Texture description through histograms of equivalent patterns: A unifying framework for LBP and related methods

Francesco Bianconi [1]

[1]Department of Engineering, Università degli Studi di Perugia, Italy

Queen Mary, University of London
August 11, 2015

Outline

About the speaker

- Contact details:
  - Bianconi, Francesco;
  - Department of Engineering, Università degli Studi di Perugia
    Via G. Duranti 93, 06125 Perugia (Italy);
  - e-mail: bianco@ieee.org
- Corrent position: *Professore Associato* (eqv. UK = Senior Lecturer) in the Department of Engineering at the University of Perugia, Italy;
- Background in Industrial Engineering (MEng Mechanical Engineering, PhD Computer-aided Design) – but gradually converted to Computer Science;
- Research interests in the area of image processing, with particular focus on:
  - Texture;
  - Colour;
  - Industrial applications;
  - Biomedical applications.

About the University of Perugia

- Founded in 1308;
- About 26,000 students;
- 16 departments:
  - Agricultural Sciences; Chemistry, Biology and Biotechnologies; Economics; Political Sciences; Philosophy, Human and Social Sciences; Physics and Geology; Law; Engineering; Civil and Environmental Engineering; Ancient and Modern Literature; Mathematics and Computer Science; Medicine; Experimental Medicine; Surgery and Biomedicine; Pharmacy; Veterinary;
- Ranked 1st for the second year in a row among the Italian universities with a number of students between 20,000 and 40,000 (Magista, 2015)

About the city


- A well-preserved medieval town;
- Population of about 160,000;
- Quiet and safe but culturally active;
- Good environment for students.

About the region

Umbria: 'the green heart of Italy'




What is texture?

- A property of visual appearance, just like *colour*, *shape* and *glossiness*;
- An intuitive concept: we can recognise texture when wee see it...
- ...but rather subjective and imprecise;
- No formal definition exists...
- ...but there are lots of methods to analyse it – 'a galaxy of texture features' (Xie and Mirmehdi, 2008);
- Related to the intertwinement, combination and interaction of elements into a complex whole (from latin `texere` = *to weave*).

Texture: basic facts

- It is a property of an area and not of a point;
- Strongly depends on scale (the same scene acquired from different distance will give rise to different textures);
- It is perceived as the combination of basic elements.

Textures and non-textures

Texture

Not texture

'Quasi texture'

Texture: tentative definitions

- 'What constitutes a macroscopic region. Its structure is simply attributed to the *repetitive patterns* in which *elements* or *primitives* are arranged according to a placement rule' (Tamura, Mori, and Yamawaki, 1978)
- 'A region in an image has a constant texture if a set of *local statistics* or other local properties of the picture function are *constant*, *slowly varying*, or *approximately periodic*' (Sklansky, 1978)
- 'Variation of data *at scale smaller* than the scale of interest' (Petrou and García Sevilla, 2006)
- 'If a pattern has both *randomness* and *regularity*, then this is probably what most people would call a texture' (Davies, 2008)

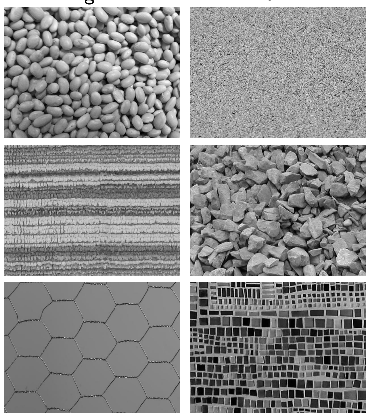
Texture: some intuitive attributes

High | Low

Coarseness

Directionality

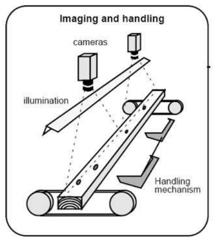
Stationariness

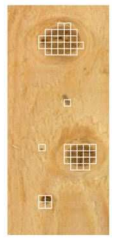


Applications of texture analysis

Industrial inspection

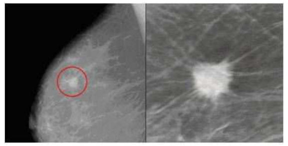
Quality control of industrial products: surface grading, surface inspection, defect detection, etc.





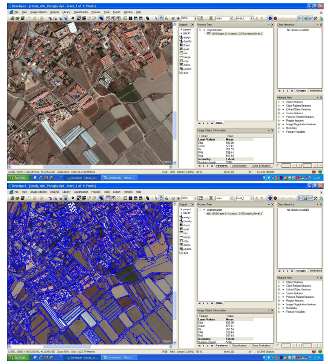
Applications of texture analysis
Computer-assisted diagnosis, staging and follow-up

Based on processing images from many sources: X-Rays, MRI, fMRI, CT, PET, histology, ultrasound, etc.



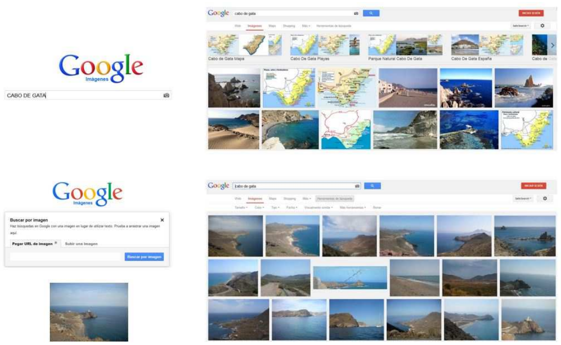
Applications of texture analysis
Remote sensing

Analysis of satellite images



Applications of texture analysis
Content-based image retrieval (CBIR)

Find images similar to a given one



Describing textures: the *bag-of-features* model

- Bag-of-features: a transposition of the *bag-of-words* model into images.
- Bag-of-words: an approach to natural language processing. Text is represented through the probability of occurrence of a set of words (dictionary) that appear in it.
- It is an *orderless* model: the relative position among the words doesn't matter.
  - Example: 'Italy rugby team keeps loosing to England' and 'England rugby team keeps loosing to Italy' are perfectly equivalent sentences in this model

The bag-of-words model: an example

He lay flat on the brown, pine-needled floor of the forest, his chin on his folded arms, and high overhead the wind blew in the tops of the pine trees. The mountainside sloped gently where he lay; but below it was steep and he could see the dark of the oiled road winding through the pass. There was a stream alongside the road and far down the pass he saw a mill beside the stream and the falling water of the dam, white in the summer sunlight.[1]

Dictionary: he, the, his and summer (dimension = four)

Total number of words = 87

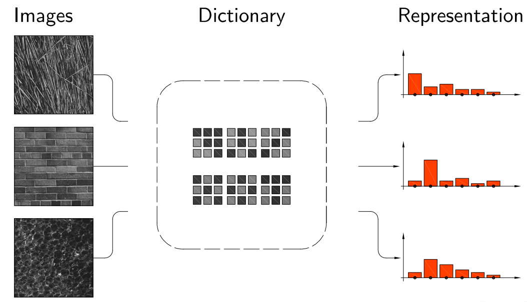
Occurrences: he = 4, the = 13, his = 2 and summer = 1

Bag-of-words representation: {4/87,13/87,2/87,1/87};

[1] From E. Hemingway, *For whom the bell tolls*

Bag-of-features: the general scheme

As the bag-of-words model represents texts through the probability of occurrence of certain *words* that appear in it, so the bag-of-features model represents images through the probability of occurrence of certain *visual words* (also referred to as *patterns*, *texels*, *textons*, etc.) that appear in the image.



Implementing a bag-of-features model: five design choices

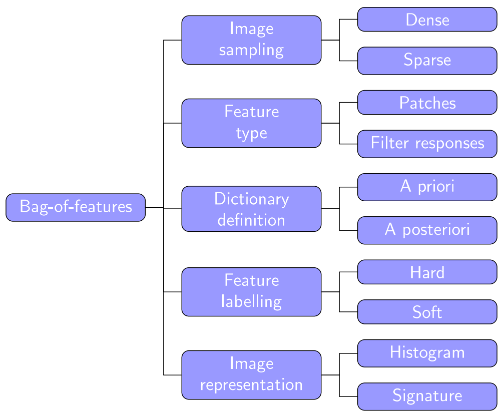


Image sampling: *dense* vs. *sparse*

- Dense: image is probed exhaustively at each pixel.
  - Very common approach
  - Avoids information loss
  - Can be computationally expensive
- Sparse: only a subset of pixels is considered. Common strategies: random sampling, equally-spaced sub-sampling, preliminary detection of 'salient points'.
  - Can reduce computing time by orders of magnitude
  - May provoke significant information loss

Comparison between the two approaches gave contradictory results:

- Jiang, Ngo, and Yang (2007) found comparable results between dense and sparse sampling...
- ...but Jurie and Triggs (2005) determined that dense sampling can be better, particularly with inhomogeneous images.

Feature type: *patches* vs. *filter responses* I

- Patches: pixels' intensities are used directly with no preliminary filtering.
  - Easy to implement
  - Computationally fast
  - Can be sensitive to noise
    - Examples: *Co-occurrence matrices* (Haralick, Shanmugam, and Dinstein, 1973); *local binary patterns* (Ojala, Pietikäinen, and Mäenpää, 2002); *patch-based classifiers* (Varma and Zisserman, 2009); random projections (Liu and Fieguth, 2012).
- Filter responses: features are computed from the output of linear filter banks.
  - Designing filter banks is not a straightforward task
  - Computationally demanding
  - Robust to noise
    - Examples: *3D textons* (Leung and Malik, 2001); *VZ classifier* (Varma and Zisserman, 2005); *basic image features* (Crosier and Griffin, 2010).

Feature type: *patches* vs. *filter responses* II

For a long time filtering was thought to be the best option due to the similarity with the human vision system (Daugman, 1985; Unser, 1986)...

...however, Varma and Zisserman (2003) showed that we can safely get rid of filter banks and obtain as good results using image patches with support as small as a 3×3 window.

More recently, in a comparative study, Ghita et al. (2012) found comparable classification accuracy between image patches (LBP) and filter responses (Gabor filters).

Dictionary definition: *a priori* vs. *a posteriori* I

- A priori ('unlearnt' methods): the dictionary is defined independently of the data.
  - General-purpose
  - Computationally fast and usually easy to implement
  - Deterministic
    - Examples: *Co-occurrence matrices* (Haralick, Shanmugam, and Dinstein, 1973); *texture units* (He and Wang, 1990); *local binary patterns* (Ojala, Pietikäinen, and Mäenpää, 2002); *binary gradient contours* (Fernández, Álvarez, and Bianconi, 2011).
- A posteriori ('learnt' methods): the dictionary ('codebook') is learnt from some training data.
  - Specific for certain classes of images
  - Computationally demanding
  - Usually involves some random procedure for clustering
    - Examples: *3D textons* (Leung and Malik, 2001); *VZ classifier* (Varma and Zisserman, 2005); *patch-based classifier* (Varma and Zisserman, 2009); *extended local binary patterns* (Liu et al., 2012).

Dictionary definition: *a priori* vs. *a posteriori* II

The a priori/a posteriori dilemma is a topic where scientific interest is high.

A priori approaches are generally faster (they don't require codebook generation). They also proved effective in many practical applications.

Yet a priori method could be insufficient when features tend to cluster over a limited portion of the feature space: in such cases a posteriori approaches could be more effective.

Rouco et al. (2011) correctly noted the following:

- a priori methods are good for broad-domain applications and large image datasets;
- a posteriori (data-driven) strategies suit better small databases containing few and similar texture classes.

Label assignment: *hard* vs. *soft*

- Hard: each image patch or filter response is assigned to *one word only*.
  - Easy to implement
  - Computationally fast
  - Can be sensitive to noise
    - Examples: *Texture units* (He and Wang, 1990); *textons* (Malik et al., 1999); *local binary patterns* (Ojala, Pietikäinen, and Mäenpää, 2002); *patch-based classifiers* (Varma and Zisserman, 2009).
- Soft: each image patch or filter response can be assigned to *more than one word*.
  - Requires defining suitable membership functions
  - Computationally demanding
  - Robust to noise; can be useful to model 'visual word ambiguity' (van Gemert et al., 2010)
    - Examples: fuzzy texture spectrum (Taur and Tao, 1998; Barcelo, Montseny, and Sobrevilla, 2005); *soft local binary patterns* (Ahonen and Pietikäinen, 2007); fuzzy local binary patterns (Iakovidis, Keramidas, and Maroulis, 2008).

Image representation: *histogram* vs. *signature*

- Histograms: an ordered, fixed-length vector containing the probability of occurrence of each visual word (but not the words).
  - Easy to implement
  - Easy to measure the distance between different textures
    - Examples: most texture descriptors.
- Signature: an unordered, variable-length set containing both the words and their probability of occurrence.
  - Compact description
  - Difficult to measure the distance between different textures
    - Examples: *earth mover's distance* (Rubner, Tomasi, and Guibas, 1998); *sparse texture representation* (Lazebnik, Schmid, and Ponce, 2005).

Neighbourhoods I

Neighbourhoods are sets of points on the plane.

Can be of any shape: even elliptical, hyperbolic and spiral neighbourhoods have been proposed (Nanni, Lumini, and Brahnam, 2010).

Most typically they are *balls*, i.e. sets of points within a given distance from the center:

Neighbourhood

$$\mathcal{N}(P) = \{\mathbf{x}_p \in \mathbb{R}^2 \;:\; d(\mathbf{x}_p, \mathbf{x}_c) = r,\ 0 \le p < P-1\}$$

where $\mathbf{x}_p$ represents the generic point, P the number of points, $r \in \mathbb{R}$ the radius, $\mathbf{x}_c \in \mathbb{R}^2$ the center, and d a proper *metric*.

The points can coincide with the image's pixels or not: if not, the corresponding grey-values are estimated through interpolation.

Neighbourhoods II

The actual shape of the neighbourhood depends on the type of metric used. Let us briefly recall what a metric is (Dugundji, 1966, Ch. 9):

Metric

A metric (or distance function) on a set X is a map $X \times X \to \mathbb{R}$ with the properties:

- $d(\mathbf{x}_i, \mathbf{x}_j) \ge 0$ for each pair $\mathbf{x}_i, \mathbf{x}_j$ (non-negativity);
- $d(\mathbf{x}_i, \mathbf{x}_j) = 0$ if and only if $\mathbf{x}_i = \mathbf{x}_j$(coincidence axiom);
- $d(\mathbf{x}_i, \mathbf{x}_j) = d(\mathbf{x}_j, \mathbf{x}_i)$ for each pair $\mathbf{x}_i, \mathbf{x}_j$ (symmetry);
- $d(\mathbf{x}_i, \mathbf{x}_k) \le d(\mathbf{x}_i, \mathbf{x}_j) + d(\mathbf{x}_j, \mathbf{x}_k)$ for each triple $\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k$ (triangle inequality).

Neighbourhoods III

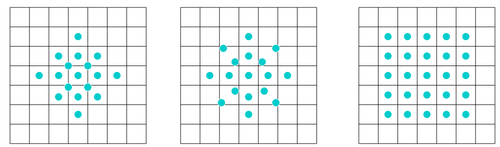
Now consider, for instance, the Minkowski distance.

Minkowski distance

The Minkowski distance of order s between two points $\mathbf{x}_1$, $\mathbf{x}_2 \in \mathbb{R}^n$ is defined as:

$$d(\mathbf{x}_1, \mathbf{x}_2) = \sqrt[s]{\sum_{r=1}^{n} \left(\|x_{1r} - x_{2r}\|\right)^s}$$

Corresponding neighbourhoods for different values of s and $d = 2$ are:



$s = 1$ (Manhattan) $s = 2$ (Euclidean) $s = \infty$ (Tchebychev)

Patterns I

A *pattern* $\mathcal{P}(P, K)$ is an instance of a neighbourhood where each point of the neighbourhood is associated to a whole number I_p in the following way:

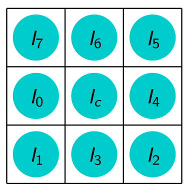
Pattern

$\mathcal{P}(P, K) = \{(\mathbf{x}_p, I_p) \ : \ I_p \in \mathbb{R}; 0 \leq I_p \leq G - 1\}$, where $p \in \{0, \ldots, P-1\}$.

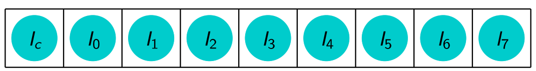
- For the sake of simplicity we assume that I_p is the grey-value (intensity) of the p-th point.
- The number of possible pattern generated by a neighbourhood of P points and G levels is G^P.

Patterns II

Now let's look at patterns more closely. For the sake of simplicity let us consider a 3×3 square neighbourhood and a generic pattern based on it:



If we unfold the pattern...



...we see that patterns are just points in the P-dimensional unit hypercube.

Equivalent patterns

To describe texture we could count how many times each pattern appears in an image. This would however generate a very high-dimensional descriptor: with $P = 4$ and $G = 256$ we would have more than four billion patterns!

The solution is therefore to partition the patterns into *equivalence classes* and count how many times *each class* appears.

The gist of the question is therefore that of defining a suitable partition of the patterns' space. Mathematically we seek a function f which establishes an equivalence relation $\sim$ acting as follows:

$$\mathcal{P}_a \sim \mathcal{P}_b \leftrightarrow f(\mathcal{P}_a) = f(\mathcal{P}_b)$$

Where $\mathcal{P}_a$ and $\mathcal{P}_b$ are two generic patterns.

Histograms of equivalent patterns I

We are almost ready to define what histograms of equivalent patterns are. Beforehand, we need to define what a texture descriptor is (sorry for the excess of formalism).

Texture descriptor

A texture descriptor is a function F which takes and image $\mathbf{I}$ as input and returns a vector $\mathbf{h}$ as output:

$$\mathbf{h} = F(\mathbf{I})$$

Where **h** is usually referred to as the *feature vector*.

Histograms of equivalent patterns II

Here comes the definition:

Histograms of equivalent patterns

Histograms of equivakent patterns (HEP) is a class of texture descriptors for which the k-th element of the feature vector is computed as follows:

$$h_k = \frac{1}{D} \sum_{r=R_{\min}}^{R_{\max}} \sum_{c=C_{\min}}^{C_{\max}} \delta\left[f\left(\mathcal{P}_{r,c}, \mathbf{T}\right) - k\right] \tag{1}$$

Where r and c are the row- and column-wise pixel indices, $\mathcal{P}_{r,c}$ the pattern centered at (r, c), $\mathbf{T}$ a vector of parameters, D a normalizing factor to guarantee that the feature vector sums one and δ the function defined in Eq. 1.

$$\delta(x) = \begin{cases} 1, & \text{if } x = 0 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

Histograms of equivalent patterns III

In plane words histograms of equivalent patterns work as follows:

- scan the image by steps of one pixel (image sampling = *dense*);
- for each position assign the pattern (feature type = *patches*) to one among K possible classes (feature labelling = *hard*);
- the corresponding k-th element of the feature vector is incremented by $1/D$ (image representation = *histogram*).

The resulting feature vector represents the probability of occurrence of each class.

The definition of a texture descriptor belonging to the HEP is therefore a matter of defining a suitable function f (*kernel function*).

Local binary patterns and related methods in the framework of the HEP

Shortly we will see that many texture descriptors are actually members of the HEP family. Among the others are (ordered by date of first publication):

- *Texture spectrum* (He and Wang, 1990);
- *Rank transform* (Zabih and Woodfill, 1994);
- *Local binary patterns* (Ojala, Pietikäinen, and Harwood, 1996);
- *Improved local binary patterns* (Jin et al., 2004);
- *Center-symmetric local binary patterns* (Heikkilä, Pietikäinen, and Schmid, 2006);
- *Binary gradient contours* (Fernández, Álvarez, and Bianconi, 2011);

...and many others

What really makes the descriptors different from one another is the definition of the kernel function f.

Texture spectrum (TS0)

The grey-levels of the pixels in the periphery are compared with the value of the central pixel.

$$\begin{array}{|c|c|c|}\hline 0.39 & 0.52 & 0.65 \\ \hline 0.75 & 0.65 & 0.85 \\ \hline 0.35 & 0.81 & 0.61 \\ \hline \end{array} \longrightarrow \begin{array}{|c|c|c|}\hline 0 & 0 & 1 \\ \hline 1 & & 2 \\ \hline 0 & 2 & 0 \\ \hline \end{array} \times \begin{array}{|c|c|c|}\hline 3^7 & 3^6 & 3^5 \\ \hline 3^0 & & 3^4 \\ \hline 3^1 & 3^2 & 3^3 \\ \hline \end{array} \quad \sum = 426$$

The kernel function is:

$$f\left(\mathcal{P}\right) = \sum_{j=0}^{7} t\left(I_j - I_c, 0\right) \tag{3}$$

Where $t(x)$ is the ternary thresholding function:

$$t(x, T) = \begin{cases} 0, & \text{if } x < -T \\ 1, & \text{if } -T \leq x \leq T \\ 2, & \text{otherwise} \end{cases} \tag{4}$$

Rank transform (RT)

Number of pixels in the periphery of the pattern whose grey-values are less than that of the central pixel.

$$\begin{array}{|c|c|c|}\hline 0.39 & 0.52 & 0.65 \\ \hline 0.75 & 0.65 & 0.85 \\ \hline 0.35 & 0.81 & 0.61 \\ \hline \end{array} \longrightarrow \begin{array}{|c|c|c|}\hline 1 & 1 & 0 \\ \hline 0 & & 0 \\ \hline 1 & 0 & 1 \\ \hline \end{array} \quad \sum = 4$$

The kernel function is:

$$f\left(\mathcal{P}\right) = \sum_{j=0}^{7} b\left(I_c - I_j - 1\right) \tag{5}$$

Local binary patterns (LBP)

The grey-levels of the pixels in the periphery are compared with the value of the central pixel.

$$\begin{array}{|c|c|c|}\hline 0.39 & 0.52 & 0.65 \\ \hline 0.75 & 0.65 & 0.85 \\ \hline 0.35 & 0.81 & 0.61 \\ \hline \end{array} \longrightarrow \begin{array}{|c|c|c|}\hline 0 & 0 & 1 \\ \hline 1 & & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array} \times \begin{array}{|c|c|c|}\hline 2^7 & 2^6 & 2^5 \\ \hline 2^0 & & 2^4 \\ \hline 2^1 & 2^2 & 2^3 \\ \hline \end{array} \quad \sum = 53$$

The kernel function is:

$$f\left(\mathcal{P}\right) = \sum_{j=0}^{7} b\left(I_j - I_c\right) \tag{6}$$

Where $b(x)$ is the binary thresholding function:

$$b(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Improved local binary patterns (ILBP)

The grey-levels of all the pixels of the pattern are compared with the average value.

$$\underset{\mu = 0.62}{\begin{array}{|c|c|c|}\hline 0.39 & 0.52 & 0.65 \\ \hline 0.75 & 0.65 & 0.85 \\ \hline 0.35 & 0.81 & 0.61 \\ \hline \end{array}} \longrightarrow \begin{array}{|c|c|c|}\hline 0 & 0 & 1 \\ \hline 1 & 1 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array} \times \begin{array}{|c|c|c|}\hline 2^7 & 2^6 & 2^5 \\ \hline 2^0 & 2^8 & 2^4 \\ \hline 2^1 & 2^2 & 2^3 \\ \hline \end{array} \quad \sum\text{-1} = 309$$

The kernel function is:

$$f\left(\mathcal{P}\right) = b\left(I_j - \mu\right) + \sum_{j=0}^{7} b\left(I_j - \mu\right) - 1 \tag{8}$$

Where μ is the average intensity of the pattern:

$$\mu = \frac{1}{8}\left(I_c + \sum_{j=0}^{7} I_j\right) \tag{9}$$

Center-symmetric local binary patterns (CS-LBP)

Pairwise comparsion of the grey-values between couples of diametrically opposite pixels, e.g.: (I_0, I_4), (I_1, I_5), etc.

0.39	0.52	0.65
0.75	0.65	0.85
0.35	0.81	0.61

$\longrightarrow$

0		
0	1	1

$\times$

2^0		
2^1	2^2	2^3

$\sum = 12$

The kernel function is:

$$f\left(\mathcal{P}\right) = \sum_{j=0}^{3} b\left(I_j - I_{j+4}\right) 2^j \tag{10}$$

Binary gradient contours (BGC)

Also referred to as transition LBP (tLBP)

Pairwise comparison of the grey-values between adjacent pixels in the periphery.

0.39	0.52	0.65
0.75	0.65	0.85
0.35	0.81	0.61

$\longrightarrow$

0	1	1
1		1
0	1	0

$\times$

2^7	2^6	2^5
2^0		2^4
2^1	2^2	2^3

$\sum$ -1 = 116

The kernel function is:

$$f\left(\mathcal{P}\right) = \sum_{j=0}^{7} b\left[I_j - I_{(j+1) \bmod 8}\right] 2^j - 1 \tag{11}$$

Geometrical interpretation

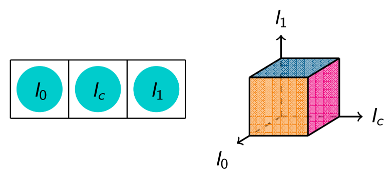
Let's recap some key concepts:

- Texture can be described through the distribution of local patterns;
- Local patterns are *points* in the P-dimensional hypercube (pattern space);
- Histograms of equivalent patterns establish partitions of the pattern space into *equivalence classes*;
- The partition is encoded by suitable *kernel functions*;
- Most kernel functions are systems of *linear inequalities* in the I_j (i.e.: *hyperplanes* in the P-dimensional space);
- The partitions are regions bounded by hyperplanes (*polytopes*).

A motivational example

To clarify the concepts let us consider a reduced-dimension situation, i.e.: a three-pixel neighbourhood.

In this case the corresponding pattern space is the standard three-dimensional cube.



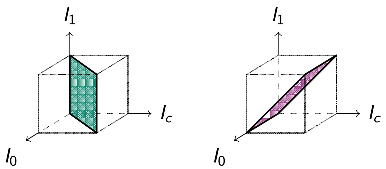
1×3 Pattern   Pattern space

Local binary patterns – revisited I

The partition of the pattern space is based on the following inequalities:

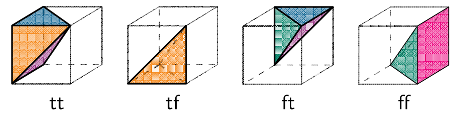
$$\begin{cases} I_0 - I_c \geq 0 \\ I_1 - I_c \geq 0 \end{cases} \tag{12}$$

Geometrically we have two splitting planes:



Local binary patterns – revisited II

Which give rise to *four* partitions into as many convex polytopes:



Where symbols 't' and 'f' in the code indicate, from left to right, respectively which inequalities in Eq. 12 each partition satisfies (t) or not (f).

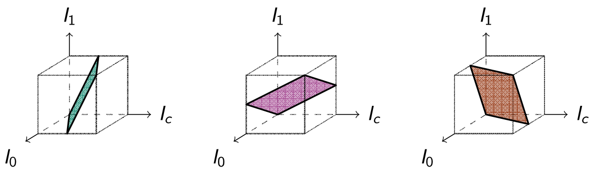
Note that each partition can be regarded as a *word* of an *a priori* visual dictionary and vice-versa.

Improved local binary patterns – revisited I

Here the inequalities that define the partitions of the pattern are:

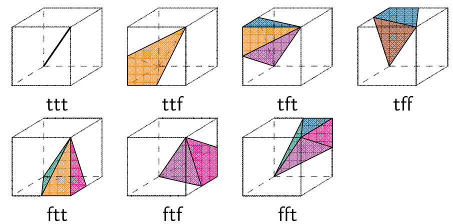
$$\begin{cases} I_0 - \frac{I_0+I_c+I_1}{3} \geq 0 \\ I_c - \frac{I_0+I_c+I_1}{3} \geq 0 \\ I_1 - \frac{I_0+I_c+I_1}{3} \geq 0 \end{cases} \tag{13}$$

Geometrically they represent three planes:



Improved local binary patterns – revisited II

Which produce *seven* partitions into as many convex polytopes:



Things to notice:

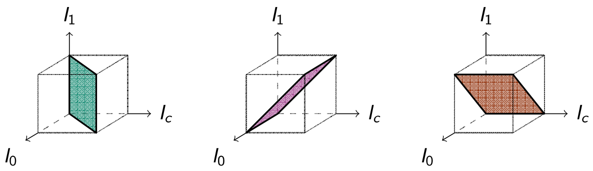
- Combination 'fff' is impossible by definition: I_0, I_c and I_1 cannot be all less than their average at the same time;
- Combination 'ttt' produce a polytope of dimension $P - 1$ (main diagonal of the cube).

Binary gradient contours – revisited I

In this case the inequalities that define the partitions of the pattern space are:

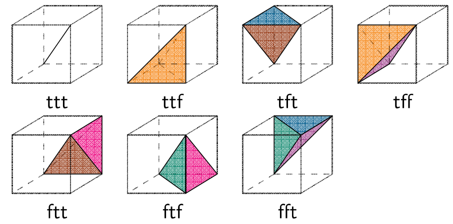
$$\begin{cases} I_0 - I_c \geq 0 \\ I_c - I_1 \geq 0 \\ I_1 - I_0 \geq 0 \end{cases} \tag{14}$$

Again they represent three planes:



Binary gradient contours – revisited II

Which produce *seven* partitions into as many convex polytopes:



Things to notice:

- As in the case of ILBP, combination 'fff' is impossible by definition: the three inequalities $I_0 < I_c$, $I_c < I_1$ and $I_1 < I_0$ cannot be satisfied simoultaneously;
- As in the case of ILBP, combination 'ttt' produce a polytope of dimension $P - 1$.

An interesting exercise I

An interesting exercise consists of computing, for each descriptor, the volume of each partition it generates[2].

If we assume a uniform density of the pattern space, the volume of a partition represents the *a priori* probability of a pattern to belong to that partition.

We have:

- Local binary patterns (1×3 configuration)
  - V(tt) $= \frac{2}{6}$, V(tf) $= \frac{1}{6}$, V(ft) $= \frac{1}{6}$, V(ff) $= \frac{2}{6}$;
- Improved local binary patterns (1×3 configuration)
  - V(ttt) $= 0$, V(ttf) $= \frac{1}{6}$, V(tft) $= \frac{1}{6}$, V(tff) $= \frac{1}{6}$, V(ftt) $= \frac{1}{6}$, V(ftf) $= \frac{1}{6}$, V(fft) $= \frac{1}{6}$;
- Binary gradient contours (1×3 configuration)
  - V(ttt) $= 0$, V(ttf) $= \frac{1}{6}$, V(tft) $= \frac{1}{6}$, V(tff) $= \frac{1}{6}$, V(ftt) $= \frac{1}{6}$, V(ftf) $= \frac{1}{6}$, V(fft) $= \frac{1}{6}$;

An interesting exercise II

Some considerations:

- The volumes of the partitions (polytopes) represent the *a priori* probabilities of occurrence of the corresponding visual words;
- The a priori probabilities of the partitions (= visual words) generated by each descriptor can be different.

[2] For computing the volumes of polytopes see Lawrence (1991) and Beck and Robins (2007)

Efficiency of a texture descriptor I

A texture descriptor can be regarded as an information source emitting symbols (visual words) with a rate p_m, $m \in \{0, \ldots, M-1\}$ over an M-ary alphabet.

The efficiency of such a channel is defined as follows (**ZORZI:1965**):

$$e = \frac{-\sum_{m=0}^{M-1} p_m \log_2 p_m}{\log_2 M} \qquad (15)$$

Clearly the efficiency maximum when the symbols are equally likely. Since we

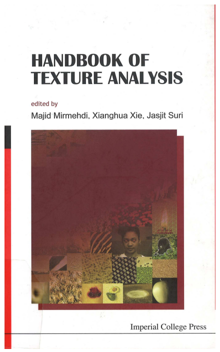
Efficiency of a texture descriptor II

know the volume of the polytopes we can estimate the theoretical efficiency of the descriptors (in the 1×3 model):

- $e_{\text{LBP}} = \frac{-2\times\left(\frac{2}{6}\log_2\frac{2}{6}+\frac{1}{6}\log_2\frac{1}{6}\right)}{\log_2 4} = 0.9591$
- $e_{\text{ILBP}} = \frac{0-6\times\left(\frac{1}{6}\log_2\frac{1}{6}\right)}{\log_2 7} = 0.9208$
- $e_{\text{BGC}} = \frac{0-6\times\left(\frac{1}{6}\log_2\frac{1}{6}\right)}{\log_2 7} = 0.9208$

with the convention that $p_m \log_2 p_m = 0$ if $p_m = 0$.

Further readings



Acknowledgements

I am very grateful to Prof. Fabrizio Smeraldi (School of Electronic Engineering and Computer Science, Queen Mary, University of London) for making this visit possible.

I would also like to thank my institution (Department of Engineering, Universit degli Studi di Perugia, Italy) for the financial support under project 41IMMA13FB – *Image processing methods and applications to the industry and other sectors.*

Most of the material presented in the transparencies is the result of a long and fruitful cooperation with Prof. Antonio Fernández (School of Industrial Engineering, Universidade de Vigo, Spain). I am indebted to him for the countless discussions on these topics.

References I

- T. Ahonen and M. Pietikäinen. "Soft histograms for local binary patterns". In: *Proceedings of the Finnish Signal Processing Symposium (FINSIG 2007)*. Oulu, Finland, Aug. 2007.
- J. Dugundji. *Topology*. Boston: Allyn and Bacon, Inc., 1966.
- A. Barcelo, E. Montseny, and P. Sobrevilla. "On Fuzzy Texture Spectrum for Natural Microtextures Characterization". In: *Proceedings EUSFLAT-LFA 2005*. 2005, pp. 685–690.
- M. Beck and S. Robins. *Computing the Continuous Discretely. Integer-point Enumeration in Polyhedra*. New York: Springer, 2007.
- M. Crosier and L. D. Griffin. "Using Basic Image Features for Texture Classification". In: *International Journal of Computer Vision* 88.3 (2010), pp. 447–460.
- J.G. Daugman. "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters". In: *Journal of the Optical Society of America A* 2 (7 1985), pp. 1160–1169.
- E. R. Davies. "Introduction to Texture Analysis". In: *Handbook of texture analysis*. Ed. by M. Mirmehdi, X. Xie, and J. Suri. Imperial College Press, 2008, pp. 1–31.

References II

- A. Fernández, M. X. Álvarez, and F. Bianconi. "Image classification with binary gradient contours". In: *Optics and Lasers in Engineering* 49.9-10 (2011), pp. 1177–1184.
- J. C. van Gemert et al. "Visual word ambiguity". In: *IEEE Transactions on Patterns Analysis and Machine Intelligence* 32.7 (2010), pp. 1271–1283.
- O. Ghita et al. "Local binary patterns versus signal processing texture analysis. A study from a performance evaluation perspective". In: *Sensor Review* 32 (2012), pp. 149–162. DOI: 10.1108/02602281211209446.
- R. M. Haralick, K. Shanmugam, and I. Dinstein. "Textural Features for Image Classification". In: *IEEE Transactions on Systems, Man, and Cybernetics* 3.6 (1973), pp. 610–621.
- D.-C. He and L. Wang. "Texture Unit, Texture Spectrum, And Texture Analysis". In: *IEEE Transactions on Geoscience and Remote Sensing* 28.4 (1990), pp. 509–512.
- M. Heikkilä, M. Pietikäinen, and C. Schmid. "Description of Interest Regions with Center-Symmetric Local Binary Patterns". In: *Proceedings of the 5th Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP 2006)*. Ed. by Prem K. Kalra and Shmuel Peleg. Vol. 4338. Lecture Notes in Computer Science. Madurai, India, Dec. 2006, pp. 58–69.

References III

- D. K. Iakovidis, E. G. Keramidas, and D. Maroulis. "Fuzzy Local Binary Patterns for Ultrasound Texture Characterization". In: *Proceedings of the 5th International Conference on Image Analysis and Recognition (ICIAR 2008)*. Ed. by Aurélio Campilho and Mohamed Kamel. Vol. 5112. Lecture Notes in Computer Science. Póvoa de Varzim, Portugal, June 2008, pp. 750–759. DOI: 10.1007/978-3-540-69812-8_74.
- Y.-G. Jiang, C.-W. Ngo, and Jun Yang. "Towards optimal bag-of-features for object categorization and semantic video retrieval". In: *Proceedings of the 6th ACM international conference on Image and Video Retrieval (CIVR'07)*. 2007, pp. 494–501. DOI: 10.1145/1282280.1282352.
- H. Jin et al. "Face detection using improved LBP under Bayesian framework". In: *Proceedings of the 3rd International Conference on Image and Graphics*. 2004, pp. 306–309. DOI: 10.1109/ICIG.2004.62.
- F. Jurie and B. Triggs. "Creating efficient codebooks for visual recognition". In: *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05)*. Vol. 1. 2005, pp. 604–610.
- J. Lawrence. "Polytope volume computation". In: *Mathematics of Computation* 57.195 (1991), pp. 259–271.

References IV

- S. Lazebnik, C. Schmid, and J. Ponce. "A Sparse Texture Representation Using Local Affine Regions". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.8 (2005), pp. 1265–1278.
- T. Leung and J. Malik. "Representing and Recognizing the Visual Appearance of Materials using Three-dimensional Textons". In: *International Journal of Computer Vision* 43.1 (2001), pp. 29–44.
- L. Liu and P. W. Fieguth. "Texture classification from random features". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.3 (2012), pp. 574–586.
- L. Liu et al. "Extended local binary patterns for texture classification". In: *Image and Vision Computing* 30.2 (2012), pp. 86–99.
- A Magista. *Grande guida Università 2015-2016*. La Repubblica grandi guide. July 2015.
- J. Malik et al. "Textons, contours and regions: cue integration in image segmentation". In: *The Proceedings of the Seventh IEEE International Conference on Computer Vision (ICCV'99)*. Vol. 2. 1999, pp. 918–925.
- L. Nanni, A. Lumini, and S. Brahnam. "Local binary patterns variants as texture descriptors for medical image analysis". In: *Artificial Intelligence in Medicine* 49.2 (2010), pp. 117–125.

References V

- T. Ojala, M. Pietikäinen, and D. Harwood. "A comparative study of texture measures with classification based on feature distributions". In: *Pattern Recognition* 29.1 (1996), pp. 51–59.
- T. Ojala, M. Pietikäinen, and T. Mäenpää. "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.7 (2002), pp. 971–987.
- M. Petrou and P. García Sevilla. *Image Processing. Dealing with Texture.* Wiley Interscience, 2006.
- J. Rouco et al. "Texture description in local scale using texton histograms with quadrature filter universal dictionaries". In: *IET Computer Vision* 5.4 (2011), pp. 211–221.
- Y. Rubner, C. Tomasi, and L. J. Guibas. "A Metric for Distributions with Applications to Image Databases". In: *Proceedings of the Sixth International Conference on Computer Vision (ICCV'98)*. 1998, pp. 59–66.
- J. Sklansky. "Image Segmentation and Feature Extraction". In: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-8.4 (1978), pp. 460–473.
- H. Tamura, T. Mori, and T. Yamawaki. "Textural Features Corresponding to Visual Perception". In: *IEEE Transactions on Systems, Man, and Cybernetics* 8.6 (1978), pp. 460–473.

References VI

- J. S. Taur and C.-W. Tao. "Texture classification using a fuzzy texture spectrum and neural networks". In: *Journal of Electronic Imaging* 7.1 (1998), pp. 29–35.
- M. Unser. "Sum and Difference Histograms for Texture Classification". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.1 (Jan. 1986), pp. 118–125.
- M. Varma and A. Zisserman. "Texture Classification: Are Filter Banks Necessary?" In: *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03)*. Vol. 2. June 2003, pp. 691–698. DOI: 10.1109/CVPR.2003.1211534.
- M. Varma and A. Zisserman. "A Statistical Approach to Texture Classification from Single Images". In: *International Journal of Computer Vision* 62.1-2 (2005), pp. 61–81.
- M. Varma and A. Zisserman. "A statistical approach to material classification using image patch exemplars". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.11 (2009), pp. 2032–2047.
- X. Xie and M. Mirmehdi. "A galaxy of texture features". In: *Handbook of texture analysis*. Ed. by M. Mirmehdi, X. Xie, and J. Suri. Imperial College Press, 2008, pp. 375–406.
- R. Zabih and J. Woodfill. "Non-parametric Local Transforms for Computing Visual Correspondence". In: *Proceedings of the 3rd European Conference on Computer Vision (ECCV 1994)*. Stockholm, Sweden: Springer-Verlag, May 1994, pp. 151–158.