Introduction to Image Description

Alexandre Xavier Falcão

Institute of Computing - UNICAMP

afalcao@ic.unicamp.br

Introduction to Image Description

- An image dataset $\mathcal{Z}$ may be represented by (sub)images, superpixels, or objects, called samples.

Introduction to Image Description

- An image dataset $\mathcal{Z}$ may be represented by (sub)images, superpixels, or objects, called samples.

- A descriptor is an algorithm that extracts a feature vector $\mathbf{x}(s) = (x_1(s), x_2(s), \ldots, x_n(s))$ from any sample $s \in \mathcal{Z}$.

Introduction to Image Description

- An image dataset $\mathcal{Z}$ may be represented by (sub)images, superpixels, or objects, called samples.

- A descriptor is an algorithm that extracts a feature vector $\mathbf{x}(s) = (x_1(s), x_2(s), \ldots, x_n(s))$ from any sample $s \in \mathcal{Z}$.

- The descriptor may also include a distance function (e.g., $d(s,t) = \|\mathbf{x}(t) - \mathbf{x}(s)\|$) to compare the dissimilarity between samples s and t in the feature space.

Introduction to Image Description

- An image dataset $\mathcal{Z}$ may be represented by (sub)images, superpixels, or objects, called samples.
- A descriptor is an algorithm that extracts a feature vector $\mathbf{x}(s) = (x_1(s), x_2(s), \ldots, x_n(s))$ from any sample $s \in \mathcal{Z}$.
- The descriptor may also include a distance function (e.g., $d(s,t) = \|\mathbf{x}(t) - \mathbf{x}(s)\|$) to compare the dissimilarity between samples s and t in the feature space.

How can we extract and combine simple image descriptors?

Agenda

This lecture covers examples of simple image descriptors and their combination by genetic programming [1].

Agenda

This lecture covers examples of simple image descriptors and their combination by genetic programming [1].

- Color: color histogram and BIC [2].

Agenda

This lecture covers examples of simple image descriptors and their combination by genetic programming [1].

- Color: color histogram and BIC [2].
- Texture: LBP [3].

Agenda

This lecture covers examples of simple image descriptors and their combination by genetic programming [1].

- Color: color histogram and BIC [2].
- Texture: LBP [3].
- Shape: multiscale fractal dimension [4] and shape saliences [5].

Agenda

This lecture covers examples of simple image descriptors and their combination by genetic programming [1].

- Color: color histogram and BIC [2].
- Texture: LBP [3].
- Shape: multiscale fractal dimension [4] and shape saliences [5].

The subsequent lectures will focus on popular texture descriptors: HoG, BoVW, and CNN.

Color histogram

The histogram $h(i)$ of a grayscale image $\hat{I} = (D_I, I)$ is defined as

$$\begin{aligned} h(i) &= \sum_{\forall p \in D_I} \delta\left(I(p) - i\right), \\ \delta\left(I(p) - i\right) &= \begin{cases} 1, & \text{when } i = I(p), \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

Color histogram

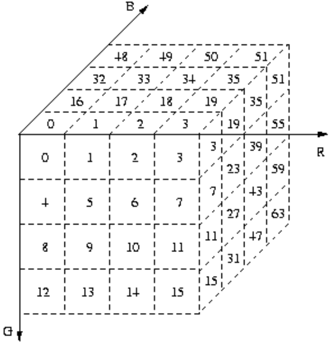
The histogram $h(i)$ of a grayscale image $\hat{I} = (D_I, I)$ is defined as

$$\begin{aligned} h(i) &= \sum_{\forall p \in D_I} \delta\left(I(p) - i\right), \\ \delta\left(I(p) - i\right) &= \begin{cases} 1, & \text{when } i = I(p), \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

For color images $\hat{I} = (D_I, \mathbf{I})$, where $\mathbf{I}(p) = (I_1(p), I_2(p), I_3(p))$ in some color space (e.g., RGB, YCbCr, Lab), this definition would lead to a sparse and, in this case, likely ineffective feature vector $\mathbf{x}(s) = h$, where $s = \hat{I}$ and $x_i(s) = h(i)$.

Color histogram

The problem is addressed by dividing each axis of the color space into fixed intervals, called bins. For a 64-bin histogram of a 24-bit-RGB image, $I_1 = R$, $I_2 = G$, and $I_3 = B$, the histogram h is

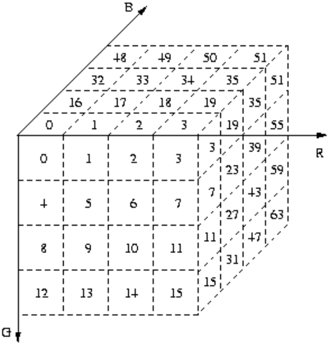


$$
\begin{aligned}
h(i) &= \sum_{\forall p \in D_I} \delta\left(V(p) - i\right), \\
V(p) &= \frac{R(p)}{64} + 4\left[\frac{G(p)}{64}\right] + 16\left[\frac{B(p)}{64}\right],
\end{aligned}
$$

where $V(p) \in [0, 63]$.

Color histogram

The problem is addressed by dividing each axis of the color space into fixed intervals, called bins. For a 64-bin histogram of a 24-bit-RGB image, $I_1 = R$, $I_2 = G$, and $I_3 = B$, the histogram h is



$$
\begin{aligned}
h(i) &= \sum_{\forall p \in D_I} \delta\left(V(p) - i\right), \\
V(p) &= \frac{R(p)}{64} + 4\left[\frac{G(p)}{64}\right] + 16\left[\frac{B(p)}{64}\right],
\end{aligned}
$$

where $V(p) \in [0, 63]$. It is also common to create normalized histograms by setting $h(i) \leftarrow \frac{h(i)}{|D_I|}$, $i = 1, 2, \ldots, 64$.

BIC — Border and Interior Classification

The BIC descriptor consists of three components.

- A simple and yet effective pixel classification algorithm into image regions defined as either border (high frequency) or interior (low frequency).
- A compact region representation based on color histograms.
- A logarithmic distance function to compare histograms from two images.

BIC — Border and Interior Classification

- Let $L(p) \in \{0, 1\}$ indicate when p is an interior or border pixel.

BIC — Border and Interior Classification

- Let $L(p) \in \{0,1\}$ indicate when p is an interior or border pixel.

- Once $V(p)$ is computed for each pixel $p \in D_I$, a pixel p is classified as follows.

$$L(p) = \begin{cases} 1, & \text{when } \exists q \in \mathcal{A}_1(p) \mid V(q) \neq V(p), \text{ and} \\ 0, & \text{otherwise.} \end{cases}$$

BIC — Border and Interior Classification

- Let $L(p) \in \{0, 1\}$ indicate when p is an interior or border pixel.

- Once $V(p)$ is computed for each pixel $p \in D_I$, a pixel p is classified as follows.

$$L(p) = \begin{cases} 1, & \text{when } \exists q \in \mathcal{A}_1(p) \mid V(q) \neq V(p), \text{ and} \\ 0, & \text{otherwise.} \end{cases}$$

- The normalized color histograms h_0 (interior) and h_1 (border) of each region are computed and then quantized from 0 to 255 by setting $h_0(i) \leftarrow 255 h_0(i)$ and $h_1(i) \leftarrow 255 h_1(i)$.

BIC — Border and Interior Classification

- Assuming the $L1$ metric, the logarithmic distance (log2) is encoded in the histograms by mapping $h_b(i)$, $b = \{0, 1\}$ from $[0, 255]$ to $[0, 9]$ (4 bits per bin) as follows.

$$h_b(i) \leftarrow \begin{cases} 0 & \text{if } h_b(i) = 0, \\ 1 & \text{if } h_b(i) < 1, \\ 2 & \text{if } h_b(i) < 2, \\ 3 & \text{if } h_b(i) < 4, \\ 4 & \text{if } h_b(i) < 8, \\ 5 & \text{if } h_b(i) < 16, \\ 6 & \text{if } h_b(i) < 32, \\ 7 & \text{if } h_b(i) < 64, \\ 8 & \text{if } h_b(i) < 128, \\ 9 & \text{otherwise.} \end{cases}$$

BIC — Border and Interior Classification

- Assuming the $L1$ metric, the logarithmic distance (log2) is encoded in the histograms by mapping $h_b(i)$, $b = \{0, 1\}$ from $[0, 255]$ to $[0, 9]$ (4 bits per bin) as follows.

$$h_b(i) \leftarrow \begin{cases} 0 & \text{if } h_b(i) = 0, \\ 1 & \text{if } h_b(i) < 1, \\ 2 & \text{if } h_b(i) < 2, \\ 3 & \text{if } h_b(i) < 4, \\ 4 & \text{if } h_b(i) < 8, \\ 5 & \text{if } h_b(i) < 16, \\ 6 & \text{if } h_b(i) < 32, \\ 7 & \text{if } h_b(i) < 64, \\ 8 & \text{if } h_b(i) < 128, \\ 9 & \text{otherwise.} \end{cases}$$

- Finally, the histograms h_0 and h_1 are concatenated into a single histogram $\mathbf{x}$.

Texture: Local Binary Patterns (LBP)

For a given grayscale image $\hat{I} = (D_I, I)$ and adjacency set $\mathcal{A}_{\sqrt{2}}(p) = \{q_1, q_2, \ldots, q_8\}$ for $p \in D_I$.

- A local binary pattern $B(p) \in [0, 255]$ is assigned to p by setting each bit $b_k(z)$, $k = 1, 2, \ldots, 8$, of $z = B(p)$ as

$$b_k(z) \quad \leftarrow \quad \begin{cases} 1 & \text{if } I(p) > I(q_k), \\ 0 & \text{otherwise.} \end{cases}$$

Texture: Local Binary Patterns (LBP)

For a given grayscale image $\hat{I} = (D_I, I)$ and adjacency set $\mathcal{A}_{\sqrt{2}}(p) = \{q_1, q_2, \ldots, q_8\}$ for $p \in D_I$.

- A local binary pattern $B(p) \in [0, 255]$ is assigned to p by setting each bit $b_k(z)$, $k = 1, 2, \ldots, 8$, of $z = B(p)$ as

$$b_k(z) \quad \leftarrow \quad \begin{cases} 1 & \text{if } I(p) > I(q_k), \\ 0 & \text{otherwise.} \end{cases}$$

- The histogram of the map B can be used as LBP feature vector.

Texture: Local Binary Patterns (LBP)

For a given grayscale image $\hat{I} = (D_I, I)$ and adjacency set $\mathcal{A}_{\sqrt{2}}(p) = \{q_1, q_2, \ldots, q_8\}$ for $p \in D_I$.

- A local binary pattern $B(p) \in [0, 255]$ is assigned to p by setting each bit $b_k(z)$, $k = 1, 2, \ldots, 8$, of $z = B(p)$ as

$$b_k(z) \leftarrow \begin{cases} 1 & \text{if } I(p) > I(q_k), \\ 0 & \text{otherwise.} \end{cases}$$

- The histogram of the map B can be used as LBP feature vector.
- Alternatively, D_I can be divided into cells of $N \times N$ pixels, one LBP histogram can be extracted per cells, and the histograms concatenated into a single feature vector per image.

Texture: Local Binary Patterns (LBP)

- However, the LBP histograms are not rotation invariant, which has inspired many variants.

Texture: Local Binary Patterns (LBP)

- However, the LBP histograms are not rotation invariant, which has inspired many variants.

- Some of them treat rotation in the distance function and others incorporate rotation invariance in the feature vector.

Texture: Local Binary Patterns (LBP)

- However, the LBP histograms are not rotation invariant, which has inspired many variants.
- Some of them treat rotation in the distance function and others incorporate rotation invariance in the feature vector.
- The problem is also not critical, when the images are aligned.

Texture: Local Binary Patterns (LBP)

- However, the LBP histograms are not rotation invariant, which has inspired many variants.
- Some of them treat rotation in the distance function and others incorporate rotation invariance in the feature vector.
- The problem is also not critical, when the images are aligned.
- The extension to color images can simply concatenate the histograms from each band.

Shape: Multiscale fractal dimension

The fractal dimension of a 2D point set $\mathcal{S}$ (contour, skeleton) by Minkowski-Bouligand is a number $F \in [0, 2]$,

$$F \;=\; 2 - \lim_{r \to 0} \frac{\ln(A(r))}{\ln(r)},$$

where $A(r)$ is the number of propagated points (area) when $\mathcal{S}$ is dilated by a disk of radius r.

Shape: Multiscale fractal dimension

The fractal dimension of a 2D point set $\mathcal{S}$ (contour, skeleton) by Minkowski-Bouligand is a number $F \in [0, 2]$,

$$F = 2 - \lim_{r \to 0} \frac{\ln(A(r))}{\ln(r)},$$

where $A(r)$ is the number of propagated points (area) when $\mathcal{S}$ is dilated by a disk of radius r.

- The fractal dimension represents the self-similarity of S when r tends to zero.

Shape: Multiscale fractal dimension

The fractal dimension of a 2D point set $\mathcal{S}$ (contour, skeleton) by Minkowski-Bouligand is a number $F \in [0, 2]$,

$$F \quad = \quad 2 - \lim_{r \to 0} \frac{\ln(A(r))}{\ln(r)},$$

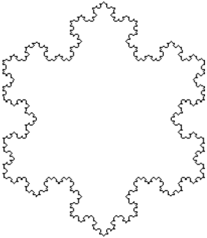
where $A(r)$ is the number of propagated points (area) when $\mathcal{S}$ is dilated by a disk of radius r.

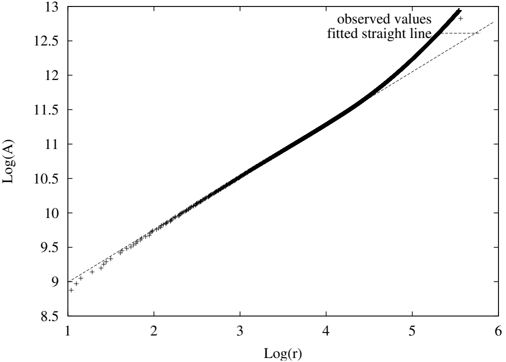
- The fractal dimension represents the self-similarity of S when r tends to zero.

- Note that A can be obtained from the cumulative histogram of the Euclidean distance map of $\mathcal{S}$ upto the distance r.

Shape: Multiscale fractal dimension

It is known, for instance, that the fractal dimension of a koch star (on the left) is $F \approx 1.26$ [4].

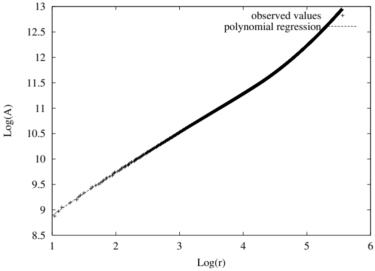


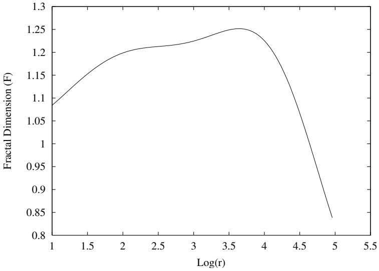


One can then fit a line to the logarithmic curve of the cumulative histogram of the EDT (on the right) and use the slope m of the line to estimate $F = 2 - m \approx 1.23$.

Shape: Multiscale fractal dimension

By fitting a polynomial curve (on the left) and computing its first derivative, the resulting curve F (on the right) is a feature vector of the shape, called multiscale fractal dimension.

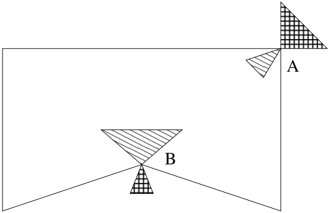




Note that its maximum value (on the right) is ≈ 1.25 for some value $\ln(r) \in [3.5, 4]$.

Shape: Point/segment saliences

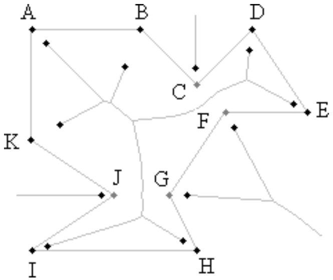
The dilation of a contour $\mathcal{S}$ by a disk of small radius r (e.g., $r = 10$) shows that the outside area $A_{out}(p)$ of the influence zone of a point $p \in \mathcal{S}$ is higher than its inside area $A_{in}(p)$, when p is convex, and the other way around when it is concave [5].

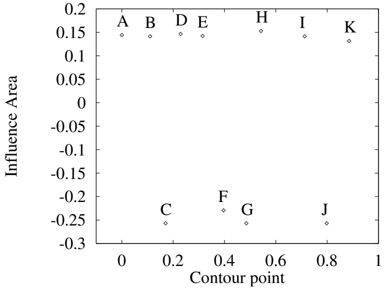


Such areas come from the root histogram of the EDT, which can be normalized for the purpose of shape description.

Shape: Point/segment saliences

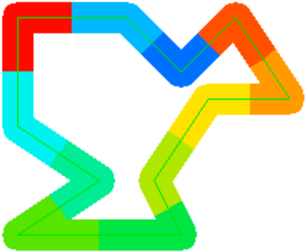
By considering only the salience points $p \in \mathcal{S}$, a point-salience feature vector can encode the respective positive (convex) and negative (concave) areas.





Shape: Point/segment saliences

Similar idea works for a segment-salience feature vector, when replacing points by contour segments around the salience points.



Note that, in any case, the distance function must account for possible shape rotations.

Combining descriptors by genetic programming

- Let $\{D_1, D_2, \ldots, D_k\}$ be a set of descriptors such that $D_i = (v_i, d_i)$, $i = 1, 2, \ldots, k$, consists of
  - an algorithm v_i that extracts feature vectors $v_i(s)$ and $v_i(t)$ from samples s and t, and
  - a distance function d_i that assigns a dissimilarity value $d_i(s,t)$ in the feature space between samples s and t (e.g., $\|v_i(t) - v_i(s)\|$).

Combining descriptors by genetic programming

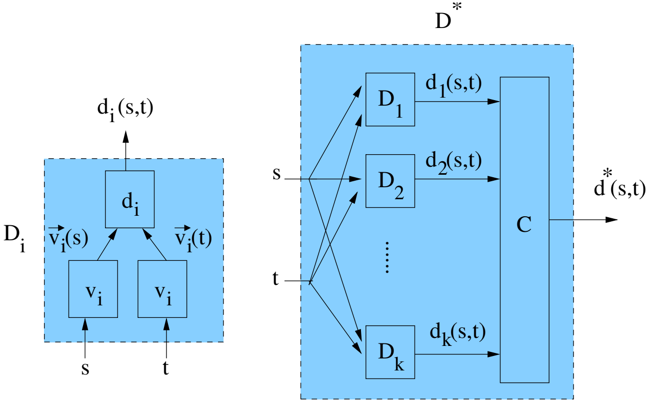
- Let $\{D_1, D_2, \ldots, D_k\}$ be a set of descriptors such that $D_i = (v_i, d_i)$, $i = 1, 2, \ldots, k$, consists of
  - an algorithm v_i that extracts feature vectors $v_i(s)$ and $v_i(t)$ from samples s and t, and
  - a distance function d_i that assigns a dissimilarity value $d_i(s,t)$ in the feature space between samples s and t (e.g., $\|v_i(t) - v_i(s)\|$).
- The distance functions d_i, $i = 1, 2, \ldots, k$, can be combined into a single distance function d using, for instance, Genetic Programming (GP) [1].

Combining descriptors by genetic programming

- Let $\{D_1, D_2, \ldots, D_k\}$ be a set of descriptors such that $D_i = (v_i, d_i)$, $i = 1, 2, \ldots, k$, consists of
  - an algorithm v_i that extracts feature vectors $v_i(s)$ and $v_i(t)$ from samples s and t, and
  - a distance function d_i that assigns a dissimilarity value $d_i(s,t)$ in the feature space between samples s and t (e.g., $\|v_i(t) - v_i(s)\|$).
- The distance functions d_i, $i = 1, 2, \ldots, k$, can be combined into a single distance function d using, for instance, Genetic Programming (GP) [1].
- The resulting descriptor D is called a composite descriptor.

Combining descriptors by genetic programming

Illustration of a composite descriptor D^* with the GP combiner C (one may use any other optimization technique).



(a) (b)

Combining descriptors by genetic programming

- GP is an Artificial Intelligence technique based on biological principles of heritage and evolution.

Combining descriptors by genetic programming

- GP is an Artificial Intelligence technique based on biological principles of heritage and evolution.

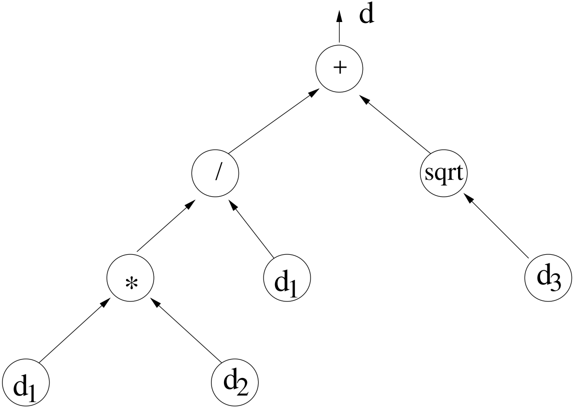
- Each candidate solution is an individual of a population, as represented by a data structure (e.g., tree, list, stack) whose nodes are mathematical operations, rather than a sequence of numbers, such as in genetic algorithms.

Combining descriptors by genetic programming

- GP is an Artificial Intelligence technique based on biological principles of heritage and evolution.
- Each candidate solution is an individual of a population, as represented by a data structure (e.g., tree, list, stack) whose nodes are mathematical operations, rather than a sequence of numbers, such as in genetic algorithms.
- From some initial random population, the most promising individuals pass through genetic transformations (e.g., mutations) that make the population more diverse and suitable to solve the problem.

Combining descriptors by genetic programming

Example of individual, where the distances d_i, $i = 1, 2, \ldots, k$, are the terminal nodes of a binary tree and the remaining nodes are other mathematical operations.



Combining descriptors by genetic programming

In addition to the mathematical operations (see examples in [6]),

- the reproduction selects the most effective individuals to the next population,
- the crossover exchanges subtrees between selected individuals to increase diversity, generating new trees (sons), and
- the mutation replaces a subtree of a selected individual by another subtree randomly chosen.

The individuals are assessed by a fitness function, which can be the accuracy of classification in a validation set.

Combining descriptors by genetic programming

The algorithm can be sketched as follows.

```
1. Generate an initial random population (first generation).
2. For each generation from a given maximum number do.
3.     Evaluate each individual by the fitness function.
4.     Select a number of the most effective ones.
5.     Generate the next population by reproduction, crossover,
       and mutation of the selected individuals.
6. Select the best individual as the final solution.
```

[1] R. da S. Torres, A.X. Falcão, M.A. Gonçalves, J.P. Papa, B. Zhang, W. Fan, and E.A. Fox.

A genetic programming framework for content-based image retrieval.

Pattern Recognition, 42(2):283 – 292, 2009.

Learning Semantics from Multimedia Content.

[2] Renato O. Stehling, Mario A. Nascimento, and Alexandre X. Falcão.

A compact and efficient image retrieval approach based on border/interior pixel classification.

In *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, pages 102–109, 2002.

[3] Rafael C. Gonzalez and Richard E. Woods.

Digital Image Processing (4th Edition).

Pearson, 2018.

`www.imageprocessingplace.com`.

[4] R. da S. Torres, A.X. Falcão, and L. da F. Costa.

A graph-based approach for multiscale shape analysis.

Pattern Recognition, 37(6):1163 – 1174, 2004.

[5] R. da S. Torres and A.X. Falcão.

Contour salience descriptors for effective image retrieval and analysis.

Image and Vision Computing, 25(1):3 – 13, 2007.

[6] R. da S. Torres, A.X. Falcão, M.A. Gonçalves, B. Zhang, W. Fan, and E.A. Fox.

A new framework to combine descriptors for content-based image retrieval.

Technical Report IC-05-21, Institute of Computing, University of Campinas, September 2005.