful parameters:

Coarseness
Contrast
Directionality
Line-like
Regularity
Roughness

Image Data Similarity Search

Texture:

Texture can also be analyzed with wavelets

Similar textures possess similarities in the wavelet subbands

Image Data Similarity Search

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 ?

Image Data Similarity Search

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 $\theta(l)$

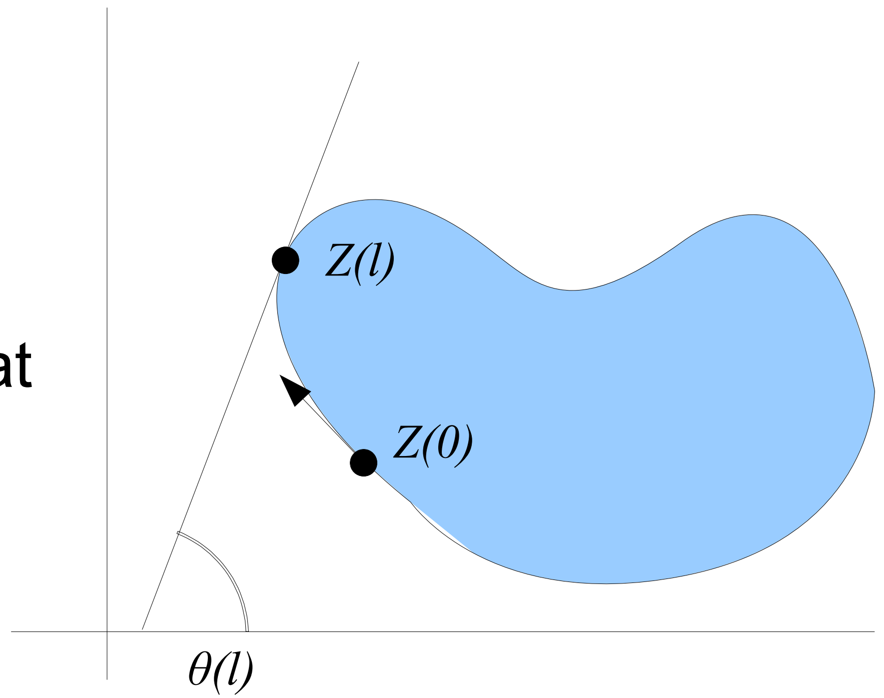


Image Data Similarity Search

Fourier Descriptors:

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

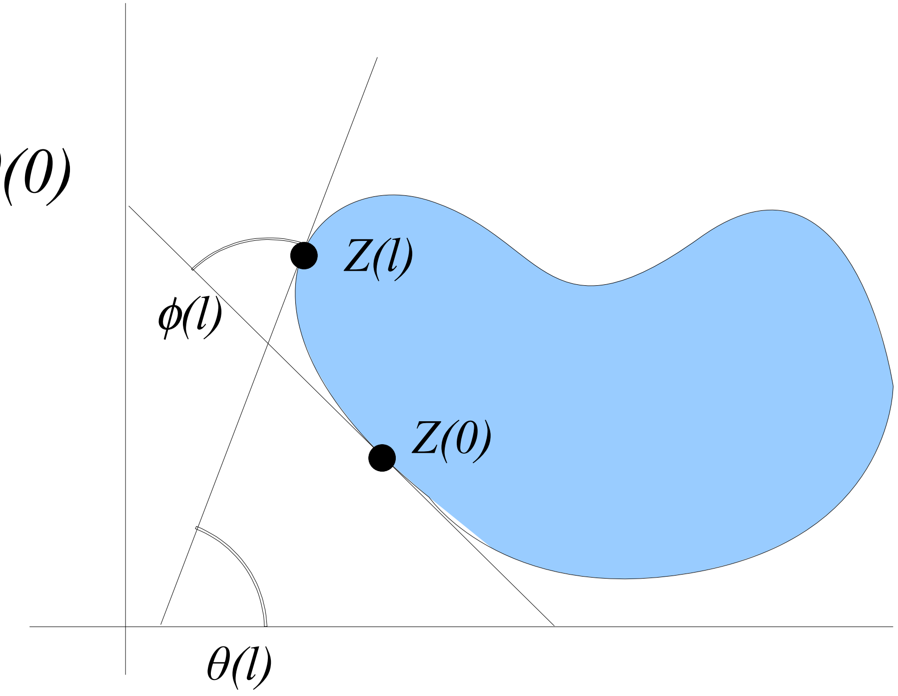


Image Data Similarity Search

Fourier Descriptors:

$$\phi^*(t) = \phi\left(\frac{Lt}{2\pi}\right) + t,\ 0 \leq t \leq 2\pi$$

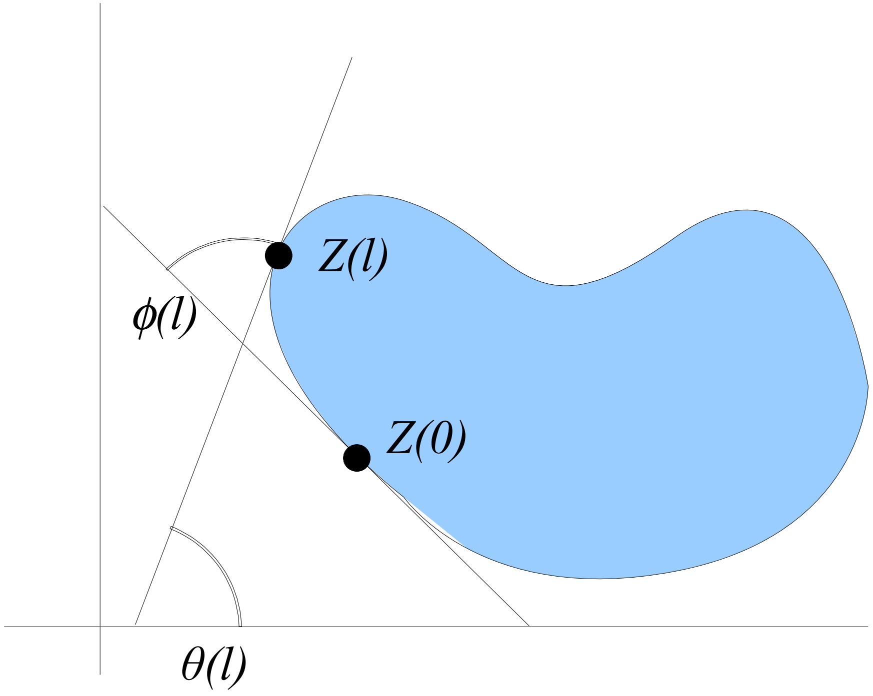
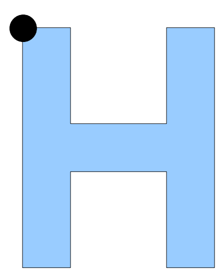
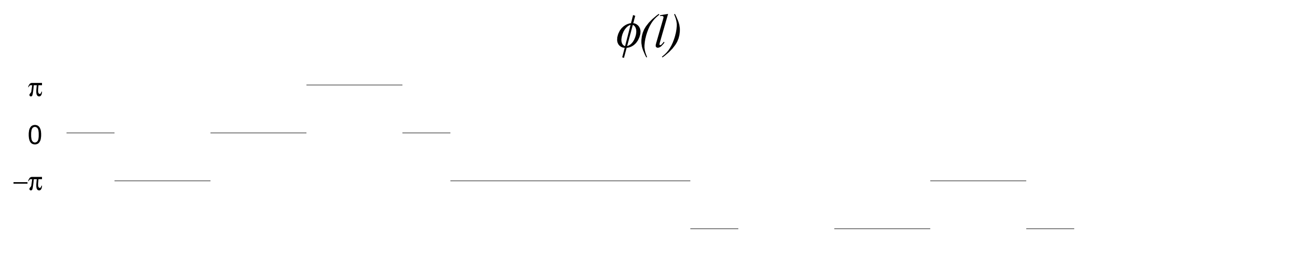
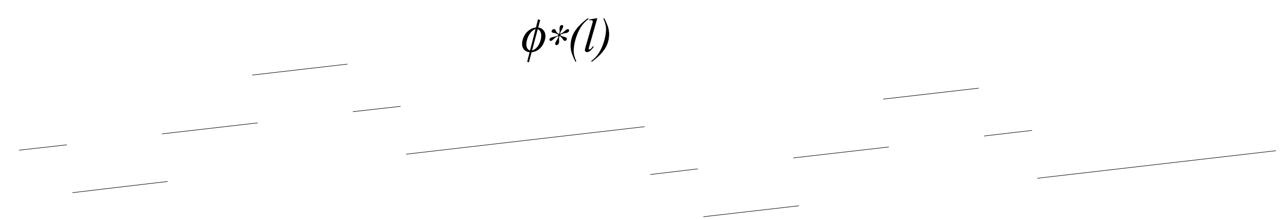


Image Data Similarity Search

Fourier Descriptors:







Fourier Descriptors:

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

Image Data Similarity Search

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:

Image Data Similarity Search

Implementation:

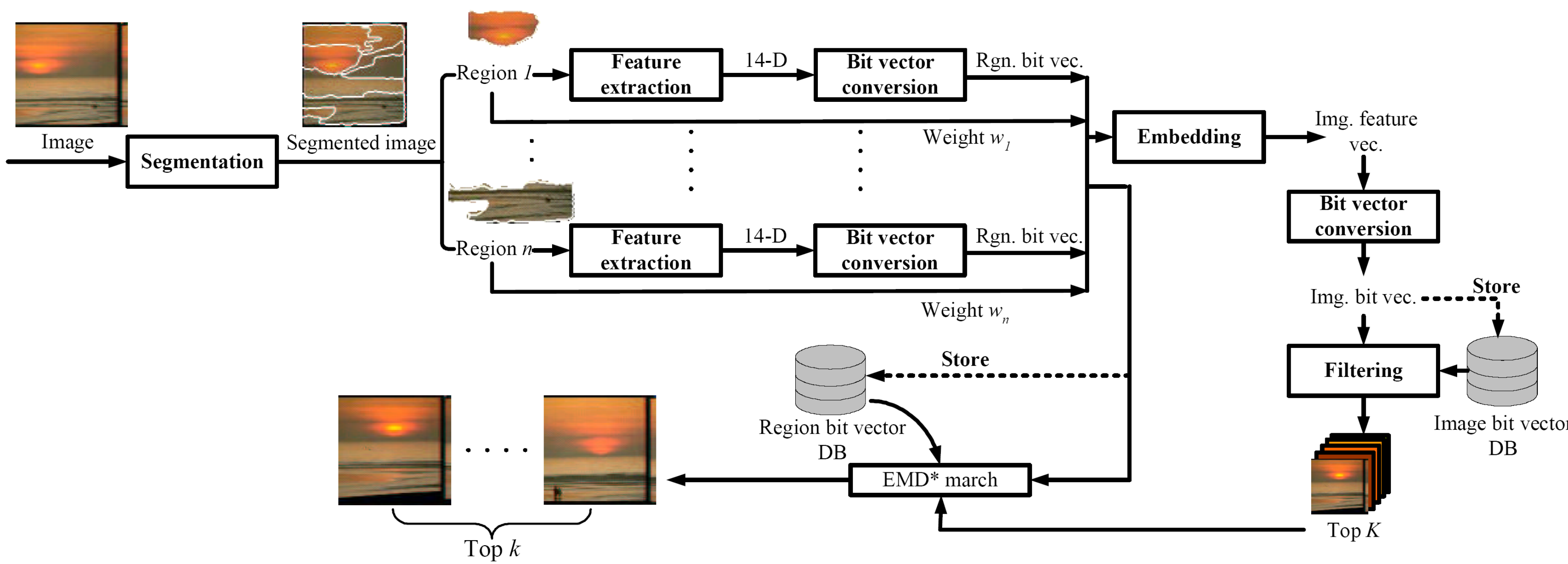


Image Data Similarity Search

Segmentation:

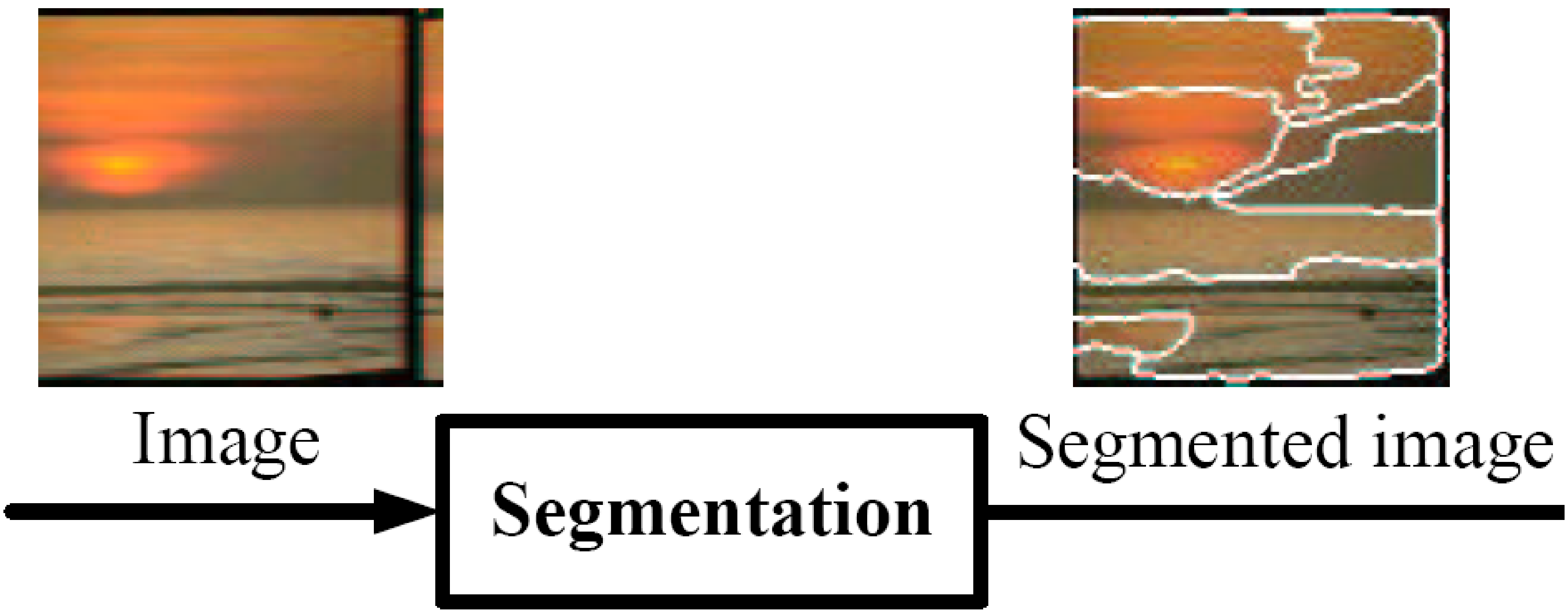


Image Data Similarity Search

Feature Extraction:

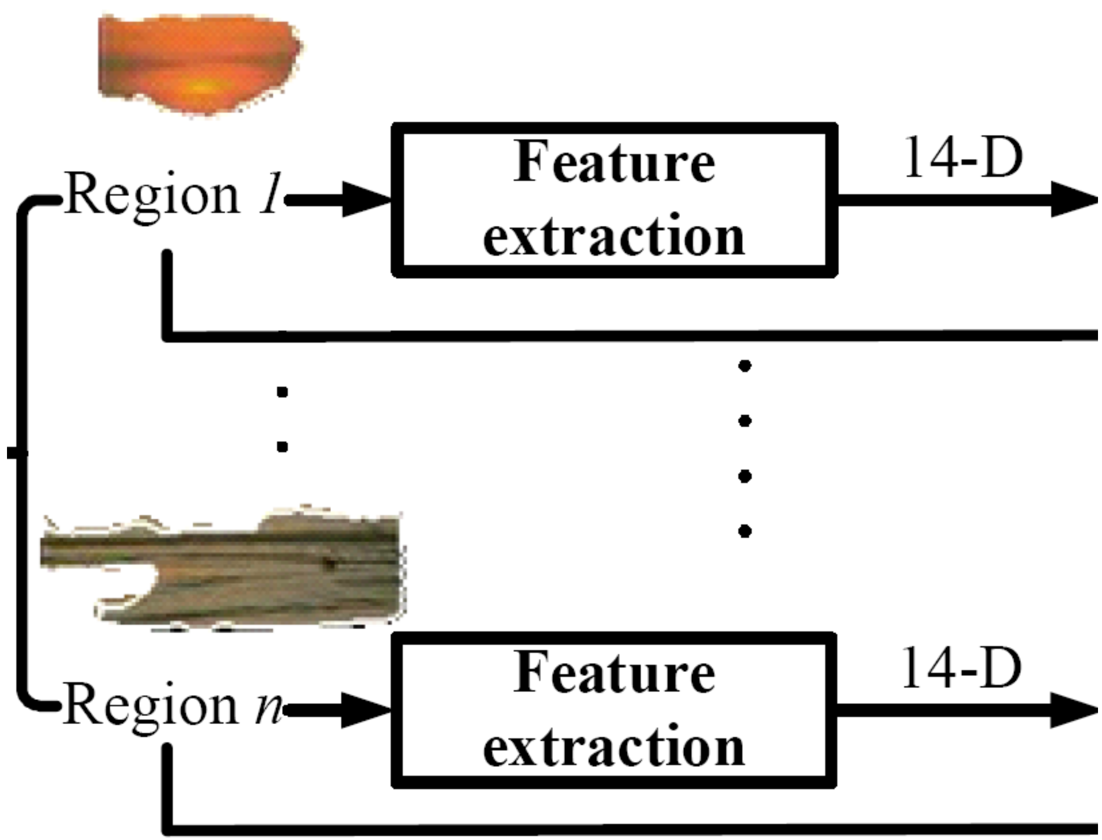


Image Data Similarity Search

Feature Extraction:

For each region in the image, calculate:

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

Image Data Similarity Search

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)

Image Data Similarity Search

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

Image Data Similarity Search

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

Image Data Similarity Search

Bit Vector Conversion:

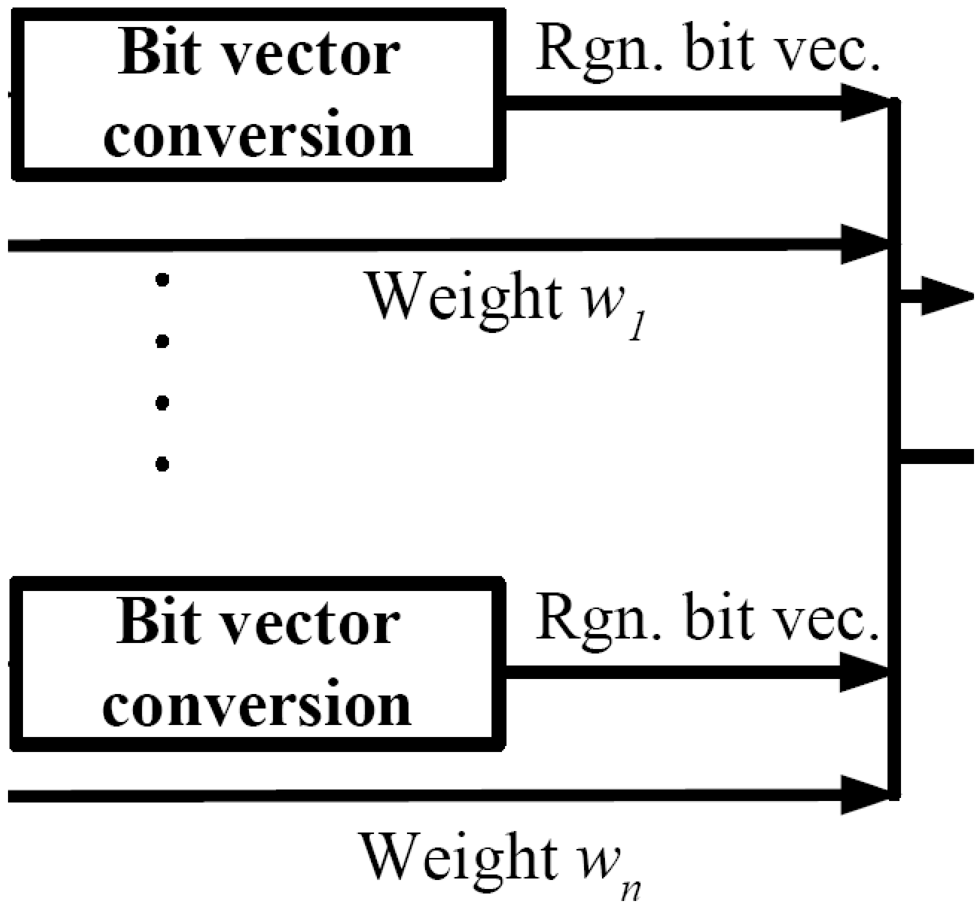


Image Data Similarity Search

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 : $[0, d\text{-}1]$ with probability p_i
Pick t : $[l_i, h_i]$

Image Data Similarity Search

Bit Vector Conversion:

Pick i : [0, $d\text{-}1$] with probability p_i

Pick t : [l_i, h_i]

$$bit = \begin{Bmatrix} v_i < t \,, 0 \\ v_i \geq t \,, 1 \end{Bmatrix}$$

Image Data Similarity Search

Bit Vector Conversion:

Lemma 1: $\left\| u-v \right\|_{L_{1w}} = x \Rightarrow Pr\left(bit(u) \neq bit(v)\right) = x/T$

Proof:

$$Pr\left(bit(u) \neq bit(v) | C_i\right) = \left\| u_i - v_i \right\|_{L_1} / r_i$$

$$Pr\left(bit(u) \neq bit(v)\right) = \sum_{i=0}^{d-1} Pr\left(bit(u) \neq bit(v) | C_i\right) \times Pr\left(C_i\right)$$

$$Pr\left(bit(u) \neq bit(v)\right) = \sum_{i=0}^{d-1} \frac{\left\| u_i - v_i \right\|_{L_1}}{r_i} \times \frac{w_i \times r_i}{T}$$

$$Pr\left(bit(u) \neq bit(v)\right) = \sum_{i=0}^{d-1} w_i \times \left\| u_i - v_i \right\|_{L_1} / T$$

$$Pr\left(bit(u) \neq bit(v)\right) = x/T$$

Bit Vector Conversion:

XOR groups of K bits to produce a single bit

101010 $\rightarrow$ 1
100100 $\rightarrow$ 0
111010 $\rightarrow$ 0
100101 $\rightarrow$ 1

Bit Vector Conversion:

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

Proof:

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

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

$$q=\frac{1}{2}\left(1-\left(1-2p\right)^K\right)=0.5\left(1-\left(1-2x/T\right)^K\right)$$

Image Data Similarity Search

Embedding:

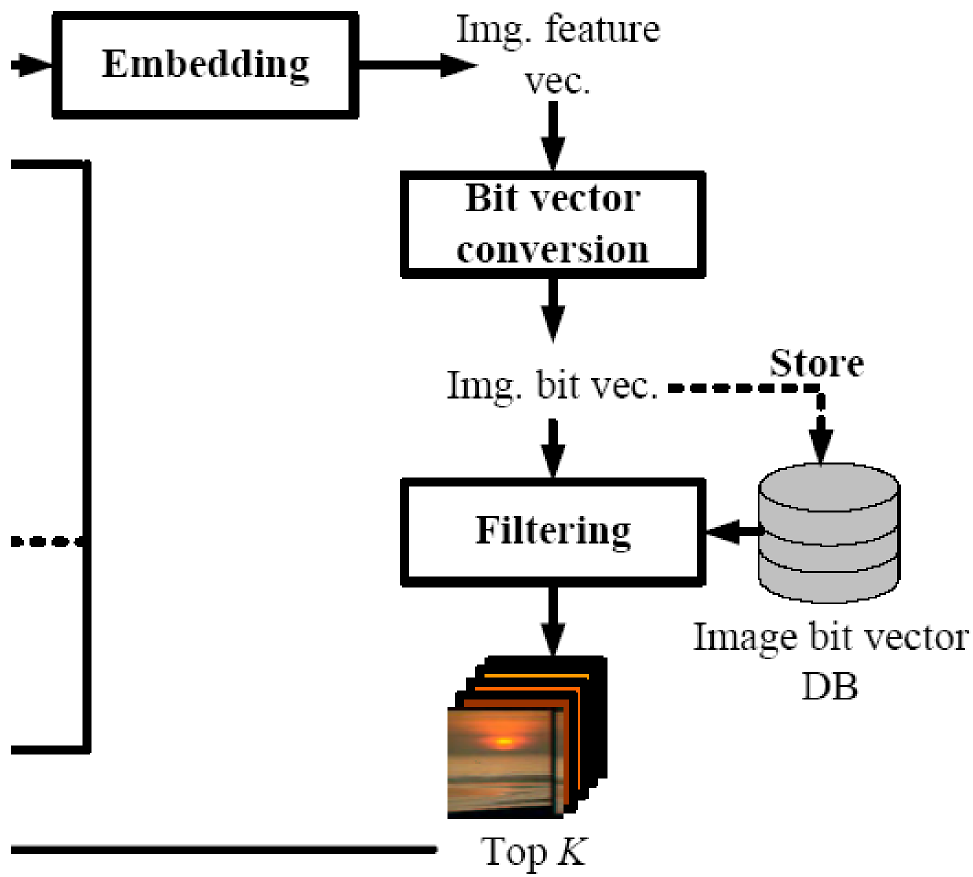


Image Data Similarity Search

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

Image Data Similarity Search

Embedding:

An image is a set of regions (bit vectors)

	1	2	3	4	5	6	7	8	w_i
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

Image Data Similarity Search

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

$$r_{p_j} = b_j \quad for\ j = 1, 2, ..., h$$

with r_{p_j} signifying the p_j^{th} bit in the bit vector representing r

Image Data Similarity Search

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))

	1	2	3	4	5	6	7	8	w_i
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***
$MW(p)$									***0.4***

Image Data Similarity Search

Embedding:

Repeat this process with several distinct patterns $\{P_1, P_2, ..., P_m\}$. 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.