

Texture

For both first-order, and second-order statistics, it is common to use a sliding window approach, that in some sense resembles the filtering discussed in the first weeks notes. For each pixel $[i, j]$ in the original image, you compute some features based on a neighbourhood of this pixel. This neighbourhood is sometimes called a window, and its size is somewhat difficult to determine. Small window-sizes gives precise estimates of region boundaries, but imprecise feature values. Large window sizes on the other hand, gives precise feature values, but imprecise boundaries between regions.

Table of contents.

- 1. order statistics
- 2. order statistics
- Laws' texture energy

1. order statistics

For each pixel $[i, j]$ in an image, we compute a normalized histogram h of the grey-level intensities in the region defined by the window, such that $h(x)$ gives the frequency of intensity x in the window neighbourhood of $[i, j]$. From this we can extract several features, which we denote ϕ , and each pixel will have a feature value assigned to it.

Mean

$$\phi = \mu = \sum_x xh(x)$$

Hardly usefull.

Variance

$$\phi = \sigma^2 = \sum_x (x - \mu)^2 h(x)$$

Measure for roughness

Skewness

$$\phi = \sum_x (x - \mu)^3 h(x)$$

Measure if the texel is lighter or darker than average.

Kurtosis

$$\phi = \sum_x (x - \mu)^4 h(x) - 3$$

How peaked the greylevel distribution is.

Entropy

$$\phi = - \sum_x \log(h(x))h(x)$$

A measure of the uniformity of the distribution. Has a minimum value of 0 if there is only one greylevel intensity present in the region (very homogenous), and attends its maximum if all greylevel intensities are present with equal frequency (uniform distribution of the image intensities). This maximum can be computed quite easily by realizing that for a uniform histogram, all values $h(x) = 1/N_g$, where N_g are the number of gray levels. The maximum is thus $\log(N_g)$.

Energy

$$\phi = \sum_x h(x)h(x)$$

Also a measure of the uniformity of the distribution, but somewhat the opposite of the entropy measure. Reaches minimum value of $1/N_g$, when all intensities are present with equal frequency (uniformly distribution of greylevel intensities). Reaches its maximum of 1 for a region with only one greylevel intensity.

Smoothness

$$\phi = 1 - \frac{1}{1+\sigma^2}$$

Tends to 0 for homogenous regions, and to 1 for regions with high variance.

Coefficient of variance

$$\phi = \frac{\sigma}{\mu}$$

Yet again a measure of uniformity.

There are several problems with first order statistics. As is obvious, it can discriminate between regions, but it is completely isotropic, and in that sense, it can not gather information about the orientation of the texel.

2. order statistics

Assume you have an $M \times N$ image f , with b bits per element, and that $f[i, j]$ gives the intensity in pixel $[i, j]$. GLCM is then a function $G : \mathcal{G}^2 \rightarrow \mathbb{N}$, where \mathcal{G} is the set of all intensities in the original image. $G[x, y]$ is the number of times we go from intensity $f[i, j] = x$ to intensity $f[i + \Delta i, j + \Delta j] = y$ when we are traversing the index set $\mathcal{I} = \{(i, j) : i, (i + \Delta i) \in [0, M) \wedge j, (j + \Delta j) \in [0, N)\}$ That is

$$G[x, y] = |\{(i, j) : x = f[i, j]; y = f[i + \Delta i, j + \Delta j]; \forall (i, j) \in \mathcal{I}\}|.$$

note that as before, when analyzing a whole image, it is common to compute the glcm for a local window surrounding a pixel $[i, j]$ for all pixels in the image. In this case, just substitute the dimensions of the whole image M, N by the local dimensions of the local window.

If we normalize G , and view the greylevels as random variables X and Y we get a probability mass function $p_{XY} = G / (\sum_x \sum_y G(x, y))$, where it is implicit that we are summing over the level-set $x, y \in \mathcal{G}$. $p_{XY}(x, y) = Pr(X = x, Y = y)$ can then be viewed as the probability of going from graylevel x to graylevel y if we go a step of $(\Delta i, \Delta j)$ in the image that the GLCM is based on.

For each p_{XY} we can compute a multiple of features ϕ , where each feature is a weighted sum of all elements in the (normalized) GLCM

$$\phi = \sum_x \sum_y w(x, y) p_{XY}(x, y)$$

The weights are either value based $w(x, y) = \alpha(p_{XY}(x, y))$, or position based $w(x, y) = \beta(x, y)$ for some functions α, β . Some different features are presented below.

Max probability

$$\phi = \max_{x, y} p_{X, Y}(x, y)$$

This gives an indication of the strongest response to G .

Element difference moment of order n

$$\phi = \sum_x \sum_y (x - y)^n p_{XY}(x, y)$$

This one have a relatively low value for diagonal dominant G .

GLCM Inertia (contrast, sum of squares inertia)

$$\phi = \sum_x \sum_y (x - y)^2 p_{XY}(x, y).$$

Here $w(x, y) = (x - y)^2$, and this feature is in a way the reciprocal of the homogeneity feature, with zero along the main diagonal, and ascending rapidly away from it. Images with high local greylevel contrast will have high inertia values.

Inverse element difference moment of order n

$$\phi = \sum_x \sum_y \frac{1}{(x - y)^n} p_{XY}(x, y), \quad x \neq y$$

This is the reciprocal of the above, and has relatively high values for diagonal dominant G .

GLCM Homogeneity (inverse difference moment)

$$\phi = \sum_x \sum_y \frac{1}{1 + (x - y)^2} p_{XY}(x, y).$$

Here $w(x, y) = 1/(1 + (x - y)^2)$, and this feature has high values for matrices with high concentration of a few greylevels and low value for matrices with low concentration of many greylevels. ϕ attains its maximum value of 1 on the main diagonal, and decreases rapidly when moving away from it. With this, images with a slowly varying greylevel intensity will have high homogeneity values.

Entropy

$$\phi = - \sum_x \sum_y \log(p_{XY}(x, y)) p_{XY}(x, y)$$

Entropy is a measure for randomness, and have the highest value when all elements of G are equal.

Uniformity (angular second moment)

$$\phi = \sum_x \sum_y p_{XY}(x, y) p_{XY}(x, y)$$

Conversly, *uniformity* attains its *lowest* value when all elements of G are equal.

GLCM mean

$$\phi = \mu = \frac{1}{N_g^2} \sum_x \sum_y G(x, y)$$

GLCM variance

$$\begin{aligned} \phi = \sigma &= \sum_x \sum_y (x - \mu)^2 p_{XY}(x, y) \\ &= \sum_x \sum_y (y - \mu)^2 p_{XY}(x, y) \end{aligned}$$

Notice that this so-called *glcm mean* and *glcm variance* is a measure of the mean and variance of the grey levels in the glcm, and should not be confused with the marginal means and variances defined below.

Marginal statistics

The rest depends on some statistics we define here. We let p_X and p_Y denote the marginal probabilities for X and Y , and from them we can compute their individual means and variances.

$$\begin{aligned}
p_X(x) &= Pr(X = x) \\
&= \sum_y p_{XY}(x, y) \\
p_Y(y) &= Pr(Y = y) \\
&= \sum_x p_{XY}(x, y) \\
\mu_X &= E[X] \\
&= \sum_x x p_X(x) \\
&= \sum_x \sum_y x p_{XY}(x, y) \\
\mu_Y &= E[Y] \\
&= \sum_y y p_Y(y) \\
&= \sum_x \sum_y y p_{XY}(x, y) \\
\sigma_X^2 &= Var[X] \\
&= E[(X - E[X])^2] \\
&= \sum_x (x - \mu_X)^2 p_X(x) \\
&= \sum_x x^2 p_X(x) - \left(\sum_x x p_X(x) \right)^2 \\
\sigma_Y^2 &= Var[Y] \\
&= E[(Y - E[Y])^2] \\
&= \sum_y (y - \mu_Y)^2 p_Y(y) \\
&= \sum_y y^2 p_Y(y) - \left(\sum_y y p_Y(y) \right)^2.
\end{aligned}$$

Note that for a symmetric GLCM, $p_X(x) = p_Y(y)$ and therefore their means and variances are also equal.

Correlation

$$\phi = \sum_x \sum_y \frac{(x - \mu_X)(y - \mu_Y)}{\sigma_X \sigma_Y} p_{XY}(x, y)$$

Correlation is a measure of grey level linear dependence between the pixels at the specified positions relative to each other.

Variance

$$\begin{aligned}
\sigma_X^2 &= \text{Var}[X] \\
&= E[(X - E[X])^2] \\
&= \sum_x (x - \mu_X)^2 p_X(x) \\
&= \sum_x x^2 p_X(x) - \left(\sum_x x p_X(x)\right)^2 \\
\sigma_Y^2 &= \text{Var}[Y] \\
&= E[(Y - E[Y])^2] \\
&= \sum_y (y - \mu_Y)^2 p_Y(y) \\
&= \sum_y y^2 p_Y(y) - \left(\sum_y y p_Y(y)\right)^2.
\end{aligned}$$

GLCM cluster shade

$$\phi = \sum_x \sum_y (x - \mu_X + y - \mu_Y)^3 p_{XY}(x, y).$$

This is a measure of the joint bivariate skewness of the glcm and the weights $w(x, y) = (x - \mu_X + y - \mu_Y)^3$ will attain its minimum at $x = y = \min\{\mathcal{G}\}$ and its maximum at $x = y = \max\{\mathcal{G}\}$. It is an odd function along the axis where $(x - \mu_X) = -(y - \mu_Y)$, where also the weights will be zero. This feature will therefore discriminate images with unequal graylevel histogram. That is, large negative values for low intensity dominant images, and large positive values for high intensity dominant images.

GLCM cluster prominence

$$\phi = \sum_x \sum_y (x - \mu_X + y - \mu_Y)^4 p_{XY}(x, y).$$

Important note

The definitions above can in terms of *position*, be interpreted somewhat differently than what the lecture notes, and this external resource suggests. That is, when traversing x and y , you can interpret them to be indices ranging from 0 to the number of graylevels, or you can interpret them to be actual graylevels (and because of the requantization, these can differ from the indices).

In short, for an image requantized to only contain the gray levels $\{x_1, x_2, x_3, x_4\}$, the notes suggests that features weighted by *position* should be weighted by a function of the explicit indices $\{1, 2, 3, 4\}$, while I believe that it makes more sense to weigh them by a function of the actual gray levels $\{x_1, x_2, x_3, x_4\}$. And this is what is stated in the equations above.

Until I have gotten some clarification, I encourage you to use the definitions from the lecture notes, but this note will stay as it is until I can get a proper explanation. Hopefully this discussion will clear up a few things why I interpreted it the way I did, and also illuminate the official definitions. So even though it can seem confusing, I think that a thorough treatment can be useful. I state two examples to illustrate my point.

Whatever interpretation, the formulas stated above are valid, you only need to change the interpretation of x and y . Also, for all practical purposes, as long as you are consistent, the results in terms of discriminating between texture, should be the same.

INF4300 - DIGITAL IMAGE ANALYSIS

The mean is an *expected value* of a random variable, say X . Suppose that X can take the values $\{x_1, x_2, \dots, x_n\}$ and that the probability of X taking some value x is given by the *probability mass function* $p_X(x)$ (assuming a discrete random variable). Then the expected value $E[X]$ is defined to be by

$$\mu = E[X] := \sum_{i=1}^n x_i p_X(x_i).$$

Now, the lecture notes defines the mean as

$$\mu = \sum_{i=1}^{N_g-1} i p(i)$$

where p is the normalized histogram of the present grey level intensities. This is all good if your gray levels happen to be $\mathcal{G} = \{0, 1, \dots, N_g - 1\}$ but as your image most likely consist of a different set of gray-levels (say evenly distributed between 0 and 255 for a 8bit integer image), this is simply wrong. This is with the assumption that we have requantized our image to only consist of say 8 or 16 graylevels (to use all the original graylevels in the image is simply grotesque from a computation point of view).

As a disclaimer, it could very well be that they have defined their own measure and called it *mean*, but I think that a redefinition of such an established term should be greatly emphasized to avoid confusion. From their definition above, I can interpret it in a few different ways

- You take it at face value, in which case $p(i)$ is interpreted as the probability of getting a value i (which makes little sense since we are only looking at the N_g first values, when in reality our image could have greylevels above the value $N_g - 1$ if you have requantized your image).
- The second interpretation is that they are summing over indices, but somewhat implicitly maps the index to the grey value at this index. This makes sense, and is what I have interpreted it to be. Explicitly this would be

$$\mu = \sum_{i=1}^{N_g-1} \mathcal{G}[i] p(\mathcal{G}[i])$$

- A third interpretation is a mix of the two above, and which I realize now, is most likely what is meant in the lecture notes. When i stands alone it acts as the index (or *position*) i , but in the distribution p we implicitly map it to the appropriate grey-level. This could make sense, since from an implementation point of view, when creating p it would make sense to only include values which are non-zero, in which case the value at $p[i]$ is the value $p(\mathcal{G}[i])$. Mathematically, this does not make sense to me (it is in the least misleading to be calling it the mean). Explicitly, what is computed is

$$\mu = \sum_{i=1}^n i p_X(x_i)$$

which seems somewhat arbitrary, and is clearly not the same as the definition of an expected value. Also, this value (whatever it is) is invariant to the graylevels you choose to requantize your image with, which seems like an odd property to have (but I did not design it, so perhaps this is the meaning).

Example 2: Second order glcm contrast.

INF4300 - DIGITAL IMAGE ANALYSIS

$$\phi = \sum_i \sum_j (i - j)^2 p[i, j].$$

This feature measures contrast by emphasizing entries in the glcm away from the diagonal. The way it is stated above, the weights increases by the square of the distance (measured in indices). My interpretation is that it should rather be

$$\phi = \sum_i \sum_j (x_i - y_j)^2 p(x_i, y_j).$$

To see that, consider the case where the graylevels present in the requantized image is $\{1, 100, 101, 200\}$. In the first case (weighing by indices), the transition from index 1 to index 2 (1 to 100) will be weighted the same as the transition from index 2 to index 3 (100 to 101), namely with 1 (since $(2 - 1)^2 = 1$, and $(3 - 2)^2 = 1$. In my case the weights will be 99^2 in the first transition, and 1^2 in the second case, which reflect the different in gray level transition. After all, in terms of contrast, a transition from gray level 1 to 100 is huge, while a transition from 100 to 101 is barely noticable, so it does really not make any sense to weigh them equally.

Laws' texture energy measure

Another approach to generating texture features is to use local masks to detect various types of texture. Laws developed a texture-energy approach that measures the amount of variation within a fixed-size window.

First, the originl image f is preprocesssed by shifting it to zero mean. This means subtracting the original image by a mean image. This mean image is computed by sliding a smaller window (e.g. 15×15) over the original image, and for each pixel $[i, j]$ in the original image, compute its average value inside the window neighbourhood.

Then we compute 16 filter images. The filters are computed from the following vectors

$$\begin{aligned} L_5 &= \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix} \text{ (Level)} \\ E_5 &= \begin{bmatrix} -1 & -2 & 0 & 2 & 1 \end{bmatrix} \text{ (Edge)} \\ S_5 &= \begin{bmatrix} -1 & 0 & 2 & 0 & -1 \end{bmatrix} \text{ (Spot)} \\ R_5 &= \begin{bmatrix} 1 & -4 & 6 & -4 & 1 \end{bmatrix} \text{ (Ripple)} \end{aligned}$$

And by treating them as a component of a separable filter, one can generate the 16 wanted filter images, e.g.

$$E_5 L_5 = E_5 \otimes L_5 = \begin{pmatrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{pmatrix}$$

where \otimes denotes the outer product. What I mean by this, is that each of the 16 possible combinations $\{L_5 L_5, L_5 E_5, \dots, R_5 R_5\}$ represent a 5×5 filter kernel as the one above. But since they are

INF4300 - DIGITAL IMAGE ANALYSIS

Let each of the resulting filtered images be denoted g_k for $k \in \{E_5, L_5, S_5, R_5\}^2$. From them we can construct the texture energy images h_k by computing the sum of the absolute intensities of the respective feature images in a $(2P + 1) \times (2Q + 1)$ neighbourhood ($P = Q = 15$ is suggested)

$$h_k[i, j] = \sum_{p=-P}^P \sum_{q=-Q}^Q |g_k[i - p, j - q]|.$$

Now, we finish off by averaging some of the energy images. For example, $E_5 L_5$ measures the horizontal edge content, and $L_5 E_5$ measures the vertical edge content, and by averaging them, we get the total edge content. The nine resulting energy maps are then

$$\begin{aligned} h_1 &: (E_5 E_5) \\ h_2 &: (S_5 S_5) \\ h_3 &: (R_5 R_5) \\ h_4 &: \overline{(L_5 E_5, E_5 L_5)} \\ h_5 &: \overline{(L_5 S_5, S_5 L_5)} \\ h_6 &: \overline{(L_5 R_5, R_5 L_5)} \\ h_7 &: \overline{(E_5 S_5, S_5 E_5)} \\ h_8 &: \overline{(E_5 R_5, R_5 E_5)} \\ h_9 &: \overline{(S_5 R_5, R_5 S_5)} \end{aligned}$$

where the names identify from which filters they originate, and the overline between two, means that we have used the average between the two. Large parts of this explanation is based on [this document](#).

•