Texture analysis in the Logarithmic Image Processing (LIP) framework

Muhammad Inam Ul Haq

THESE

Présentée à l'Université Jean Monnet, Saint-Etienne

ECOLE DOCTORALE « Sciences, Ingénierie, Santé »

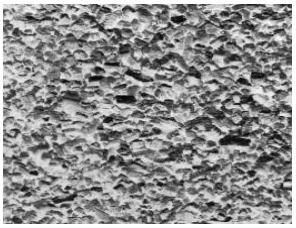
par **Muhammad INAM UL HAQ**

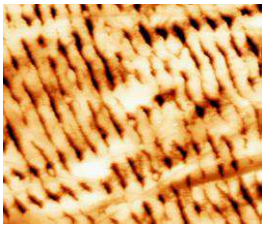
pour l'obtention du titre de : DOCTEUR en SCIENCES

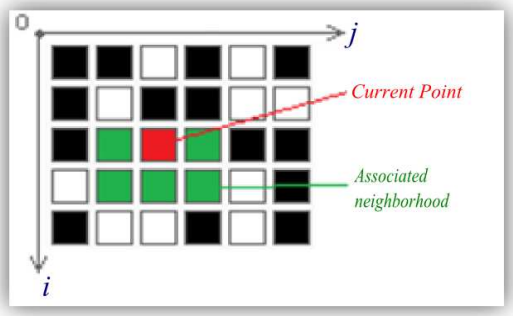
Spécialité : TRAITEMENT D'IMAGES

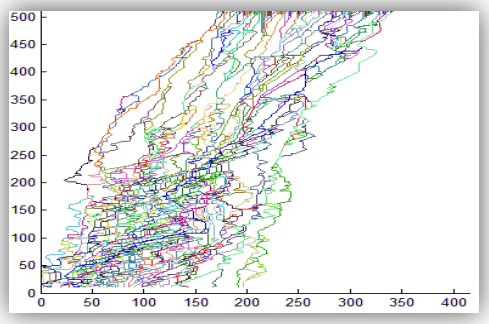
Texture Analysis in the Logarithmic Image Processing (LIP) Framework

Le 27 Juin 2013









Devant le jury composé de :

Prof. Jacques DEMONGEOT, Rapporteur

Prof. Frank MARZANI, Rapporteur

Mr. Guy COURBEBAISSE, Examinateur

Prof. Thierry FOURNEL, Président

Prof. Michel JOURLIN, Directeur

THESIS

Presented at Jean Monnet University, Saint Etienne, France.

ECOLE DOCTORALE “Sciences, Ingénierie, Santé ”

by Muhammad INAM UL HAQ

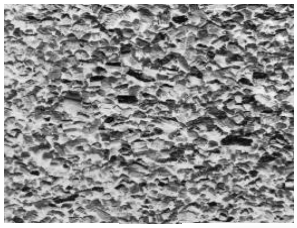
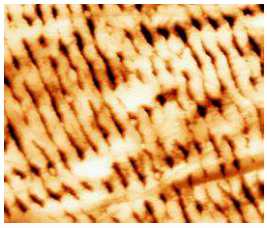
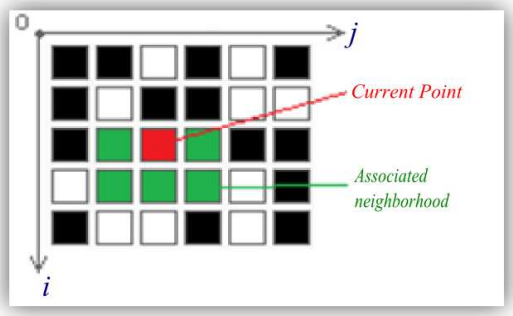
Supervised by Professor Michel JOURLIN

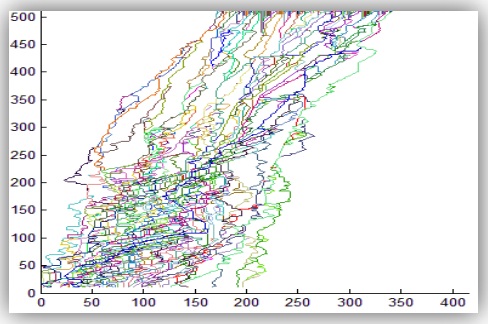
to obtain the title of PhD of Science

Specialty: Image Processing

June 27, 2013.

Texture Analysis in the Logarithmic Image Processing (LIP) Framework



DISSERTATION COMMITTEE :

Reviewers :	Prof. Jacques DEMONGEOT	Laboratoire AGIM, Univ. Joseph Fourier.
	Prof. Frank MARZANI	Laboratoire LE2I, Univ. de Bourgogne.
Examiners :	Mr. Guy COURBEBAISSE	Laboratoire CREATIS, INSA-Lyon.
President :	Prof. Thierry FOURNEL	Université Jean Monnet.
Supervisor :	Prof. Michel JOURLIN	Université Jean Monnet.

Remerciements :

Tout d'abord, je remercie mon Directeur de thèse, Professeur Michel JOURLIN, pour m'avoir proposé un sujet très novateur sur l'analyse d'images texturées mais aussi pour son aide et ses encouragements permanents tout au long de ce travail. Il est certain que sans son aide il m'aurait été très difficile d'atteindre cet objectif.

Je remercie ensuite les deux rapporteurs, Professeurs Franck MARZANI et Jacques DEMONGEOT pour avoir accepté d'évaluer mon manuscrit, ainsi que pour leurs nombreux commentaires et suggestions.

Monsieur Guy COURBEBAISSE a examiné mon travail et participé à mon jury malgré les lourdes de charges qu'il assume à la tête d'un important projet européen. Qu'il en soit sincèrement remercié.

Je veux exprimer ma gratitude au Professeur Thierry FOURNEL pour son analyse détaillée du mémoire et ses suggestions de corrections très précises.

Je remercie également la Commission de l'Education Supérieure du Pakistan, (Higher Education Commission.) pour m'avoir sélectionné et financé mes études en France .

Il serait inconvenant de ne pas mentionner mes collègues du Laboratoire Hubert Curien et mes amis partout, qui m'ont encouragé et témoigné leur amitié.

Enfin, je veux préciser que je n'aurais pas pu obtenir le titre de Docteur sans le soutien de mes parents, mes frères, ma sœur et bien sûr ma femme. Je les remercie tous, et plus particulièrement, mon "Abbu",(père) qui a été mon modèle, à chaque étape de ma vie. Il reste pour moi et pour d'autres personnes, une source d'inspiration inépuisable.

Muhammad INAM UL HAQ

Acknowledgements:

My first and foremost genuine appreciativeness goes to my thesis supervisor Professor Michel JOURLIN to have offered me a very innovative topic on the analysis of textured images, for his guidance, support and encouragement at each step of my thesis. Certainly, without all what he did, it was very difficult to achieve this goal.

Also I am thankful to the two reviewers of my thesis, Professor Franck MARZANI and Professor Jacques DOMENGEOT for accepting my manuscript to review it and for their valuable comments and suggestions.

I am also thankful to Mr. Guy COURBEBAISSE who examined my project and was the part of the dissertation committee despite the heavy work load he assumes as the leader of an important european project.

I also thank Professor Thierry FOURNEL for detailed analysis of my manuscript and his prices suggestions for correcting it.

I am thankful to Higher Education Commission of Pakistan for selecting me for foreign scholarship and financing me during my study in France.

It would be unjust if I do not mention my colleagues in Laboratoire Hubert Curien and everywhere else, who have been very supportive, encouraging and friendly.

The last but certainly not the least, I want to register that it was impossible to have this prestige without the support of my parents, my brothers and sister and surely that of my wife. So I am very much thankful to them. Especially, I want to thank my Abbu (my father as I call him), who is a model for me at each dimension of life and is the source of inspiration for me and many other people.

Muhammad INAM UL HAQ

Table of Contents :

SOMMAIRE

Chapter 1 – Introduction: what is texture?

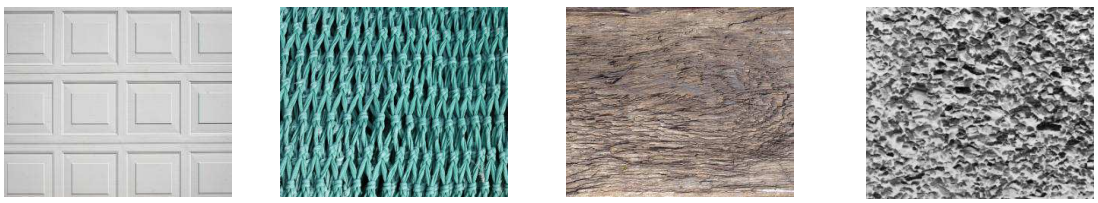
Chapter 1 – Introduction: what is texture?

Definition of Texture:

Texture is referred to the distinctive physical composition or structure of something, especially with respect to the size, shape, and arrangement of its parts: *the texture of sandy soil; the texture of cooked fish.* [1]. Usually a texture is illustrated as smooth or rough, coarse of fine, soft or hard, etc.

Textures can be categorised as *tactile* and *visual* textures. Tactile texture is defined as the physical sensation of some surface i.e. by touching. Visual texture is not a tangible but a visual phenomenon. For example image of a rock or a wooden surface is not visually smooth but have some pattern of (quasi) repeated variations in intensity levels and can be referred as visual texture. Visual texture is easily perceived by human eyes and provides plenty of information about an object. Texture is the property of the surface that gives rise to the local variabilities [2].

It should be noted that this work is about visual textures, so in rest of the document it will be referred to as "texture" unless stated otherwise. Fig.1.1 shows some examples of visual textures.



a. b. c. d.

Fig: 1.1: *Visual Textures*

Normally textures are complex visual patterns composed of entities, or sub-patterns having characteristic like brightness, colour, slope, size, etc. Thus texture can be regarded as a similarity grouping in an image [3]. The local sub-pattern properties give rise to the perceived lightness, uniformity, density, roughness, regularity, linearity, frequency, phase,

directionality, coarseness, randomness, fineness, smoothness, granulation, etc., of the texture as a whole [4].

Initial appearance of Textures:

Initially texture was only termed as the appearance or feeling of a surface like that of the metallic surface or textile surface and in 1939 the initial published research was "Texturen Metallischer Werkstoff (textures in metallic materials)" [5].

One definition of the texture is the property of the surface that gives rise to the local variability. The local variability arises because of the surface roughness which causes random light scattering and hence the increase or decrease of the local reflectance in the direction of view. But this varies case to case as discussed in the following examples.

- A very smooth and clean surface also has this property like white paper to some extent and eggshell more so.
- A surface structure can also be a cause of enhanced or reduced local reflectance like in woven materials which exhibits periodic variation in reflectance.
- The textures of wooden surfaces vary from fine to coarse patterns because of their intrinsic materials no matter how much are they smoothened.
- Water ripples demonstrate coarse texture although it will be a case of rapid temporal development.
- Separate objects like sand on beach, grass leaves of the grassy lawn, bushes in a hedge, collectively gives the appearance of unique texture.
- A spatially homogeneous pattern that typically contains repeated structures, often with some random variations.
- A set of "elements" at periodic or random locations within an image.
- A visual pattern which has a "stationary distribution of features": the texture elements (e.g. texels) is repeated everywhere in the image.

Texels:

A surface is said to be textured if there is large enough number of texels (texture elements), else it is a set of objects. Texels in a texture may appear overlapped or non-overlapped, regular or irregular, directional or non-directional. Textures can vary based on their different property like periodicity, randomness, directionality and orientation.

What give rise to textures?

There is a debate on what should be called as texture and what not? A perfect periodic appearance of intensity variation is generally termed as a periodic pattern not texture and pattern with complete randomness could be called as noise not as texture. It remains a subjective matter based on the observers' judgment. Mostly a pattern is termed as texture which has both randomness and regularity. This is an academic question and can be dealt by giving relative importance if to which extant the mix of the both is to be termed as texture depending on the situation. „Busyness' is an important feature of texture and is independent of „randomness',' regularity' and „directionality'. Texture is characterized to have busy microstructures and uniform macrostructure.

Texture as defined by researchers:

All that result in the fact there is no precise definition of this notion. It is due to the difference even contradiction in the properties of the textures like regularity and randomness, uniformity and distortion, etc which are very difficult to describe uniformly.

Visually texture is recognized but very difficult to define. This difficulty is shown by the following amount of texture definitions given by the different researchers based on their own prospective and research needs.

- "A texture is a set of repetitive sub-patterns, which follow a set of well defined placement rules. These sub patterns themselves are made up of more fundamental units, called primitives. Such characterization of textures is generally applicable mainly to deterministic type of textures, such as line arrays, checker boards, hexagonal tiling, etc." [6]
- "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." [7]
- "An attribute representing the spatial arrangement of the grey levels of the pixels in a region." [8].
- "We may regard texture as what constitutes a macroscopic region. It structure is only contributed to the repetitive patterns in which elements or primitives are arranged according to a placement rule."[9].

- "Textures are the information obtained from a set of local measures (statistical, geometric, semantic,) in a region (visual window) of an image". [10]

Texture classes.

Periodic Textures:

Periodic textures are also called regular textures. In periodic textures the primitive patterns are repeated with some method of periodicity. Mostly manmade objects have these types of textures. Chess board is good example of periodic textures.

Non-periodic Textures:

Non-periodic textures are also called irregular or aperiodic textures. In non-periodic textures there is no periodic replication of primitive patterns. Natural textures often represent such textures like grass, bushes etc.

Semi-periodic Textures:

Semi-periodic textures will be later on referred as pseudo-periodic textures. In semi-periodic textures there is a variation in the regularity of the primitive pattern repetition like brick walls etc.

Periodic texture | *Pseudo periodic texture* | *Pseudo periodic texture* | *Pseudo or non periodic texture?* | *Non periodic texture*

Fig: 1.2: *Examples of periodic, pseudo periodic and non periodic Textures*

The Pseudo –periodic textures:

Pseudo-periodic textures are the textures which have variation in the regularity of its primitive's repetition. It is evident that pseudo-periodic textures are apparently periodic yet they are not. Brick walls, floor tiles, woven woolen crafts etc. Such textures are constituted of

some basic primitives, which are placed under some irregularity rules. By finding the primitives and their placement rules, the pseudo-periodic textures can be easily evaluated.

The periodicity of a texture is one of its important visual characteristics. Indicating how textural patterns are repeated in the texture, it has been often used as a measure for texture discrimination at the structural level. A regular texture such as wallpaper can be generated by tiling its motif, which means the minimal repeating region of the texture. Therefore, the motif gives a compact representation of the texture. Since the motif of a texture can be determined from its periodicity, knowledge of the periodicity is particularly useful for wallpaper design and compression [11].

Krumm and Schaffer [12] have worked on the finding method to derive shapes from textures using spectrogram. It was theoretically meant to be effective for both periodic and random textures. The method works best on textures that are closest to being purely periodic, like the cosine and canvas textures.

According to [13], it is time consuming to determine the textural periodicity using co-occurrence matrices. So they propose a distance matching function that can be driven directly from a texture. Given a displacement vector, the function value with respect to the vector is equivalent to the inertia of the corresponding co-occurrence matrix. Let us consider an image I of dimension $m \times n$, where m and n denote respectively its rows and columns numbers. Employing the correlation theorem, it is shown that this function can be evaluated at all positions over a texture in $O(mn\log mn)$ time, which constitutes a major improvement if we consider that computing inertias of all co-occurrence matrices requires $O(m^2n^2)$ time.

A set of statistical functions have been chosen to extract features from textures [14]. An algorithm has been developed that synthesizes images with the required statistical properties. The algorithm has been successful in synthesizing a large number of textures – both natural and synthetic.

The presence of periodic textures can facilitate wide-baseline matching by providing the periodicity distinguished regions (PDRs) that efficiently constrain the search for correspondence [15].

A periodic texture has contrast functions with deep and periodic minima.

In [16], Balas considers pseudo-periodic textures to be images based on a spatially regular repeated pattern, which may vary slightly across the image. (Text is considered as pseudo periodic because of the even spacing of rows.) To be more specific, pseudo periodic textures seem to rely relatively equally (and weakly) on both of these sets of parameters (magnitude correlations, coefficient correlation), given that the removal of each does not cause a large increase in the number of correct detections. It was found that for pseudo-periodic textures, only the removal of the first-order statistics produces a rate of oddball detection significantly greater than the „„full set'' images ($p < 0.05$).

Since we can observe texture in images (both of the natural and artificial) , and seems as the repetition of basic texture elements called texels or textons made of pixels whose placement obey some rule. In the case of periodic textures, there are essentially two methods to (1) adapt waveforms to the shape of each region or (2) perform an extrapolation and in this case solution (2) was opted for [17]. Since periodic textures are very appropriate for a spectral characterization, so the algorithm belongs to a family of descriptors that extrapolate the signal to fit a rectangular window.

Periodic or Regular textures are simply periodic patterns where the color/intensity and shape of all texture elements are repeating in equal intervals.

A limitation of the near-regular (pseudo periodic) texture synthesis algorithm is that the input texture sample must contain at least two complete tiles so that the underlying lattice and the tile set are well defined. The near-regular texture synthesis algorithm preserves the square pattern because it demonstrates stronger periodicity, but the hexagon net is discontinuous between squares.[18]

Yanxi Liu et al [19] have presented a computational model for periodic pattern perception based on the mathematical theory of crystallographic groups.

In order to extract a tile from a texture, its size, and consequently the period, must first be known. A variety of methods exist for determining the period of a near-periodic texture. In literature, the period is determined using co-occurrence matrices and other statistical tools such as the Chi-square ($\chi 2$) test, κ statistics and inertias. Certain others use an approach based on the Fourier transform, because if a texture is near-periodic, all of the energy in the power spectrum is concentrated in the frequencies corresponding to the periods. Concentrating on the rectangular tiles, it was argued that autocorrelation proves to be a very good method and

the one that is simplest to implement, and was opted for another similar method i.e. Sum of Squared Differences (SSD), for texture periodicity detection [20].

The SSD is a similarity measure used to verify whether two images are similar or significantly different. The following formula defines the SSD for two images I_1 and I_2, both of them possessing m rows and n columns:

$$SSD\ I_1, I_2 = \sum_{i}^{m} \sum_{j}^{n} (I_1\ i,j\ - I_2(i,j))^2$$

According to Djado et al [20], near-periodic textures are all around us, in brick walls, fabrics, mosaics, and many other manifestations. The basic motif can be extracted, yielding a small image known as a tile. The authors have given method for extracting a representative tile from a near-periodic texture, working from a photo. Beginning with determining the size of the tile using an SSD, they establish criteria that allow the selection of only the best tiles, insisting upon urging to stay possibly close to the original image.

There is a concept of fractional Fourier texture masks. Fourier texture masks (procedural textures for the regular parts) are derived by extracting the regular structure from a texture using fractional Fourier analysis and "enlarging" it to a desired size. These masks are used to guide sample-based synthesis algorithms in faithful selection and placement of copied pixels or patches, effectively enforcing global, regular structure and therefore leading to drastically improved synthesis quality for near-regular textures. A more appropriate selection criterion is based on two observations. First, since the regular structures in the image signal dominate the overall appearance of most near-regular textures, they carry a much larger amount of energy than the irregular patterns. Second, while the energy contained in strongly visible irregular structures is distributed among a large number of frequencies, the main part of the energy contained in the periodic structures is carried by a few frequencies [21].

Regular texture refers to periodic patterns that present non-trivial translation symmetry, with the possible addition of rotation, reflection and glide-reflection symmetries.

Near-regular texture is referring to textures that are not strictly symmetrical ([22]). The irregularity can be caused by various statistical departures from regular textures. It has focused on faithful texture synthesis of near-regular textures where departure from regularity is primarily caused by statistical color and intensity variations, while the underlying structural

regularity remains. There are many examples of this type of near-regular textures, e.g. brick walls, tiled floors, carpets, and woven sheets, where the texture patterns (each brick, tile, straw or bamboo strip) vary only locally.

Perfect regularities are rarely found in the real world, while varying degrees of deviation from regularity are common to observe.

Regular textures are simply periodic patterns where the color/intensity and shape of all texture elements are repeating in equal intervals. That is, a texture element is a unit tile in a regular texture, which can be synthesized by tiling the space with the unit tile. An example of regular textures is wallpaper. In the real-world, however, few textures are exactly regular. Most of the time, the textures we see in the real-world are near-regular, such as cloth, basket, windows, brick walls… Near-regular textures can be considered as departures of regular textures in different spaces with different degrees. For example, in brick wall textures, the major departure happens in the color/intensity space as the shape of each brick is regular but the color/intensity may vary.

Near regular textures are pervasive in both man-made and natural world. Even though textures are usually classified as either (structurally) regular or stochastic, most real-world textures fall somewhere in-between these two extremes. Such textures can be viewed as statistical departures of regular textures along different dimensions [23].

Homogeneity cues were used to detect the texel size in periodic, corrupted periodic and near-periodic textures [24].

By selecting appropriate parameters for explosive cladding makes it possible to obtain bimetallic semi-finished products which have a cross section with a periodic texture and strong bonds between the layers and are suited for subsequent hot rolling [25].

Towards the non-periodic textures:

Irregular textures refer to geometrically-irregular textures or textures with straightforwardly identifiable primitives (e.g. granite, sand). It is important to note that irregular textures (clouds, grass, spiral pattern etc,) are very difficult or even impossible to be represented by regularly arranged patterns.

Some proto-typical examples of irregular textures which are known as random macro-textures, are ceramic, marble and granite images, which are of particular concern in this

paper. As a matter of fact, the placement of the primitives within these is purely random and highly irregular [26].

Stam has worked on “a-periodic textures” [27]. He used a standard grid of square patches to tile the plane, and mapping 16 samples of different texture of a homogeneous texture. To achieve a non-periodic approach, he applied an algorithm for a-periodically tiling the plane with convex polygons of different colors (the tiles colors mark the boundaries of texture tiles).

Neyret et al [28] designed a method for covering the surface with an homogeneous non-periodic texture such as those that can be found on many natural objects.

Conclusion:

In this introduction have been presented the results of a bibliographical study we have made in order to propose a definition of the concept of texture. We can conclude that such a definition does not exist. Nevertheless, all the authors cited converge towards the same ideas concerning pseudo-periodic textures i.e. textures constituted initially on the basis of a pattern or tile, which is repeated with small variations of shape, orientation, lighting, scale and so on.

The main part of our work will be dedicated to such textures, and the manuscript will conclude on an attempt in direction of random textures.

Bibliography of chapter 1:

1. http://www.thefreedictionary.com/texture

   The American Heritage® Dictionary of the English Language, Fourth Edition published by Houghton Mifflin Company. Updated in 2009.

2. E.R. Davies "Introduction To Texture Analysis", Handbook Of Texture Analysis, Pp 1-31,2008.
3. A. Rosenfeld and A. Kak, Digital Picture Processing, vol. 1, Academic Press, 1982.
4. M. Levine, Vision in Man and Machine, McGraw-Hill, 1985.
5. H. J. Bunge, "Fifty Years Textures in Research and Practice," Textures and Microstructures, vol. 8, pp. 55-75, 1988.
6. Tinku Acharya, Ajoy K. Ray. Image processing: Principles and Applications, Published by John Wiley & Sons, Inc., Hoboken, New Jersey., 2005.
7. Sklansky, Jack. "Image segmentation and feature extraction." *IEEE Transactions on Systems, Man and Cybernetics, volume* 8 ,issue 4, pp. 237-247. 1978.
8. IEEE Standard 610.4- 1990, IEEE Standard Glossary of Image Processing and Pattern Recognition Terminology, IEEE Press, New York 1990.
9. Tamura, H., S. Mori, and Y Yamawaki , "Textural features corresponding to Visual Perception,", IEEE Transaction on Systems , Man and Cybernetics SMC-8, pp. 237-247, 1978.
10. Sébastien Mavromatis "Analyse de Texture et Visualisation Scientifique", Doctoral dissertation, Université de la Méditerranée, 2001.
11. Stevens, P. S. Handbook of Regular Patterns - An Introduction to Symmetry in Two Dimensions. The MIT Press, 168–331, 1991.
12. Krumm John, and Steven A. Shafer. "Shape from periodic texture using the spectrogram" *Proc. Int. Conf. Computer vision and Pattern Recognition*, pp.284 -289 1992
13. Gyuhwan Oh, Seungyong Lee, and Sung Yong Shin "Fast determination of textural periodicity using distance matching function" Pattern Recognition Letters, Volume 20, Issue 2, pp. 191-197, 1999.
14. G Bellala "Parametric Texture Model based on Joint Statistics" 2000

15. Dmitry Chetverikov and Jiri Matas “Periodic Textures as Distinguished Regions for Wide-Baseline Stereo Correspondence” In Proc. of the 2nd International Workshop on Texture Analysis and Synthesis, pp. 25-30, 2002
16. Benjamin J. Balas,”Texture synthesis and perception: Using computational models to study texture representations in the human visual system “ , Vision Research, 46, pp. 299-309, 2006.
17. Marc V Droogenbroeck, and Hugues Talbot. "Segmentation by adaptive prediction and region merging." *Digital Image Computing Techniques and Applications*. Vol. 2, 2003.
18. Lin, Wen-Chieh, James Hays, Chenyu Wu, Vivek Kwatra, and Yanxi Liu. *A comparison study of four texture synthesis algorithms on regular and near-regular textures*. , the Robotics Institute Carnegie Mellon University, 2004.
19. Liu, Yanxi, Robert T. Collins, and Yanghai Tsin. "A computational model for periodic pattern perception based on frieze and wallpaper groups." , IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 26, issue 3, pp.354-371, 2004 .
20. Khalid Djado , Richard Egli , François Deschênes,” Extraction of a representative tile from a near-periodic texture”, Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia, Dunedin, New Zealand November 29-December 02, 2005,
21. A. Nicoll and J. Meseth and G. Müller and R. Klein “Fractional Fourier Texture Masks: Guiding Near-Regular Texture Synthesis” Computer Graphics Forum. Vol. 24. No. 3. Blackwell Publishing, Inc., 2005.
22. Yanxi Liu , Yanghai Tsin , Wen-Chieh Lin, The Promise and Perils of Near-Regular Texture, International Journal of Computer Vision, volume .62 n.1-2, p.145-159, 2005.
23. Wen-Chieh Lin et al “Quantitative Evaluation on Near Regular Texture Synthesis” Computer Vision and Pattern Recognition Conference (CVPR '06), pp. 427 – 434,2006.
24. Lizarraga-Morales, R.A., Sanchez-Yanez, R.E., Ayala-Ramirez, V.: Homogeneity Cues for Texel Size Estimation of Periodic and Near-Periodic Textures. In: Martínez-Trinidad, J.F., Carrasco-Ochoa, J.A., Ben-Youssef Brants, C., Hancock, E.R. (eds.) MCPR 2011. LNCS, vol. 6718, pp. 220–229, 2011.
25. S. Sawicki Properties of bimetallic “carbon steel – stainless-steel” bars with periodic texture obtained by explosive cladding and rolling Metal Science and Heat Treatment,

Vol. 54, Nos. 5 – 6, September, 2012 (Russian Original Nos. 5 – 6, May – June, 2012).

26. J. Kittler, R. Marik, M. Mirmehdi, M. Petrou, J. Song, Detection of defects in colour texture surfaces. IAPR Proc. of Machine Vision Applications 94, pp. 558–567.
27. Jos Stam. Aperiodic texture mapping. Technical Report R046, European Research Consortium for Intbrmatics and Mathematics (ERCIM), January 1997
28. Fabrice Neyret , Marie-Paule Cani, Pattern-based texturing revisited, Proceedings of the 26th annual conference on Computer graphics and interactive techniques, p.235-242, July 1999

Résumé Chapitre 1 :

Dans ce chapitre, nous présentons le sujet de la thèse, consacré à l'étude des images texturées, qu'elles soient de nature pseudo-périodique ou aléatoire. Nous montrons la difficulté de donner une définition scientifiquement précise de la texture, bien que cette notion soit parfaitement perçue par le système visuel humain.

Chapter 2 – Classical tools for studying pseudo-periodic textured images.

2-1- Co-occurrence matrices

2-1-1- Definition, examples

2-1-2- Weakness of this approach and solutions to overcome this drawback

- Classification of the image (example : k-means)
- Haralick parameters

2-2- Haralick parameters

2-3- Covariograms

2-4- Conclusion: Why the LIP framework?

Chapter 2 – Classical tools for studying pseudo-periodic textured images

2-1- Co-occurrence matrices

2-1-1- Definition:

Co-occurrence matrix (or co-occurrence distribution) is a matrix which can be defined over (any) matrix as a distribution of values co-occurring at a certain offset. A very common practice of using such matrices is over images (over binary, grayscale or color images) which are two dimensional distributions of some numbers, and the values are actually the pixels intensities of the image. It can be given as

$$C\ i,j\ = \sum_{x=1}^{r}\sum_{y=1}^{c} \begin{cases} 1 & if\ I\ x,y\ = i\ and\ I\ x+\Delta x, y+\Delta y\ = j, \\ 0 & otherwise \end{cases} \quad \dots(2.1)$$

Co-occurrence matrices can describe a texture in terms of its statistics. A co-occurrence matrix can be viewed as a generalized/abstract form of a two dimensional histogram. The two axes of a 2 dimensional histogram of a two-channel image represent the pixel values in the first and the second channel. Each bin of this histogram represents the number of occurrences of a certain combination of values in a channel of the image. In co-occurrence matrices, the second channel is only limited to the representation of the second channel of the image, but can be some neighboring pixel for example. In a low contrast image the neighboring pixels are usually similar in contrary to the ones with high contrast, where plenty of neighboring pixels present quite different values. The co-occurrence matrix of an image, a chessboard for example, with the condition of having similar right side neighboring pixels will result in such a matrix , which is empty along diagonal and concentrated enough at left in top and right in bottom.

A co-occurrence matrix of an image can depict the distance and angular spatial relationship of an image sub-region of specific size, and that's why it is called *spatial dependence matrix*.

A co-occurrence matrix is defined as a two dimensional array C in which rows and columns represent some set of possible image values V**.** This V can be a set of possible gray tones for gray level images and a set of possible colours for colour images. The value $C(i,j)$

shows how many times value i co-occurs with value j in some specific predefined spatial order. This spatial order can be let's say that value i occurs immediately to the right of value j. Precisely we are looking for a specific case where set V represents the set of grey tones and the spatial order is defined by a translation ***t*** between the pixels having values ***i*** and ***j***. As at most of the cases we use co-occurrence matrices of the images having gray level intensities so we call them grey level co-occurrence matrices or shortly GLCMs.

For example, we can consider the translation t_r, t_c , where t_r represents row wise translation and t_c represents column wise translation. Let V be a set of gray tones. The C_t, a gray tone co-occurrence matrix, for image ***I*** can be defined as

$$C_t\ i,j\ =\ (r,c)\ I\ r,c\ = i\ and\ \ r+d_r, c+d_c\ = j \qquad \ldots(2.2)$$

Examples:

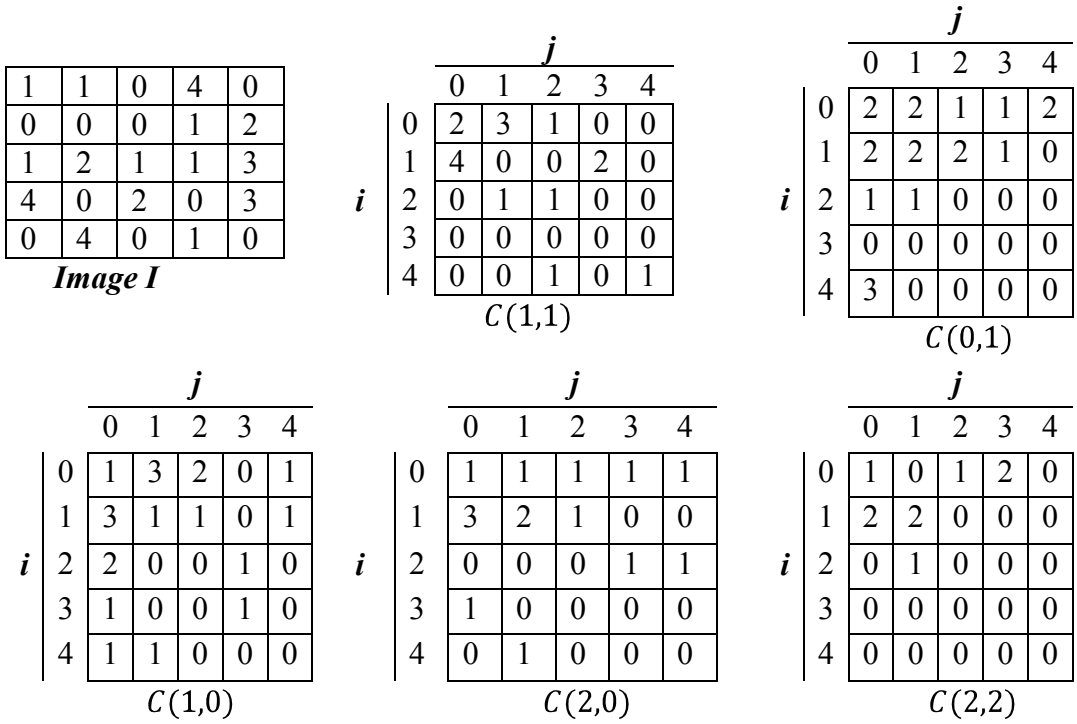


Figure 2.1: A 5x5 image matrix I and few co-occurrence matrices $C(1,1)$, $C(0,1)$, $1,0$, $C(2,0)$ *and* $C(2,2)$.

In $C\ 1,1$ at position (0,0) we have value 2, showing that $j = 0$ lies 2 times diagonally down (i.e. on the right side down in the adjacent row and adjacent column) to $i = 1$ in ***I***, while position (0,1) has values 3, showing that $j = 1$ occurs 3 times diagonally at right side of $i = 0$. The largest value i.e. 4 appears at (1, 0) indicating that $j = 0$ occurs 4 times to the

right down diagonal of $i = 0$. One thing very interesting is that the sum of entries in the co-occurrence matrix is always equal to total number of pixels compared. For example for $C(1,1)$ the number of pixels compared are 16 and we get the sum of entries in the co-occurrence matrices is 12. For $C\ 1,0$ and $C(0,1)$ the number of pixels are 20 and we get the sum of entries equal to 20 while for $C(2,2)$ it becomes 9.

i \ j	0	32	79	125	159	192	224	255
0	3	1	0	2	3	0	3	0
32	1	0	1	1	2	0	1	2
79	1	2	2	1	3	2	2	3
125	1	2	2	0	0	0	1	1
159	2	2	5	0	0	1	2	1
192	1	0	1	1	2	0	1	1
224	1	0	0	2	2	0	1	1
255	0	1	2	2	1	4	0	1

$C(1,1)$

Fig 2.2 b: the co-occurrence matrix C(1,1) of 2.2. a

0	159	255	159	79	79	0	32	192	159
32	125	79	192	192	192	79	159	125	224
0	79	79	224	255	0	159	125	0	224
255	224	255	32	159	125	224	32	79	32
255	255	0	159	255	79	0	159	224	0
0	192	79	0	32	79	255	159	79	224
125	125	159	159	0	0	32	32	0	192
192	255	32	224	79	159	224	159	159	0
79	79	192	255	125	79	224	255	79	255
224	159	224	125	125	32	255	125	192	159

Fig 2.2 a: we have a 10x10 image with 8 different gray levels ranging from 0 to 255.

Shapiro et al [1] has mentioned the following two important variations of the standard co-occurrence matrix.

a) Normalised co-occurrence matrix N_t given as

$$N_{t(i,j)} = \frac{C_t(i,j)}{\sum_i \sum_j C_t(i,j)} \quad \dots (2.3)$$

This normalizes the co-occurrence values to lie between zero and one which can be thought of as probabilities in the large matrix.

b) The second is symmetric grey tone co-occurrence matrix $S_t(i,j)$ that is

$$S_t(i,j) = C_t(i,j) + C_{-t}(i,j) \quad \dots (2.4)$$

But the problem with the co-occurrence matrix is that it neglects the image contents as the distance of co-occurrence enlarges. So it does not remain useful in that case.

Co-occurrence matrix of a gray level image is a square matrix. The size of the co-occurrence matrix is given by k^2, where k is the number of intensities in the image.

2-1-2- Weakness of this approach and solutions to overcome this drawback

Let us note that the method based on the computation of co-occurrence matrices is not so easy to perform:

1. The previous approach requires a lot of computation (many matrices to be computed)

2. Features are not invariant to rotation or scale changes in the texture.

3. It neglects the image contents as the distance of co-occurrence enlarges. So it does not remain useful in that case.

To overcome these shortcomings following two solutions are possible.

Classification of the image (example: k-means algorithm)

Classification like k-means is applied over image to regroup the image intensities into lesser number of intensities so that to remove the small local variations and decrease the co-occurrence matrix dimension. It is also called clustering.

K-means is simple unsupervised learning algorithm. It simply classifies the studied dataset into the given fixed 'k' number of clusters. The basic idea is to specify k-centroids $(C_1, C_2, \dots \dots C_k)$, i.e. a centroid for each cluster. These centroids should be selected carefully as the selection of different centroids produces different results. The better choice could be to choose them with sufficient gap in between them. In the next step, each point in the data set is associated to its closest centroid, by finding lowest distance, using any of Euclidean, Manhattan or Bray Circuit etc., between a point and a centroid. This step ends when there is no point unassociated, and the initial k classes are obtained. Then re-calculation of new k-centroids is needed for the classes obtained from the earlier step. After having calculated the new k-centroids $(C_1^1, C_2^1, C_3^1, , , , , C_k^1)$, a new binding takes place between the same data set points and the relative nearest new centroid. This step is to be repeated, n times for example, until the k centroids for the n^{th} iteration $(C_1^n, C_2^n, \dots \dots .., C_k^n)$ and those for $n-1$ th iteration $(C_1^{n-1}, C_2^{n-1}, \dots \dots, C_k^{n-1})$, i.e. the centroids do not move any more. It is very popular algorithm for unsupervised learning of neural network, pattern recognition, classification analysis, clustering analysis etc.

- Computation of Haralick parameters

Now let us explain another way to extract more information of initial co-occurrence matrices. This approach is due to Haralick and is presented in the following section

2-2- Haralick parameters

Robert M. Haralick [2] has given few parameters which are also called textural features, based on the co-occurrence matrices, given as follows. The aim of this technique is clearly to cumulate the few amount of information present in the co-occurrence matrix by means of summations.

a) <u>Angular Second Moment:</u>

As an example, consider the following formula:

$$f_a = \sum_i \sum_j \{p(i,j)\}^2 \quad \dots (2.5)$$

where for $p(i,j)$ is the i,j^{th} entry in a normalized gray-tone spatial dependence matrix = $P(i,j)/R$. This feature is also termed as Energy [3] or Total Energy [4]. It is the number of repeated pairs and will be high in case there are a lot of pixels pairs presenting the same intensity). Albregtsen calls it Homogeneity [5]. A scene is homogeneous if it has a little number of gray levels thus having gray level co-occurrence matrix with a little but high $p(i,j)$ values hence high sum of squares.

b) Contrast:

The second Haralick parameter is called "contrast" and is defined according to:

$$f_b = \sum_{n=0}^{N_g-1} n^2 \sum_{i=1}^{N_g} \sum_{\substack{j=1 \\ |i-j|=n}}^{N_g} p(i,j) \quad \dots (2.6)$$

where N_g represents the number of distinct gray levels in the quantized image. Karuni et al [3] have given the above formula as:

$$f_b = \sum_{i} \sum_{j} |i-j|^2 p(i,j) \dots (2.6\ a)$$

This parameter is also known as Inertia. It measures a local contrast in an image and will be low in case the gray levels are similar i.e. at the diagonal (top-left to right bottom) and will be high at the at the other corners.

c) Correlation:

Another parameter, the "correlation", is given by the following formula:

$$f_c = \frac{\sum_i \sum_j ij\ p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y} \dots (2.7)$$

where μ_x , μ_y,are the means and σ_x , σ_y are the standard deviations of p_x and p_y. The $p_x(i)$ is the i^{th} entry in the marginal-probability matrix obtained by summing the rows of $p(i,j)$ i.e. $p_x(i) = \sum_{j=1}^{N_g} P(i,j)$ and for $p_y\ j = \sum_{i=1}^{N_g} P\ i,j$.

So $\mu_x = \sum_{i=1}^{N_g} i \sum_{j=1}^{N_g} P(i,j) = \sum_{i=1}^{N_g} i p_x(i)$,

$\mu_y = \sum_{j=1}^{N_g} p_y(j)$,

$\sigma_x^2 = \sum_{i=1}^{N_g} (i-\mu_x)^2 \sum_{j=1}^{N_g} p(i,j) = \sum_{i,j}^{N_g} (i-\mu_x)^2 p(i,j)$,

$\sigma_y^2 = \sum_{i,j}^{N_g} (j-\mu_y)^2 p(i,j)$.

It measures the linear dependence of the gray levels between the pixels in some relative specified positions. Or correlation is high in case the gray level pixel pairs are strongly dependent.

d) Sum of squares or variance:

This parameter is:

$$f_d = \sum_i \sum_j (i-\mu)^2 p(i,j) \quad \ldots (2.8)$$

This is given by [3] as follows,

$$f_d = \frac{1}{2} \sum_{i}^{M} \sum_{j}^{N} (i-\mu_x)^2 + (j-\mu_y)^2 \; p(i,j) \quad \ldots (2.8\,a)$$

Variance is high valued if the gray levels in the image are spread out. It gives high weight to those $p(i,j)$ for which (i,j) values differ from their mean.

e) Inverse difference Moment:

This one is:

$$f_e = \sum_i \sum_j \frac{1}{1+(i-j)^k} p(i,j) , \qquad i \neq j \quad \ldots (2.9)$$

We observe that k=2 in [2] and [4].

This is also termed as local homogeneity [4], [5]. IDM is high when the image is locally homogeneous and will be low if there is less or no local homogeneity.

f) Sum Average, defined according to:

$$f_f = \sum_{i=2}^{2N_g} i p_{x+y}(i) \quad \dots (2.10)$$

where

$$p_{x+y}(k) = \sum_{\substack{i=1 \\ i+j=k}}^{N_g} \sum_{j=1}^{N_g} p(i,j) \text{ for } k = 2,3,4,5,\dots,2N_g. \quad \dots (2.10\ a)$$

It is determined by the homogeneity of darkness or brightness of the image. The more homogeneously bright the image is the high is the Sum Average and vice versa.

g) Sum Variance:

This one is classically defined by:

$$f_g = \sum_{i=2}^{2N_g} (i - f_f)^2 p_{x+y}(i) \quad \dots (2.11)$$

Sum Variance usually has high values when the values of the co-occurrence matrix are equally concentrated in the lowest and highest cells.

h) Sum Entropy:

Entropy is computed on the basis of events probabilities multiplied by their own logarithm. Here it gives:

$$f_h = -\sum_{i=2}^{2N_g} p_{x+y}(i) + \varepsilon \log p_{x+y}(i) + \varepsilon \quad \dots (2.12)$$

It is possible that i can be zero, and thus undefined value for $\log(0)$, so an arbitrary number ε is used to avoid such situation.

i) Entropy:

$$f_i = -\sum_i \sum_j p(i,j)\log(p(i,j)) \quad \ldots(2.13)$$

Since it is the measure of disorder of the image so the images with homogeneous scenes will have low entropy and the opposite ones have high.

j) Difference Variance:

$$f_j = variance\ of\ p_{x-y} \quad \ldots(2.14)$$

k) Difference Entropy:

$$f_k = -\sum_{i=0}^{N_{g-1}} p_{x-y}\ (i+\varepsilon)\ \log\ p_{x-y}\ (i+\varepsilon) \quad \ldots(2.15)$$

where

$$p_{x-y}\ (k) = \sum_{\substack{i=1 \\ i-j\ =k}}^{N_g} \sum_{j=1}^{N_g} p(i,j), for\ k = 0,1,2,\ldots,N_g - 1 \quad \ldots(2.15\ a)$$

l) Information measure of correlation –I:

$$f_l = \frac{f_i - H_{xy}}{\max\{H_x, H_y\}} \quad \ldots(2.16)$$

where H_x and H_y are entropies of $p_x$and p_y.

$$H_{xy} = -\sum_i \sum_j p(i,j) log\ p_x\ (i)\ p_y(j) \quad \ldots(2.17)$$

m) Information measure of correlation-II:

$$f_m = (1 - \exp[-2(I_{xy} - H_{xy})])^{1/2} \quad \dots (2.18)$$

where $I_{xy} = -\ _i\ _j p_x\ i\ p_y\ j\ log\ p_x\ i\ p_y(j)$ $\dots (2.18\ a)$

n) Maximal Correlation coefficient:

$$f_n = (Second\ Largest\ eigenvalue\ of\ Q)^{1/2} \quad \dots (2.19)$$

where

$$Q\ i,j\ = \ _k \frac{p\ i,k\ p\ j,k}{p_x\ i\ p_y\ k} \quad \dots (2.19\ a)$$

Comments about Haralick parameters:

A lot of scientific papers have been dedicated to Haralick parameters which have demonstrated their ability to classify textures. Such works are well-known by the community of searchers interested in texture estimation and textures images classification, so our aim is not here to deeply discuss their interest and application field.

Nevertheless, at the very beginning of this thesis work, we planned to adapt Haralick parameters in the context of the LIP (Logarithmic Image Processing) Model. In fact, this Model has been demonstrated as consistent with Human Vision, and it is an evidence to consider the concept of texture as strongly linked with the Human Visual system. We spent a significant time to define Logarithmic Haralick parameters, but we have met a real obstacle due to the necessity to compute grey levels products. We have failed in defining a grey level multiplication with a real meaning in the field of LIP. This is the reason why we decided to orient our investigation towards *Covariograms*, which constitute a neighboring notion of co-occurrence matrices and are exploitable thanks to *Metrics*, which are easily defined in the LIP framework.

2-3- Covariograms

The concept of covariogram gives the spatial description of a regionalized variable $z\ x$, where $D \subset \mathrm{R}^d$ and generally d= 2 or 3, $z\ x$ in the form of distance vector h given as:

$$g\ h\ = \ z\ x\ z\ x + h\ dx \quad \dots (2.20)$$

The integral covers the whole geographical area, or its equivalent, over the range D where x can be a point in $2D$ space, any pixel in an image for example, and $z(x)$ can be then given as pixel density. It is termed as transitive covariogram in [6] and [7].

It is zero beyond some definite distance i.e. the range. At short distances its behavior, like derivability and connectivity, depends on the spatial regularity of the regionalized variable. The integral range of the covariogram is given by $Q^2/g(0)$ [6].

Covariogram can be considered as curve $C(h)$, representing the probability that a point lying in a phase remains in the same phase when translated according to a distance h in a given direction:

$$C\ h\ = Pr\ x \in X, x + h \in X \qquad \ldots (2.21)$$

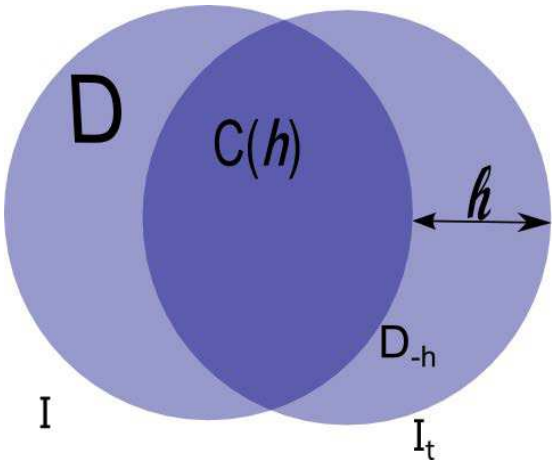


Fig 2.3: Image I translated by distance ‚h' giving I_h. C(h) is the overlapped area of I and I_h

In the following binary image I, the points which satisfy these conditions are the common points between the image I and the translated image I_h with some distance h. In practice, $C\ h$ is thus measured as the intersection of the two images (cf. Fig 2-3).

We also talk about covariance function, let the phase function is denoted by u such that:

$$u\ x\ = \begin{matrix} 1 & for\ a\ point\ inside\ the\ phase \\ 0 & otherwise \end{matrix} \qquad \ldots (2.22)$$

It can also be written as following

$$C\ h\ = \lim_{\alpha\to\infty} \frac{\int_0^{\alpha} u\ x\ u(x+h)dx}{\int_0^{\alpha} u^2 dx} \qquad \dots (2.22\ a)$$

In some books covariance is referred as the correlation of two points.

It is important to note that $C\ h$ have significant properties:

- $C\ 0\ =$ Probability that a point be in considered phase. It can be also said that it is not null in a limited field because it is cancelled outside D.
- The Cauchy-Schwarz inequality holds : $\forall\ h \neq 0,\ g(h)\ \leq g\ 0$.
- Symmetry holds and is given as $\forall\ h \in R^d,\ g(h) = g(-h)$.
- The slope at the origin is related to the number of intercepts and thus at specific perimeter:

$$C'\ 0\ = -N_L = \frac{-L_A}{\pi} \qquad \dots (2.23)$$

This property can be particularly used to correct the effect of a very tortuous outline, by calculating this slope by a mean over a finite distance; this is equivalent to the smoothening the contour of the phase by the image.

- The slope $C\ h$ reproduces the periodicity of the image if there is any; more generally due to its oscillations the possible correlation can be detected.
- If the image is not periodic the correlation fades away at large distance; i.e. that the points are no more correlated if h is large enough and $C(h)$ tends to a constant C_∞ (probability that two mutually interdependent points should be in the same phase = product of the probabilities). This can be represented by a slope with an asymptote for this value. The distance beyond which the asymptote is reached indicates the correlations scale (scope of the correlation).
- The feature scale of correlation, the integral range, is given as:

$$a = \frac{2}{C\ 0\ - C_\infty} \int_0^{\infty} C\ h\ - C_\infty\ dh \qquad \dots (2.24)$$

The covariogram can be used to choose the appropriate magnification for images: if the asymptote is not affected by an image, it means that the points of an image are (partially)

inter-correlated and that the observation range is not sufficient. Observations could therefore be quite different at another point of the image, and it should be depending on whatever we want to measure like increase the field or multiply the observation i.e. measure of the range.

Another term called relative covariogram has been given in [6] as:

$$z'(x) = g(h)/Q^2 \qquad \dots (2.25)$$

$$where\ Q = \overline{\qquad\qquad} g(h)dh \qquad \dots (2.25\ a)$$

The inverse of equivalent surface is given by $z'(x)$ at x=0.

The relative covariogram represents individuals of a population. Consider two independent individuals selected randomly. The distance between the said individuals is a vectorial random variable and the relative covariogram is given by its probability density function [7].

2-3-1-Geometric Covariogram:

A special case of transitive covariogram is geometric covariogram noted $K(h)$ and defined according to:

$$K\ h\ =\ I_D\ x\ I_D\ x+h\ dx = D \cap D_{-h} \quad where\ I_D\ x\ = \begin{matrix} 1 & for\ x\ \in D \\ 0 & otherwise \end{matrix} \qquad \dots (2.26)$$

where $K\ h$ represents the measure (length, area, volume, etc.) of the $D \cap D_{-h}$, where D_{-h} is the translation of D by the vector $-h$. In particular, $K(0)$ is the measure of D, implying the relationship given as

$$K\ 0\ ^2 = \left(I_D\ x\ dx \right)^2 = K(h)dh \qquad \dots (2.26\ \ a)$$

The geometric covariogram is a measurement of the area of intersection of the set S with its translation by h [8].

The geometric covariogram describes the form of the field D, so it always has the linear behavior at the origin. Under certain conditions, the derivative at the origin in all directions can be related to the perimeter of D (in 2-dimensional case). Even there exists the derivative at the origin in all directions yet the geometric covariogram is not differentiable at $h = 0$ [9].

It is stated in [10] that $K\ h\ = K\ -h$, and we have the following definition of geometric covariogram:

$$K\ h\ =\ D \cap D_{-h}\ , h \in R^d \qquad \ldots (2.26\ \ b)$$

2-3-2- Regionalized variable:

Regionalized variable is a single valued function defined over a metric space [8].

Or a regionalized variable is a variable that can be considered as to be *distributed* in space. This space is not limited to the three-dimensional kind of space that we move around in every day, but can be extended to include time, parameter space, property space, *etc.* This definition "distributed in space" is purely descriptive and makes no probabilistic assumptions.

A phenomenon, when spreads in a space and exhibits a certain spatial structure, is termed as regionalized. If f(x) denotes a value at a point x of a characteristics f of this phenomenon, we can say that the f(x) is regionalized variable [7].

This is a neutral and purely descriptive.

From the mathematical point of view, a regionalized variable is then simply a function f(x) of point x but is, generally, a very irregular function.

It shows two complementary characteristics:

a) A random characteristic:

It may show irregularity and unpredicted variations from one point to another. In other words, an observation of a variable at a point x_i, within the larger study area D, is a realization of a random variable $z(x_i)$ at the point x_i [12].

b) A structured characteristic

To some extent it must reflect the structural characteristics of the regularized phenomenon. It can be thought of a nonrandom component in which the random variables for the two locations $z(x_i)$ and $z(x_i + h)$ i.e. separated by a distance h, are not considered spatially independent [12].

The field of the regionalized variable is the region where it differs from zero. [7].

A regionalized phenomenon cannot be infinitely large, thus we study the regionalized variable only inside a limited domain D, the range of the variable. This range D can represent a natural zone outside which the z is undefined. This can be some particular domain where the regionalization has an interest, the points where it is not null or superior than the detection limits [9]. Regionalized variable is statistically homogenous throughout the surface [12].

One technique used to design an optimal sampling network for a regionalized variable, such as air pollution, is sequential sampling. Sequential sampling is based on extended knowledge of the area to be sampled and expertise in the factors controlling the distribution of a regionalized variable [11].

2-3-3-Our Approach of Covariogram:

Covariogram is the graph of similarity metric (correlations, distances) obtained by the superposition of an image A over an image B (B can be the replica of A) and then translating factor of one of the images in some direction. In short, we plot the translations at x-axis and the summation of similarity measure (using some similarity metric) for the superposed area of the two images on y-axis and term it as covariogram.

The distance is zero at the initial point of the superposition and it becomes minimal as the repeating pattern arrives in the image and thus valleys as well as lowest points of valleys are obtained. The repetitions of these valleys and those of lowest points show that at there is reappearance of some structure. These repeated structures are identified as texture according to its definition.

In case of correlations there is inverse relation. Initially when the two images are superposed, the correlation is high and as we translate the superposed images in some direction, the correlation begins to lower. In this case we get some peaks when there is

similarity between the superposed portions of the two images. It can be minimal or even zero when there is no similarity in the superposed images.

Let us assume that we have a binary image A of size 8×8, with white and black squares representing the intensity of pixels as zeros and ones in Fig 2.3 a. We have its duplicate image A'. In Fig 2.3 b the images are superposed in a way that they look like another image B with darker black squares. In this case, if we calculate the sum „S „ of differences of the superposed pixels, will be zero i.e. the following equation will result in zero as all the superposed pixels are similar.

$$S = \quad A_{x,y}A'_{x,y} \; where \; A_{x,y}A'_{x,y} = \begin{matrix} 1 \; for \; A_{x,y} \neq A'_{x,y} \\ 0 \; for \; A_{x,y} = A'_{x,y} \end{matrix} \qquad \ldots (2.27)$$

If from equation 2.27 we get $S = 0$, it shows that the both the images are exactly similar. Now we exhibit the superposition of image A' over image A by translating it in some directions by some step size here for example 1, i.e. each time we translate it by one pixel in some direction, until there is no superposition and for each step we calculate the sum for the pixel intensities of the superposed parts according to the equation 2.27.

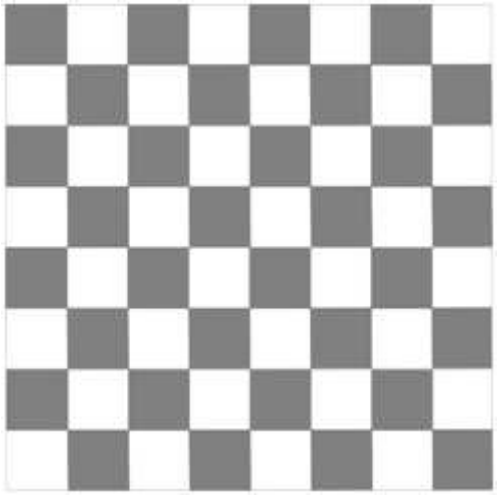


Fig 2.3 a)

Fig 2.3 b)

Fig.2.3: a) The binary image A of size 8x8 and b)Image B resultant of image A superposed by its duplicate Image A′.

Diagonal translation:

In the below Figure 2.4a, we see the image B, ie image A superposed by image A'. In Figure 2.4 b we see the image A superposed by a translated image A'. In this case first column and first row of image A are not superposed and similarly 8^{th} row and 8^{th} column of A' also

does not cover any part of image A . By using equation 2.27 to calculate the summation of the differences we get zero at this point. Similarly in Figure 2.4 c we have 6×6 pixels of image A superposed by those of the same number of image A'. But here also if we use equation 2.27 the result is zero. It is because the pixels are in the exactly the same pattern and diagonally while translation, the black pixels always cover black pixels and white pixels cover white ones.

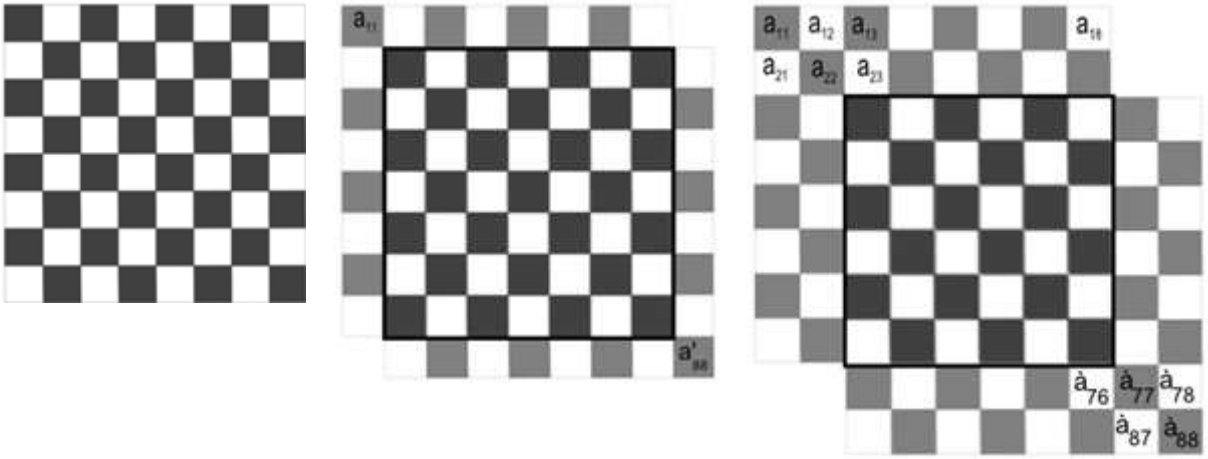


Figure 2.4 a *Figure 2.4 b* *Figure 2.4 c*

Fig 2.4 a. image B resulting from image A superposed by its duplicate Image A′, Fig 2.4 b is the first step(1,1) translation and Fig 2.4 c shows the 2nd step (1,1) translation.

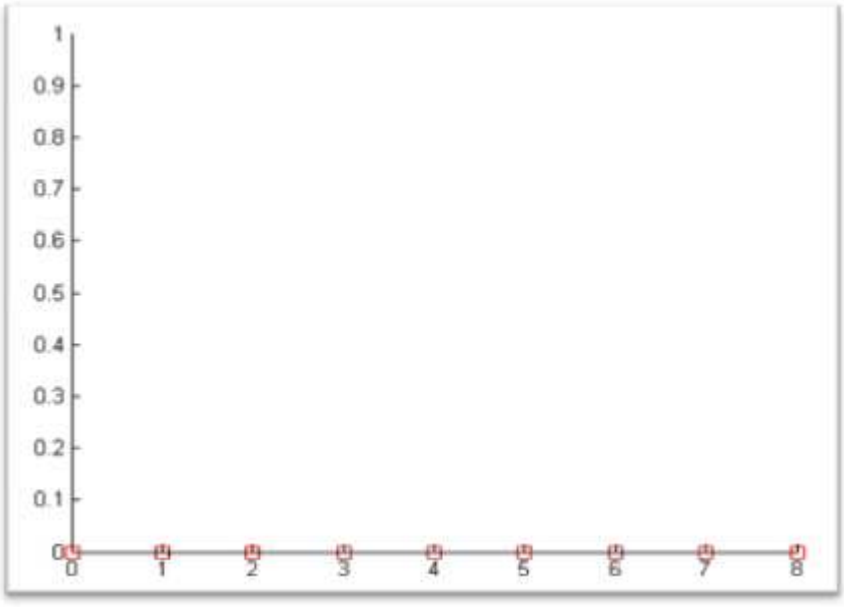


Fig 2.5: Covariogram of the image in Figure 2.4 by diagonal translations or (1,1) translation and using the distance metric as given in equation 2.27.

If we plot such the translations and the respective results of the equation 2.27 , we get the plot as in Fig. 2.5 and this is covariogram. In this figure, the translation steps are plotted on the x-axis and the resulting values from equation 2.27 for each translation step are plotted on y-axis. In this covariogram, for each translation step the value given by the equation 2.27 is zero and thus the values of y-axis do not gove above zero at any translation step.

Further more this covariogram shows that the image in figure 2.4 a is higly periodic and that's why there is no value of equation 2.27 which goes above zero. Hence the aforementioned equation calculates the distance values between the two images and in this case all the resulting values (for image A and its duplicate image A') are zero showing that there is a perfact repetition of some patterns in the image, and from this we assume that this image is a textured image.

Vertical Translation:

In Fig 2.6 a), image A is superposed by $'$, giving image B as mentioned above and the result of apply equation 2.27 is known to be zero. Now by translation one row down wards we get the Fig 2.6 b), where first row of image A and last row of image A' remain uncovered.

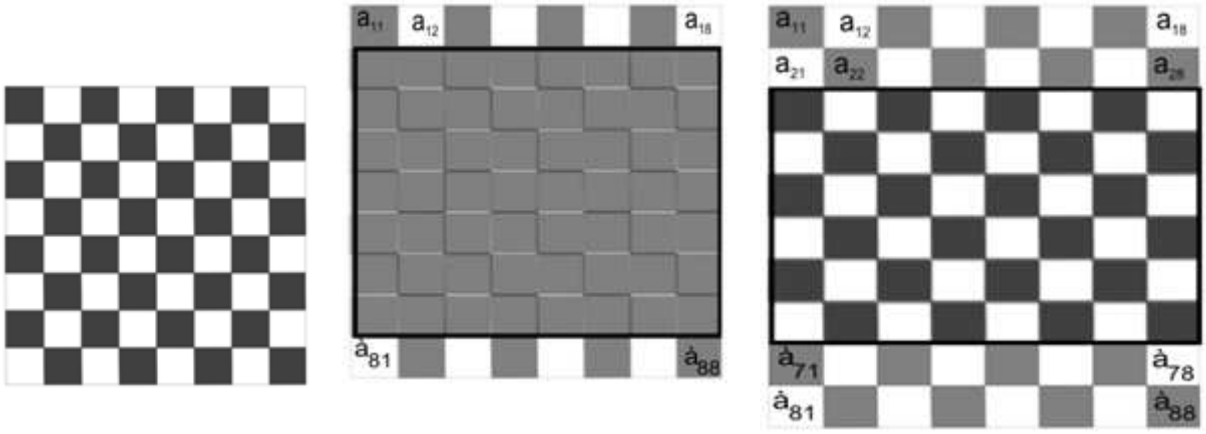


Figure 2.6 a. *Figure 2.6 b* *Figure 2.6 c*

Fig 2.6 a. image B resulting from image A superposed by its duplicate Image A', Fig 2.6 b. is the first step(1,0) translation and Fig 2.6 c. shows the 2^{nd} step (1,0) translation

But the last seven rows of Image A are superposed by the first 7 rows of image A' and thus we get the result of equation 2.27 as S=56. For Fig 2.6 c), the last 6 rows of image A are superposed by the first 6 rows of image A' , but since the white pixels cover white pixels and the black pixels cover black ones so the result of equation 2.27 is 0. For the $3rd$ step of translation it would 5 rows superposed and the value of S=40 in the equation 3.32. By

computing and plotting all these values, we get the following plot (Fig. 2.7). In this case the peaks show how much are the superposed pixels different from one another. On the other hand the valleys which are exactly zero shows that at these steps of translations the superposed portions are exactly similar.

And the repetition of this feature shows that there are some repeating patterns which we term as texture.

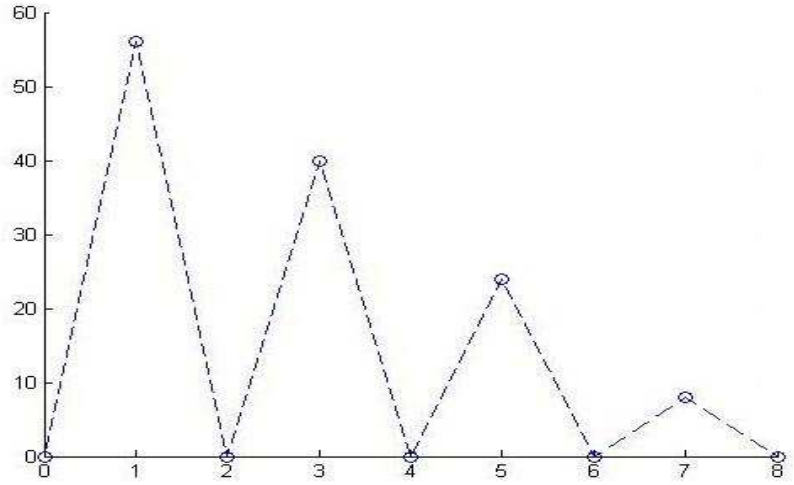


Fig. 2.7 : Covariogram of the image in Figure 2.6 by vertical translations or (1,0) translation and using the distance metric as given in equation 2.27.

Horizontal translation:

Covariogram is obtained by the column-wise translation of image A of Fig $2.3.a$ superposed by image A' of Fig $2.4.b$, and calculating the resultant distance, at each translation step, using equation 2.27. Here in $Fig\ 2.8.a$, is the image B , resulting from superposing image A by image A' at translation step '0', has the value of distance metric sum S=0 for equation 2.27 . In $Fig\ 2.8.b$, at the first step of column-wise translation, the first column of image A and the last column of image A' are out of the superposition but the last 7 columns of image A and the first 7 columns of image A' are mutually superposed resulting in S=56. In the second step the first two columns of image A and last two columns of image A' are out of the superposition while the superposed part gives S=0 because the corresponding black and white pixels superpose by relative black and white pixels.

Figure 2.8 a.

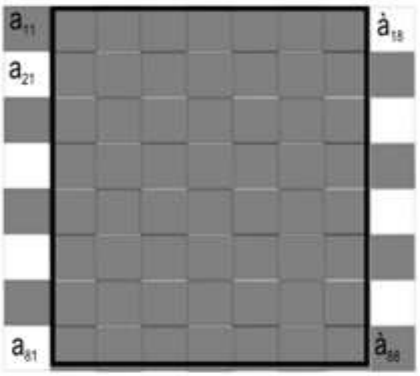


Figure 2.8 b

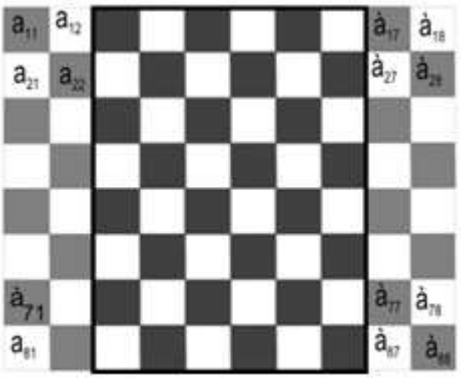


Figure 2.8 c

Fig 2.8 a. image B resulting from image A of Fig 2.4a superposed by its duplicate Image A', Fig 2.8 b. is the first step(0,1) translation and Fig 2.8 c. shows the 2^{nd} step (0,1) translation.

In the following steps of translations we get the results which we plot as follows (Fig.2.9).

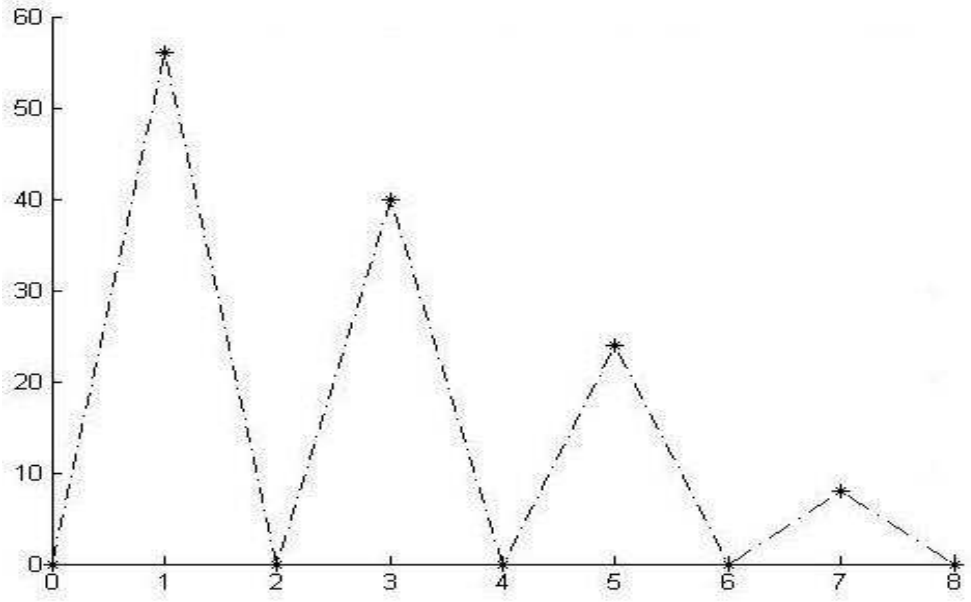


Figure 2.9: Covariogram of the image in Figure 2.8 by horizontal translations or (0,1) translation and using the distance metric as given in equation 2.27.

In this figure at initial step '0' we have distance measure S=0, showing that both the image *A* and image A' are totally identical. It produces image *B* as in $2.8.a$ in which black part square cover black squares and white squares cover white ones. But when the column-wise translation is made at the first step, the black and white squares superpose the squares of opposite colors which will result in high distances sum and that is why there is a peak at the first translation step shown in $Fig2.9$ at 1. It is repeated for 3, 5, etc for x-axis, where y-axis have high values. Similarly at 2, 4, etc, of the x-axis we get '0' for y-axis as the distance sum

S of equation 2.27 is zero, and it produce valleys. This repetition of valleys and peaks shows that there is a repeating pattern in the image which termed as texture.

Correlation:

Correlation is the parallel, reciprocal or complementary relationship (structural, functional etc) between two or more comparable objects. It permits the comparison between/among two/more things (images) based on some characteristics e.g. similarity etc.

Correlation has two variants i.e. autocorrelation and cross-correlation. Autocorrelation is applied on an image to find the correlating segments inside it while cross-correlation is used for finding such feature between two images.

While applying this to evaluate textures in images to construct covariogram we will try to find that how much are the overlapped portions correlating to one another at each translation step. We have the following formula:

$$C\ A, A' \ = \ A_{x,y} \times A'_{x,y}\ where\ A_{x,y} \times A'_{x,y} = \begin{matrix} 1\ for\ A_{x,y} = A'_{x,y} \\ 0\ for\ A_{x,y} \neq A'_{x,y} \end{matrix} \quad \ldots (2.28)$$

For each translation we note the correlation value and then plot these values against the translations steps. The values of peaks here represent high similarities while valleys show the less similarity or no similarity if it is zero. Each value of image A will be multiplied (compared in some cases) with the overlapping image A' values and thus all resultant values when summed up will be the correlation for one translation step. This will be repeated until the end of the translation. Since we have binary images here so one exemption will be made, that in both cases, when white pixel overlaps white one (1 overlaps 1) or when black over black (0 overlaps 0), we note 1 and sum these ones to get the correlation. So, image A is 8×8 and same is image A', thus using this exception we get the value $C\ A, A' \ = 64$, for zero translation. Following are the diagrams of stepwise translations in diagonal, horizontal and vertical directions with the relative covariograms. It should be noted that in all the three cases the initial value of equation $C\ A, A'$ is same and is the highest in this case i.e. 64. Later values of the equation 2.33 and the covariogram for the diagonal translation is different from the later two , notably in this case the later two remain the same in representation.

Correlation for Diagonal translation:

The initial correlation value as discussed above is 64. Then at the first diagonal translation ,as in Figure 2.10 b, the first 7 columns and first 7 rows of the image A' overlaps the corresponding last 7 last columns and 7 last rows of the image A, results in correlation value $C\ A, A'\ = 49$. In the step of translation the first 6 columns and first 6 rows of the image

A' overlaps the corresponding last 6 last columns and 6 last rows of the image A, as shown in Figure 2.10 c resulting in the correlation value $C\ A, A'\ = 36$. In the third translation step the correlation value $C\ A, A'\ = 25$, and so on. The covariogram is obtained by plotting these translation steps on x-axis and the corresponding correlation values on y-axis which is given in Fig 2.11. Here we see that the correlation value at translation step '0' is highest i.e. 64. The correlation value decreases with the following translation steps because the number of overlapping pixels decreases and finally it becomes zero when there is no overlap.

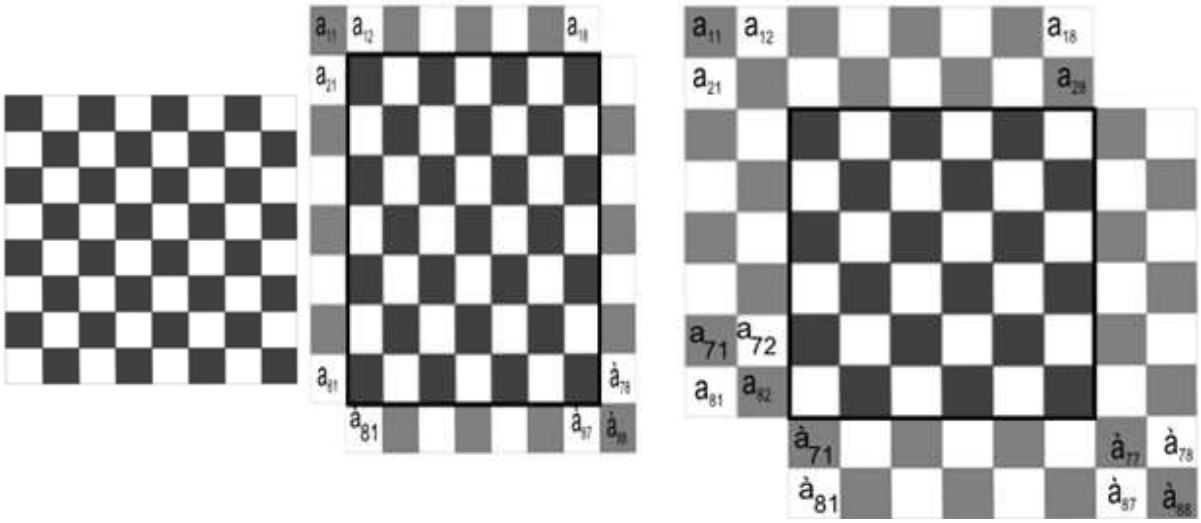


Figure 2.10.a *Figure 2.10.b* *Figure 2.10.c*

Figure 2.10 :a)shows the correlation between the superposed images A and A' at translation step '0', b) shows the correlating pixels at diagonal translation step '1', and c) shows the correlation at diagonal translation step '2'.

Since this translation is diagonal so there and this binary image has truly periodic occurrences of black and white pixels, each pixel is represented by a square of its color, the values on y-axis of the covariogram decreases in a gradual manner according to n^2 where $n = 8, 7, 6, \ldots, 0$.

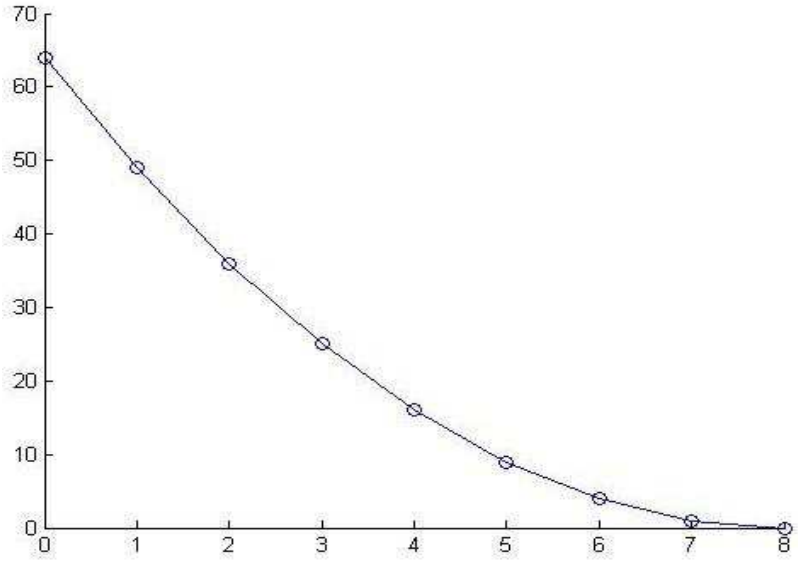


Fig. 2.11: Covariogram based on correlation of two images, which are translated diagonally.

This figure shows that at each translation there is complete correlation. It also shows at each translation step the existence of periodicity as there is high correlation value.

Correlation for Horizontal translation:

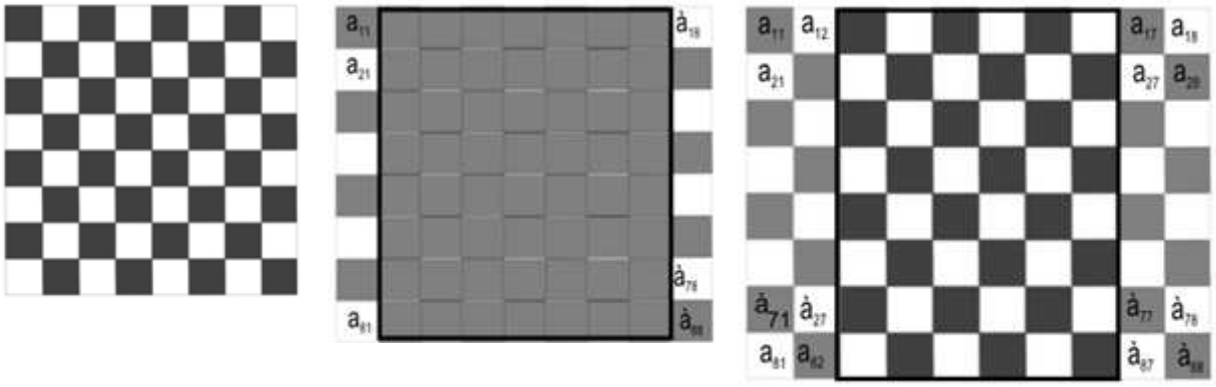


Fig. 2.12 a) *Fig. 2.12 b)* *Fig. 2.12 c)*

Fig. 2.12: a) has the identic description as that of Figure 2.10a. b) shows the correlating pixels at horizontal translation step '1', and c) shows the correlation at horizontal translation step ‚2'.

Correlation between the two superposed images, A and $'$, is observed at each translation step while translating image A' along the x-axis. In Figure 2.12a we have image B of the Figure 2.4b where image A is superposed by A' at the translation step '0' and has the correlation value $C\ A, A' = 64$, as already explained. At first translation step, the last 7 column of the image A are overlapped by first 7 columns of the image A', resulting in the

correlation value for equation 2.28 is $C\ A, A' = 0$. In the succeeding translation steps we get the values for $C\ A, A'$ are 48, 0, 32, and so on. When plotted the values of correlation on y-axis against translation steps on x-axis, we get the following covariogram (Fig. 2.13).

In this covariogram we observe that there are peaks and valleys. The peaks show the high correlation and the valleys show the low correlation. In this case no correlation as y-axis approaches zero at certain translation steps. This phenomenon is repeated showing that there is some repeating pattern which can be termed as textures.

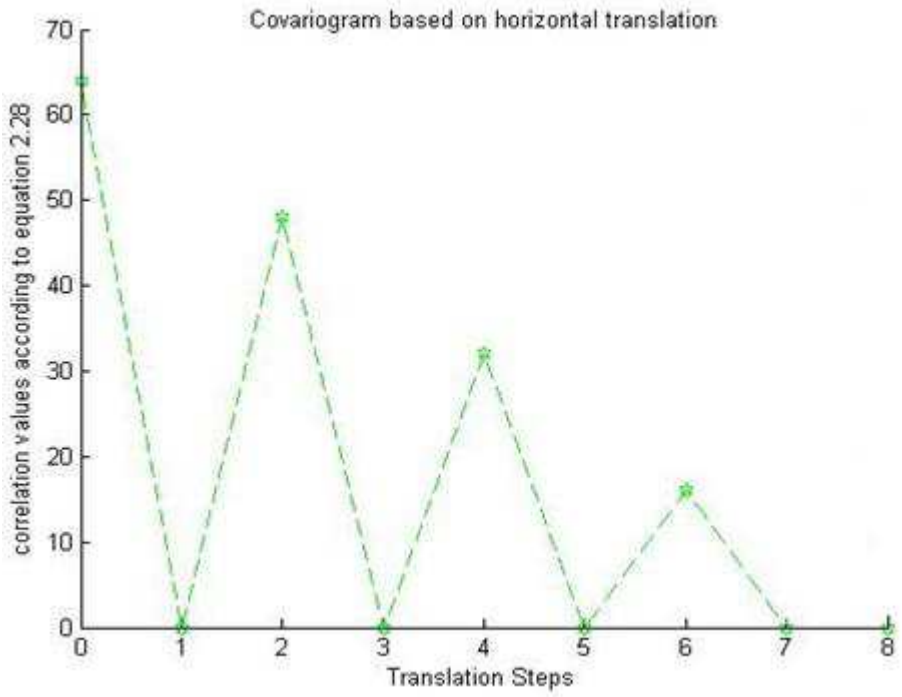


Fig. 2.13: Covariogram based on correlation of two images, which are translated horizontally.

Correlation for Vertical translation:

The initial step in this case is identic to the previous one i.e. the image A is superposed by image A' and the value of equation 2.28 at the translation step '0' is $C\ A, A' = 64$ as shown in the Fig 2.14a. By translating image A' vertically downwards the first 7 rows of image A' overlap the last 7 rows of A as in Fig 2.14b .But here the white and black squares representing the pixels of the relevant colors are superposed by those of the opposite colors so the values of correlation according to equation 2.28 is $C\ A, A' = 0$. Fig 2.14c shows that in translation step '2', the last 6 rows of image A' and the first 6 rows of image A are overlapped by each other producing the correlation values $C\ A, A' = 48$ according to equation 2.28. In

the forthcoming translation steps the values of the equation 2.28 are 0,32,0,16, and so on respectively.

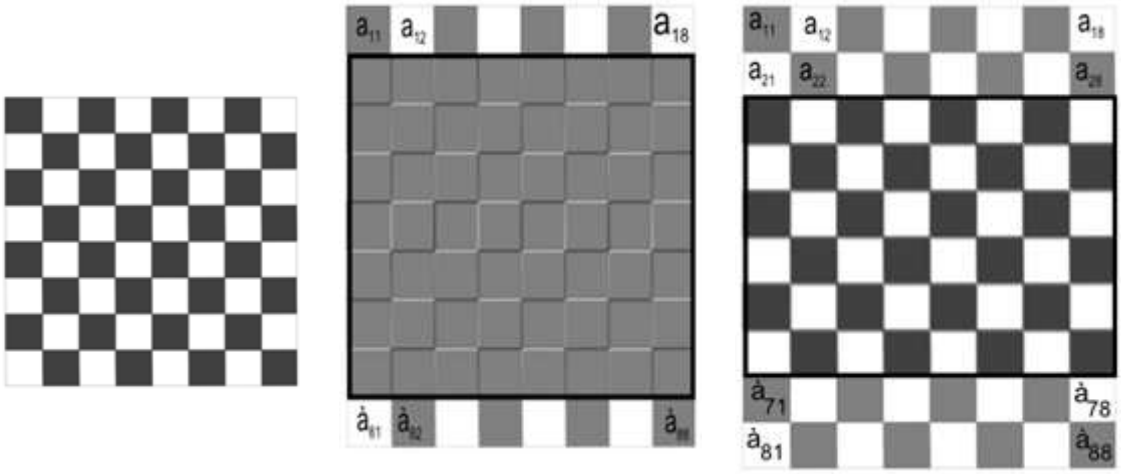


Figure. 2.14 a *Figure 2.14 b* *Figure 2.14 c*

Fig. 2.14 a) has the identic description as that of Figure 2.10a. b) shows the correlating pixels at vertical translation step '1', and c) shows the correlation at vertical translation step '2'.

The covariogram for this is given as below:

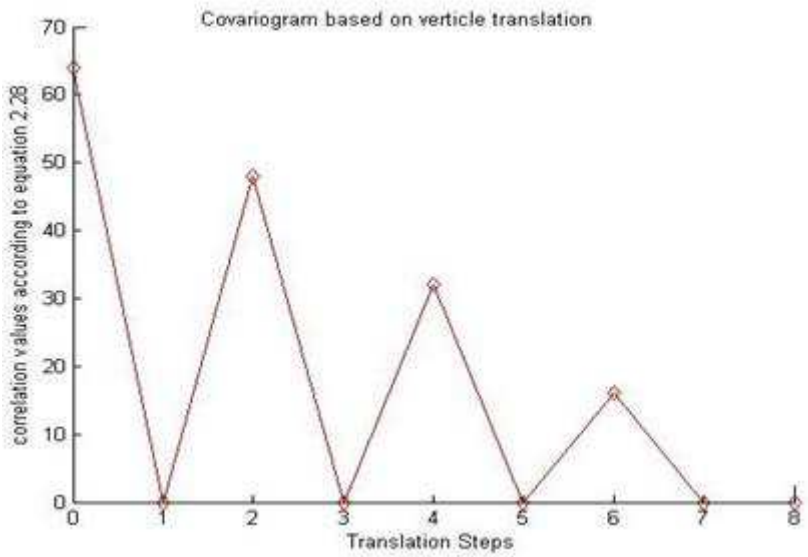


Fig. 2.15 : Covariogram based on correlation of two images, which are translated vertically.

The explanation is as discussed for the previous case of horizontal translation.

Similarity Metrics:

Distances:

Distance generally is defined as the length from point A to point B or the length of straight line segment which separates two points, A and B. In mathematics distance is defined as a quantitative variable d_{xy} which generally satisfies at least the first three of the following conditions:

- $d_{xy} \geq 0$ distance between x and y cannot be negative.
- $d_{xx} = 0$ distance satisfies the "separating" property.
- $d_{xy} = d_{yx}$ distance is symmetric.
- $d_{xy} \leq d_{xz} + d_{zy}$ distance satisfies triangular inequality.

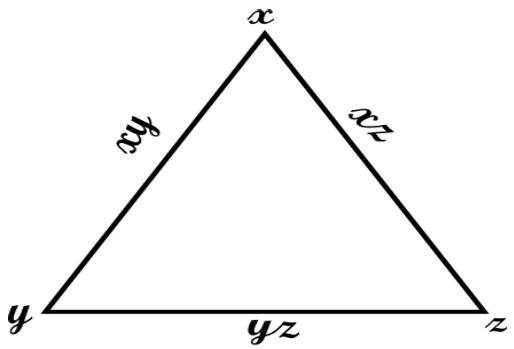


Figure 2.16:Triangle to explain triangular inequality.

For images comparison, two images are considered to be "similar" if the calculated distance between them is sufficiently small. It is to note that distance and similarity are of the same nature, as a smaller distance denotes a higher similarity. Among the too many available tools used as image similarity metrics i.e. distance, correlation, mutual information etc, the distance metric has been widely used in measuring similarities. For example, distance between two images f, g is summation of all the differences between the corresponding pixel intensities. Distance metrics such as Manhattan (L_1), Euclidean (L_2), Weighted-Mean Variance (WMV), Chebychev (L∞), Mahalanobis, Canberra, Bray-Curtis, Squared Chord and Chi-Squared distances were used in [13]. Euclidean distance (also called sum of squared differences) is commonly used for different purposes in image processing but [14] has ruled out its efficiency for all kinds of applications. Harmonic distance and geometric distance were used in [14] as similarity metrics. Earth Mover Distance has been an effective tool used as a

similarity metric [15] and [16] has exploited the distance measure between images I_1 and I_2 with the corresponding Hidden Markov Mesh Models λ_1 and λ_2 for their similarity. Some examples of distances are given below.

Manhattan Distance $L_1\ f,g\ = \ \sum_{i=1}^{n} f_i - g_i$ ... (2.29)

Euclidean Distance $L_2\ f,g\ = \ \sqrt{\sum_{i=1}^{n} f_i - g_i\ ^2}$ ... (2.30)

Mahalanobis Distance $L_{Mah}\ f,g\ = \ \sqrt{f_i - g_i\ ' Cov\ f_i\ '\ x - y}$ ... (2.31)

Correlation:

Correlation as discussed above is also a very important metric for producing covariogram.

$$Cov\ f,g\ = \frac{f_i - \mu_f\ (g_j - \mu_g)}{\delta_f \delta_g} \quad \text{... (2.32)}$$

LIP Version of the above mentioned metrics:

Distances:

LIP Distance is given as $L_\Delta\ f,g\ = M\ _{(x,y)\in D}(\ln\ \frac{M-f}{M-g}\ )^2\ ^{\frac{1}{2}}$ ... (2.33)

LIP Manhattan $L_\Delta^M = \ _\Delta(\frac{1}{S} \circledast (f \triangle g))$ ... (2.34)

where $f \triangle g = \frac{f-g}{1-\frac{g}{M}}$ such that $f > g$ and S= size of Image.

LIP Correlation:

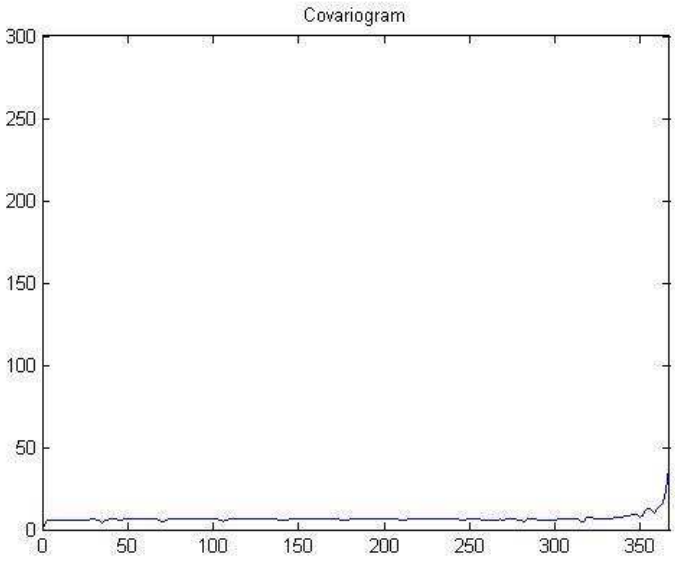
$$Corr = \frac{\ln\ 1 - \frac{225-f}{M}\ \ln(1 - \frac{255-g}{M})}{\sqrt{(\ \ln\ 1 - \frac{225-f}{M}\ )^2}\ \sqrt{(\ \ln(1 - \frac{255-g}{M}))^2}} \quad \text{... (2.35)}$$

LIP Covariogram:

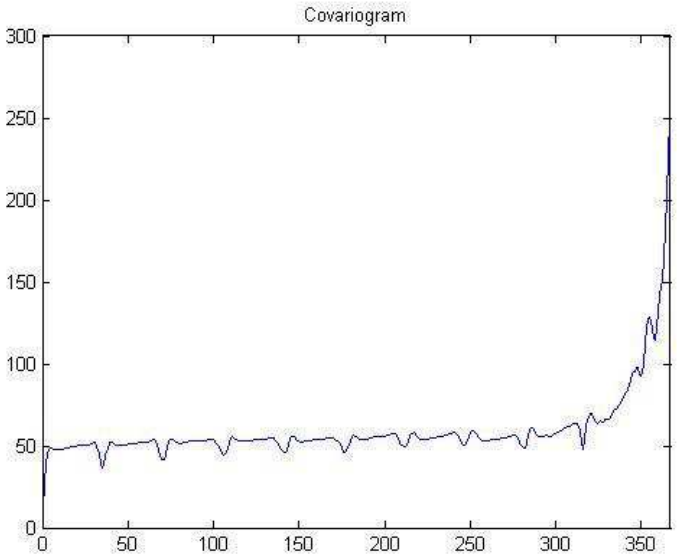
Using the given metrics we obtain the covariogram in LIP framework. Also we show the difference of the LIP covariogram from the classical one: when applied to dark images, it

is much more significant than the classical approaches (for an example, see following Figure 2.17).

a) Initial dark image



b)Covariogram of a) with classical Manhattan distance



c)Covariogram of a) with LIP Manhattan distance
Fig. 2.17 Efficiency of Logarithmic metrics applied to ,dark images

For a complete example, we invite the reader to report to Figure 3.20 of the next chapter.

2-4- Conclusion: Why the LIP framework?

In the first chapter, we have seen that the concept of texture is not mathematically defined. As a consequence, the tools created until now for evaluating textured images present some weaknesses, and particularly the notion of co-occurrence matrices. In fact, if we consider their content informations, such matrices are generally very poor at a statistical level. This is the reason why Haralick has proposed to cumulate in various manners these informations, resulting on the well-known "Haralick parameters". Another drawback of co-occurrence approach is linked to the very large amount of matrices to be computed: the same number as the translations lengths we are interested in, multiplied by the directions in which these translations are applied.

We have proposed a solution to decrease the computation volume by performing a preliminary classification on the studied image, thus limiting the size of the matrices and increasing the average number of occurrences at each location inside the matrix.

Another problem, generally neglected in classical approaches is the essential role of Human Visual System in the perception of texture. This is the reason why we have chosen the LIP (Logarithmic Image Processing) context.

To propose efficient tools for studying textured images, the LIP framework possesses very strong properties:

- It is consistent with human vision, and the notion of texture is clearly dependant of the Human Visual system
- It permits to develop tools weakly sensitive to lighting variations and/or lighting drift
- It furnishes a context where new notions of contrast and associated metrics may be defined

From such metrics, “correlation” tools may be derived in order to build covariograms which take into account Human Vision.

Bibliography of chapter 2:

1. Shapiro, L.G. and G.C. Stockman, *Computer Vision*, Upper Saddle River, NJ: Prentice-Hall, 2001.
2. R. M. Haralick et al. "Textural features for image classification", *IEEE Trans. on SMC*, vol. SMC-3, no. 6, pp.610 -621 1973
3. Arati S. Kurani et al .,"Co-occurrence matrices for volumetric data". The Seventh IASTED International Conference on Computer Graphics and Imaging-CGIM 2004, Kauai, Hawaii, USA. pp. 426-443.
4. P.A. Bautista and M.A. Lambino "Co-occurrence Matrices for Wood Texture Classification" The Samahang Pisika ng Visayas at Mindanao (Physics Society of Visayas and Mindanao) Iligan 2001.
5. Fritz Albregtsen, "Statistical Texture Measures Computed from Gray Level Cooccurrence Matrices", November 5, 2008.
6. Bez N *et al "Covariogram and related tools for structural analysis of fish survey data". In: Baafi E. Y., Schofield N. A., editors. Geostatistics Wollongong`96. Vol. 2. Dordrecht: Kluwer; 1997. p. 1316-1327*
7. Matheron G., 1971, The theory of regionalised variables and its applications, Les Cahiers du Centre de Morphologie Mathematique, Vol. 5, Ecole des Mines de Paris, Fontainebleau, 212p.
8. Olea, R.A. *Geostatistical glossary and multilingual dictionary. Oxford University Press, Oxford, England.* 1991.
9. EMERY Xavier "Géostatistique linéaire" , Ecole des Mines de Paris, 2001
10. Lantuéjoul C. "Geostatistical simulation; models and algorithms" Springer, Berlin(2002)
11. Frączek, Witold, Andrzej Bytnerowicz, and Allan Legge. "Optimizing a Monitoring Network for Assessing Ambient Air Quality in the Athabasca Oil Sands Region of Alberta, Canada." (2009).
12. Mark Dowdall, John O'Dea Geostatistical Analysis of Monitoring Data ENVIRONMENTAL MONITORING – Vol. II . Encyclopedia of Life Support Systems (EOLSS). http://www.eolss.net
13. Kokare M et al., "Comparison of similarity metrics for texture image retrieval" Proceedings of IEEE TENCON Conference, Bangalore (India),2003, pp. 571-575.

14. Jie Yu et al, “A New Study on Distance Metrics as Similarity Measurement.” *ICME*, 2006, pp.533-536.
15. Rubner Y., et al. “The earth mover's distance as a metric for image retrieval.” *International Journal of Computer Vision*, 40(2):99-121, November 2000.
16. Daniel DeMenthon et al : “Image Distance using Hidden Markov Models.”, *Proceedings 15th International Conference on Pattern Recognition*, 2000. pp 143-146.

Résumé Chapitre 2 :

Le deuxième chapitre est dédié à la présentation des principaux outils utilisés classiquement en analyse de texture, en particulier les matrices de co-occurrence, dont nous présentons les difficultés de mise en œuvre et d'interprétation, ainsi que des solutions pour pallier ces difficultés, par exemple l'approche proposée par Haralick. La technique un peu moins classique des covariogrammes est aussi détaillée. Dans la conclusion de cette partie, nous expliquons pourquoi nous avons choisi de transférer ces notions dans le cadre du Modèle LIP (Logarithmic Image Processing).

Chapter 3 – Logarithmic Image Processing Model

3-1- Notations and recalls

3-2- Logarithmic Additive Contrast and associated metrics

3-2-1- Recalls on the classical Michelson contrast

3-2-2- Definition of a Logarithmic Additive Contrast (LAC) in the LIP context

3-2-3- Examples of metrics associated to the Logarithmic Additive Contrast

3-2-3-1- Recalls on functional metrics

3-2-3-2- Logarithmic metrics

3-3- Logarithmic Multiplicative Contrast and associated metrics

3-3-1- Definition of a Logarithmic Multiplicative Contrast (LAC) in the LIP context

3-3-2- Examples of metrics associated to the Logarithmic Multiplicative Contrast

3-3-2-1- Global metric

3-3-2-2- Atomic metric

3-3-2-3- Intermediate metric

3-4- Applications of these metrics notions

Chapter 3 – Logarithmic Image Processing Model

3-1- Notations and recalls:

Introduced by Jourlin et al ([1], [2], [3]), the LIP (Logarithmic Image Processing) Model proposes first a physical and mathematical framework adapted to images acquired in transmitted light (when the observed object is placed between the source and the sensor). Based on the transmittance law, the LIP Model proposes two operations on images allowing the addition of two images and the multiplication of an image by a scalar, each of them resulting in a novel image. Such operations possess strong mathematical properties, as recalled hereafter. Furthermore, the demonstration, by Brailean ([4]) of the LIP Model compatibility with human vision, considerably enlarges the application field of the Model, particularly for images acquired in reflected light on which we aim at simulating human visual interpretation.

In the context of transmitted light, each grey level image may be identified to the observed object, as long as the acquisition conditions (source intensity and sensor aperture) remain stable. An image f is defined on a spatial support D and takes its values in the grey scale [0, M[, which may be written:

$$f : D \subset R^2 \rightarrow [0,M[\subset R$$

Note that within the LIP Model, 0 corresponds to the « white » extremity of the grey scale, which means to the source intensity, i.e. when no obstacle (object) is placed between the source and the sensor. The reason of this grey scale inversion is justified by the fact that 0 will appear as the neutral element of the addition law defined in formula (3). The other extremity M is a limit situation where no element of the source is transmitted (black value). This value is excluded of the scale, and when working with 8-bits digitized images, the 256 grey levels correspond to the interval of integers [0,..,255].

The transmittance $T_f(x)$ of an image f at $x \in D$ is defined by the ratio of the outcoming flux at x by the incoming flux (intensity of the source). In a mathematical formulation, $T_f(x)$ may be understood as the probability, for a particle of the source incident at x, to pass through the obstacle, which means to be seen by the sensor.

The addition of two images f and g corresponds to the superposition of the obstacles (objects) generating respectively f and g. The resulting image will be noted f $\triangle_+$ g. Such an addition is deducted from the transmittance law:

$$T_f \triangle_+ {}_g = T_f \text{ x } T_g \qquad (1)$$

which means that the probability, for a particle emitted by the source, to pass through the "sum" of the obstacles f and g, equals the product of the probabilities to pass through f and g, respectively. Jourlin and Pinoli ([3]) established the link between the transmittance $T_f(x)$ and the grey level f(x) :

$$T_f(x) = 1 - f(x) / M \qquad (2)$$

Replacing in formula (1) the transmittances by their values deducted from (2) yields:

$$f \triangle_+ g = f + g - f.g / M \qquad (3)$$

From this addition law, it is possible ([1], [3]) to derive the multiplication of an image by a real number λ according to:

$$\lambda \triangle_\times f = M - M (1 - f / M)^{\lambda} \qquad (4)$$

Remark 1: such laws satisfy strong mathematical properties. In fact, if I(D, [0,M[) and F(D,]–∞,M[) design respectively the set of images defined on D with values in [0,M[, and the set of functions defined on D with values in]–∞,M[, we have ([3]):

(F(D,]–∞,M[), $\triangle_+$, $\triangle_\times$) is a real vector space and (I(D, [0,M[), $\triangle_+$, $\triangle_\times$) is the positive cone of the precedent.

Considering these results, Pumo and Dhorne ([5]) consider the LIP Model as defining an **Optical Vector Space.**

Remark 2: The introduction of the "over"- space F(D,]–∞,M[), $\triangle_+$, $\triangle_\times$) permits to associate to each function an "opposite" an then obtain the vector space structure. The

opposite of a function f is noted $\triangle$f and is classically defined by the equality f $\overset{+}{\triangle}$ ($\overset{-}{\triangle}$ f) = 0. Its expression is:

$$\triangle f = (-f) / 1 - f/M$$

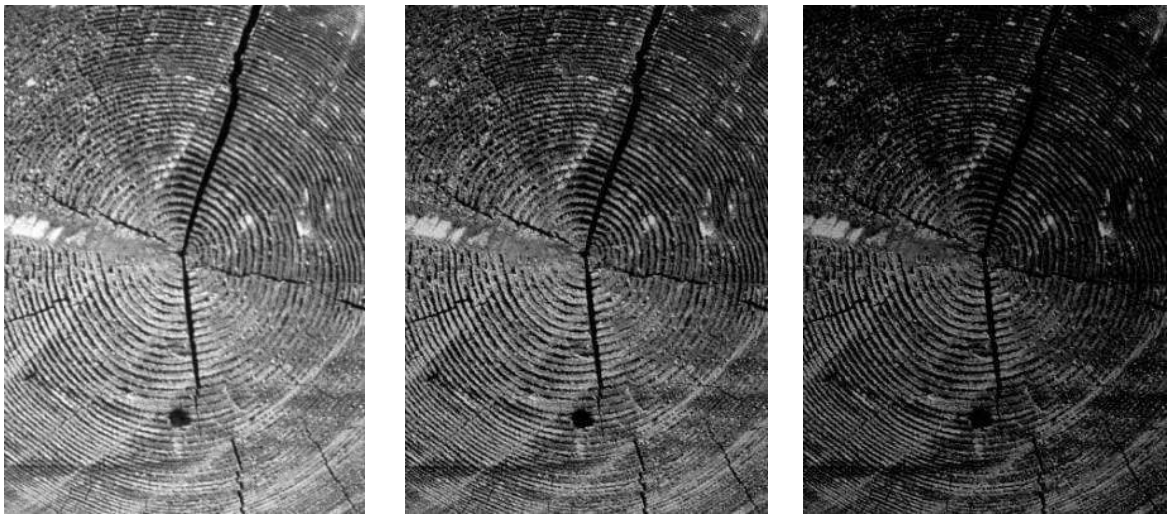
In the same way, the difference between two functions exists and satisfies:

$$f \triangle g = (f - g) / 1 - g/M \qquad (5)$$

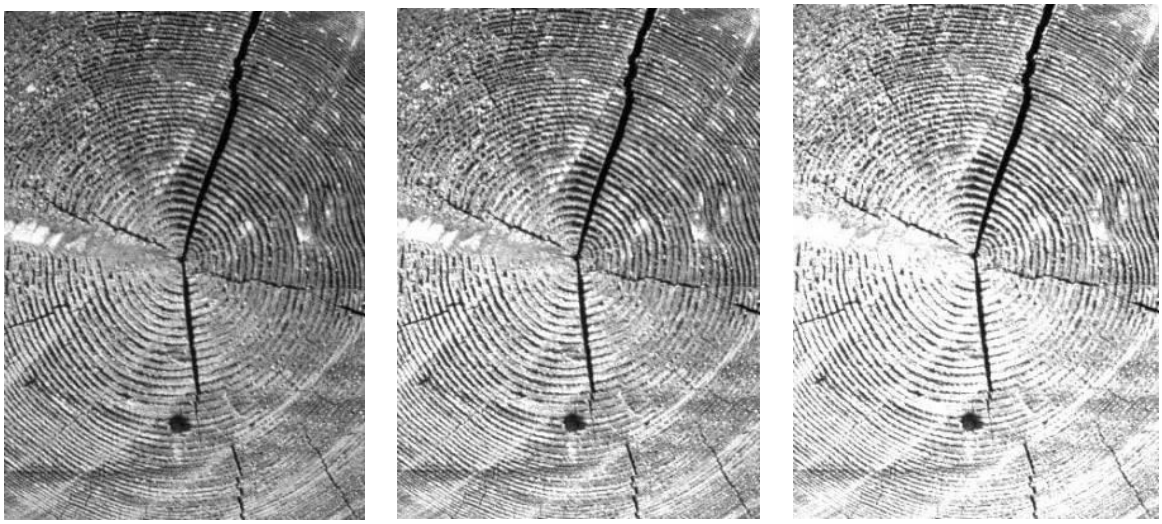
but it is clear that $|f \triangle g| \neq |g \triangle f|$

In case where $g(x) \leq f(x)$ for each x lying in D, formula (5) applies in the space of images ($I(D, [0,M[), \overset{+}{\triangle}, \overset{\times}{\triangle}$) and results in an image of the same space. Note that such a subtraction will be at the origin of the Logarithmic Additive Contrast.

Remark 3: When adding or subtracting a constant C (homogeneous image) to an image f permits darkening or brightening f (cf. Fig. 3-1):



Initial image f $f \overset{+}{\triangle} C$ *(here C=50)* $f \overset{+}{\triangle} C$ *(here C=100)*



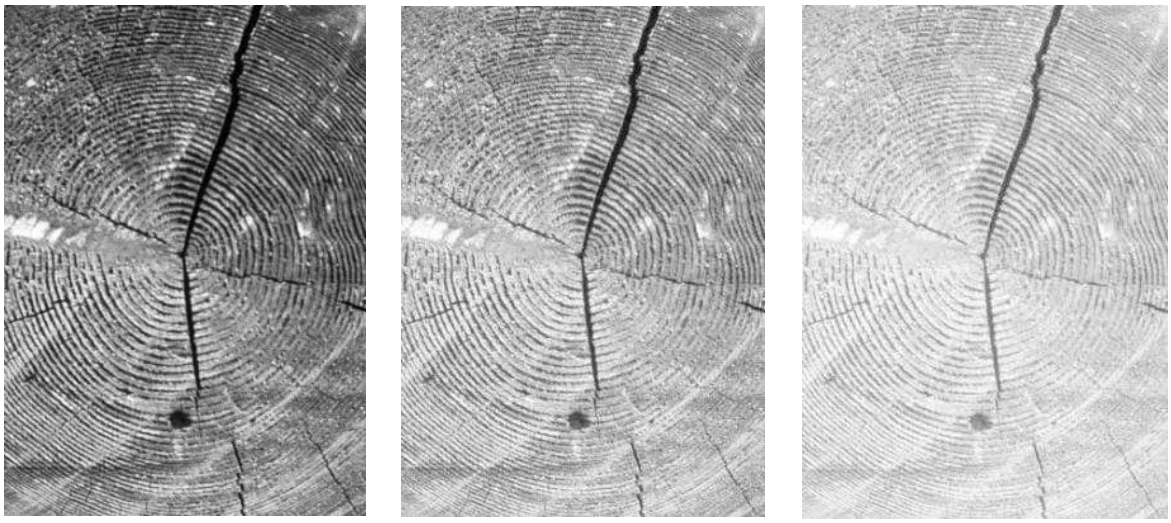
Initial image f $f \triangle C$ *(here C=50)* $f \triangle C$ *(here C = 100)*

Fig. 3-1 Adding (resp. subtracting) a constant from an image

Remark 4: The multiplication of an image f by a real number λ possesses a very strong physical interpretation: in fact λ controls the "thickness" of the considered obstacle which is doubled if λ = 2. More generally, the image $\lambda \otimes f$ is darker than f for $\lambda \geq 1$ and $\lambda \otimes f$ is darker than $\mu \otimes f$ if $\lambda \geq \mu$. On the opposite, $\lambda \otimes f$ will appear brighter than f for $\lambda \leq 1$.

From this remark, it appears clearly that the logarithmic multiplication $\lambda \otimes f$ allows the brightness control of an image (cf. fig.3-2). Furthermore corrections may be applied to images f and g acquired under variable illumination or aperture (cf. fig. 3-3). As an example, λ and μ are computed to obtain the same average grey level (here 128) for $\lambda \otimes f$ and $\mu \otimes f$.

Initial image f *2 ⊗ f* *4 ⊗ f*



Initial image *1/2 ⊗ f* *1/4 ⊗ f*

Fig. 3-2 - Brightness control of an image

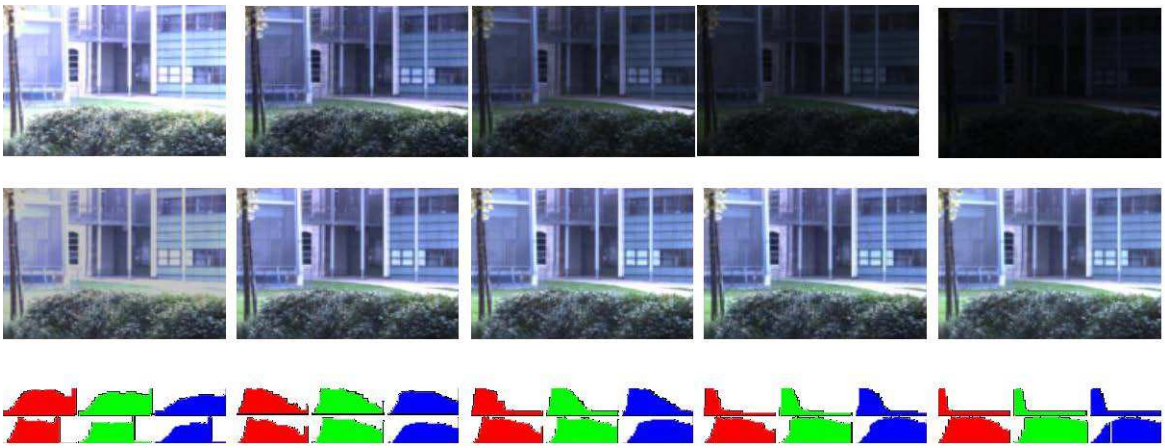


Fig.3-3 First row: initial image acquired under variable aperture conditions.

Second row corresponding homothetic images with average grey level 128.

From remarks 3 and 4, it appears clearly that each law of our vector space structure (addition-subtraction on one hand and scalar multiplication on the other) is efficient to perform brightness modifications. They are then applicable for correcting lighting variations, to enhance low-lighted images (near night vision), all the corresponding algorithms being performed in real time (25 images per second with a classical Pentium).

To conclude this introduction part, note that the presence of a **vector space structure** permits the use of various efficient tools associated to this kind of space: logarithmic interpolation, scalar product...([3]).

3-2- Logarithmic Additive Contrast and associated metrics

3-2-1- Recalls on the classical Michelson contrast

Given a grey level image f and two points x and y lying in D^2, it is common to define the "Michelson" contrast of f at the pair (x,y) according to:

$$C^{m}_{(x,y)}(f) = (\mathrm{Max}(f(x), f(y)) - \mathrm{Min}(f(x), f(y))) / (\mathrm{Max}(f(x), f(y)) + \mathrm{Min}(f(x), f(y)))$$

$$= |f(x) - f(y)| / (f(x) + f(y)) \in [0, 1]$$

Remark 5:

Take care that in the case of this classical contrast, the origin 0 of the grey scale represents the "black" extremity.

Remark 6:

The "Michelson" approach clearly overestimates the contrast of dark pairs of points compared to bright pairs with the same grey levels difference.

Remark 7:

If one of the two considered pixels is black (grey level 0), the value of their Michelson contrast equals 1, independently of the second pixel grey level. Note also that C^m is not defined when $f(x) = f(y) = 0$. For these reasons, we will limit the computation of $C^m_{(x,y)}(f)$ to the case where $f(x) \neq 0$ and $f(y) \neq 0$.

> **In such conditions, the value 1 is not reachable: $C^m_{(x,y)}(f) \in [0, 1[$. This result will appear essential in order to demonstrate an explicit link between the Logarithmic Additive Contrast and the Michelson contrast.**

3-2-2- Definition of a Logarithmic Additive Contrast (LAC) in the LIP context

In the LIP framework, Jourlin et al. ([6]) introduced the Logarithmic Additive Contrast (LAC) noted $C^{\triangle+}_{(x,y)}(f)$ of a grey level function f at a pair (x,y) of points lying in D^2. It is defined according to the following equation:

$$\text{Min}(f(x), f(y)) \mathbin{\triangle+} (C^{\triangle+}_{(x,y)}(f)) = \text{Max}(f(x), f(y)) \quad (6)$$

"Optical" interpretation: Such a contrast represents the grey level which must be added (superposed) to the brightest point (smallest grey level) in order to obtain the darkest one (highest grey level). Then this logarithmic contrast may be visualized without any normalization.

Using the addition formula (3) yields:

$$C^{\triangle+}_{(x,y)}(f) = |f(x) - f(y)| / (1 - \text{Min}(f(x), f(y)) / M) \quad (7)$$

The same reasoning permits to define the Logarithmic Additive Contrast $C \overset{+}{\triangle}_x(f,g)$ between two grey level functions f and g at a same point x of their spatial support D:

$$C \overset{+}{\triangle}_x(f,g) = |f(x) - g(x)| / (1- \text{Min}(f(x), g(x)) / M) \quad (7 \text{ bis}))$$

Properties and results:

a) It is possible to express this contrast as a LIP subtraction:

$$C \overset{+}{\triangle}_{(x,y)}(f) = \text{Max}(f(x), f(y)) \overset{-}{\triangle} \text{Min}(f(x), f(y))$$

b) The LAC is clearly a sub-additive and homogeneous operator on the space of grey level images I(D, [0,M[):

$$C \overset{+}{\triangle}_{(x,y)}(f \overset{+}{\triangle} g) \leq C \overset{+}{\triangle}_{(x,y)}(f) \overset{+}{\triangle} C \overset{+}{\triangle}_{(x,y)}(g)$$

$$C \overset{+}{\triangle}_{(x,y)}(\lambda \overset{\times}{\triangle} f) = \lambda \overset{\times}{\triangle} C \overset{+}{\triangle}_{(x,y)}(f)$$

c) It is important to observe that the considered contrasts $C^m_{(x,y)}(f)$ and $C \overset{+}{\triangle}_{(x,y)}(f)$ are of the same nature in the sense that each of them enhances the contrast of dark pairs of points compared to bright pairs with the same grey level difference |f(x)-f(y)|.
To illustrate that remark, let us consider the following results (Fig.3-4):

f(x)=240 f(y)=200 LIP Additive Contrast: $C \overset{+}{\triangle}_{x,y}^{\triangle} f = 183$

f(x)=55 f(y)=15 LIP Additive Contrast: $C \overset{+}{\triangle}_{x,y}^{\triangle} f = 42$

f(x)=15 f(y)=55 Michelson Contrast: $C^m_{x,y} f$ x 255 = 146

f(x)=200 f(y)=240 Michelson Contrast: $C^m_{x,y} f$ x 255 = 23

Fig.3-4 Comparison of LIP Additive and Michelson contrasts for pixels pairs with the same grey level difference (here 40)

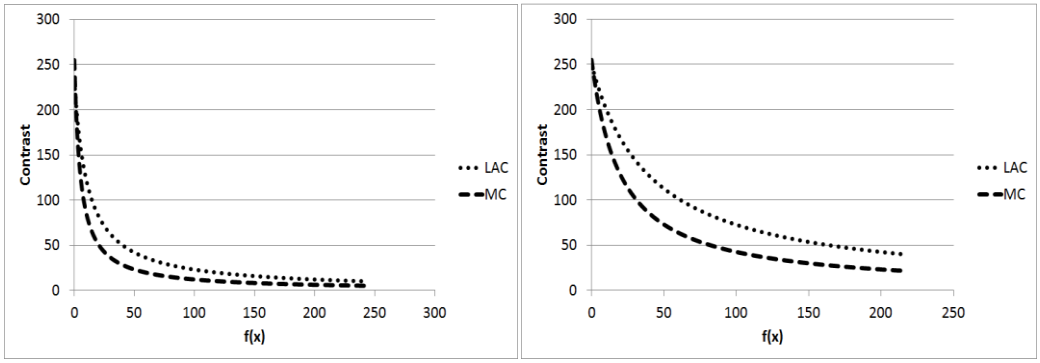
Comment: to obtain comparable values between the two contrasts, the Michelson one has been multiplied by 255. For the same reason, the chosen grey level values take into account the grey scale inversion (the "white" extremity corresponds to 0 in the LIP context).

More generally, for each pair (x,y) of pixels presenting a constant grey level difference k, i.e. $f(y) = f(x) + k$ in the LIP scale (y is darker than x), let us compute:

$C \overset{+}{\triangle}_{(x,y)}(f)$ and

$C^{m}_{(x,y)}(M - 1 - f)$

That means an inversion of the LIP scale (the function f becomes $M - 1 - f$). Moreover, the "Michelson" contrast being always lying in the interval [0, 1[, must be multiplied by 255 to be compared to $C \overset{+}{\triangle}$. Finally, we present in the following Fig. 3-5 the representative curves of $C \overset{+}{\triangle}_{(x,y)}(f)$ and $255 \times C^{m}_{(x,y)}(M - 1 - f)$ for various values of k.



a) b)

Fig. 3-5 The LAC and Michelson contrasts for k =10 (a) and k = 40 (b).

Comment: as previously observed, it clearly appears that for a same value k, the contrast in the LIP sense is greater than in the Michelson one. Nevertheless, the shape similarity between the curves let suppose that some link may exist between the two contrasts, as demonstrated by the following result:

Theorem 1: the Michelson contrast is a logarithmic contrast, according to the formula

$$\mathbf{M\, C^{m}(f(x), f(x) + 2k) = C \overset{+}{\triangle}(M - f(x), M - f(x) - k)}$$

Proof: The following Fig 3-6 represent the classical grey scale and the inverted (LIP) grey scale:

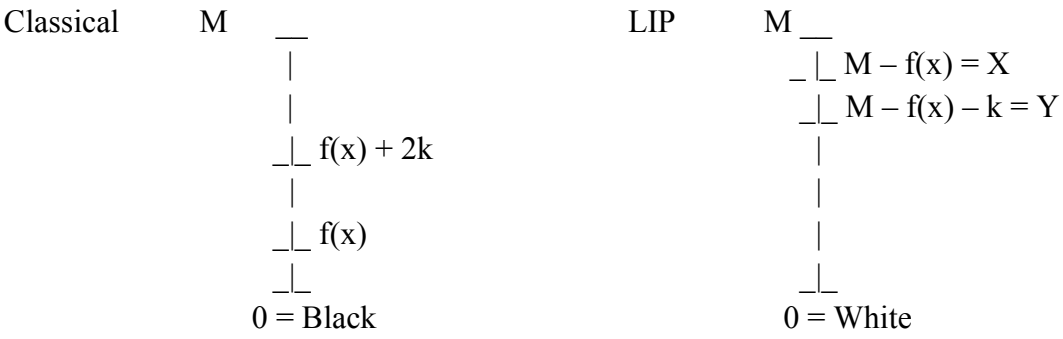


Fig. 3-6 Classical grey-scale and inverted one in the LIP context

We can write:

- in classical situation: $C^{m}_{(x, y)}(f) = (f(y) - f(x)) / (f(y) + f(x))$
  $= 2k / (2f(x) + 2k)$
  $= \mathbf{k / (f(x) + k)}$

- in the LIP context: $C_{\triangle +\ (X, Y)}(f) = | X - Y | / (1 - Min (X, Y) / M)$
  $= k / (1 - (M - f(x) - k) / M)$
  $= \mathbf{Mk / (f(x) + k)}$

and the formula is established.

Fundamental remark: the precedent result gives a precise "physical" meaning to the Michelson contrast. As the LAC, it is interpretable in terms of transmittance. Note that the "thickness" k separating the considered grey levels must be doubled for Michelson. This fact explains why, at the beginning of our comparison, the LAC seemed more "sensible".

Now let us present some applications of the LAC.

3-2-3- Examples of Metrics associated to the Logarithmic Additive Contrast

The mathematical field of "functional analysis" is mainly devoted to the study of functions and spaces of functions. Because the LIP Model gives the set of images a vector space structure, we are clearly interested in the "functional" approach. Among the

considerable number of tools created by mathematicians, we will particularly focus here on the concept of “metrics”. In fact, when studying images, we are obviously interested in detecting the presence of some object of interest (target) inside an image or in tracking this target in a sequence of images. It is also a very common task (not always easy) to estimate the similarity (or the differences) between two images in order, for example, to perform defects detection in industrial control. The concept of metrics is well adapted for such objectives.

3-2-3-1- Recalls on functional metrics

Global metrics

Given a pair (f,g) of real valuated functions which are definite and integrable on a real interval [a,b], it is classical to define a metric noted d_1 according to:

$$d_{1,}(f,g) = \int_{[a,b]} |f(x) - g(x)| \, dx$$

This definition is obviously transposable onto an analog image with a double integral of the difference |f(x)-g(x)|, computed on the points (x,y) of the region of interest (D or a subset R of D):

$$d_{1,D \text{ or } R}(f,g) = \iint_{D \text{ or } R} |f(x,y) - g(x,y)| \, dxdy$$

In the digital version, it is transformed into the double sum of the differences between pixels grey levels according to the rows and columns, multiplied by the area of one pixel. It thus evaluates the “volume” situated between the representative surfaces of images f and g:

$$d_{1,D \text{ or } R}(f,g) = [\sum \sum_{(i,j)\in D \text{ or } R} |f(i,j) - g(i,j)|] \text{ x (area of a pixel)}$$

Comment: More generally, we can use metrics derived from the norms associated with the L^p spaces (spaces made up of functions whose p^{th} power is integrable):

$$L^p \rightarrow d_p(f,g) = \left(\int_{[a,b]} |f(x) - g(x)|^p \, dx \right)^{1/p}$$

and their bi-dimensional continuous or digital versions.

All these metrics are considered of "global" or "diffuse" nature, in reference to measure theory. It means they are null when computed on neglectable sets, i.e. finite sets when considering digital images. By definition, such metrics produce an averaged

information, and are then inefficient in detecting small sized differences between two functions or two images (Fig. 10 – a)).

Atomic metrics

On the contrary, we can use "atomic" metrics, similar to measures using “weighted” points (Dirac measures). They are then perfectly adapted in detecting small differences, even as small as a pixel (Fig. 10 – b)). The most typical example is the metric d_∞ derived from the norm of uniform convergence in the L^∞ space, which is computed on the point realizing the greatest difference between f and g:

$$d_\infty(f, g) = \text{Sup}_{x\in[a,b]} |f(x)-g(x)|$$

It is defined in the same way on a two-dimensional region or domain:

$$d_\infty(f, g) = \text{Sup}_{(x,y) \in R \text{ or } D} |f(x,y)-g(x,y)|$$

and in digital version:

$$d_\infty(f, g) = \text{Sup}_{(i,j) \in R \text{ or } D} |f(i,j)-g(i,j)|$$

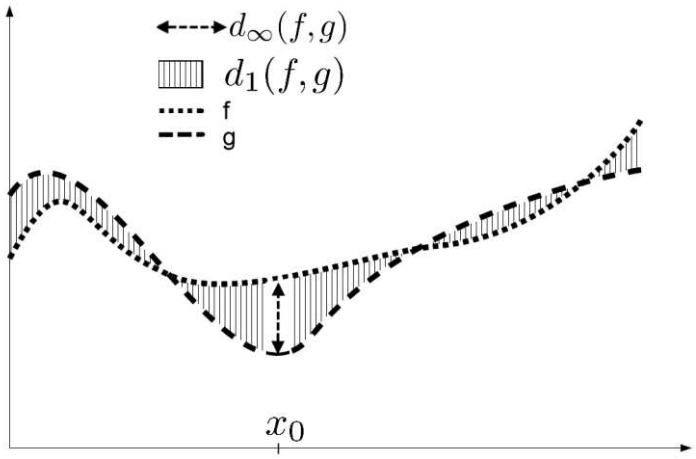
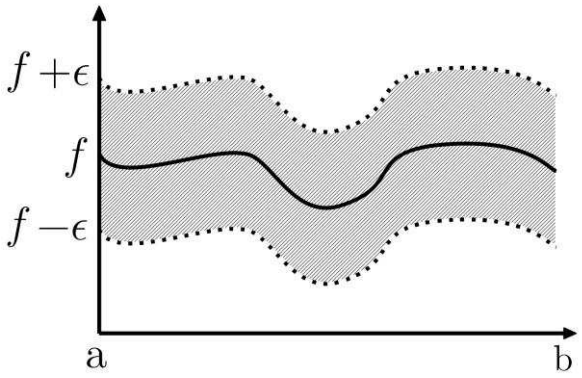


Fig 3-7 The value of $d_1(f,g)$ corresponds to the hatched area between the representative curves of f and g.
The distance $d_\infty(f, g)$ is realized at the point x_0

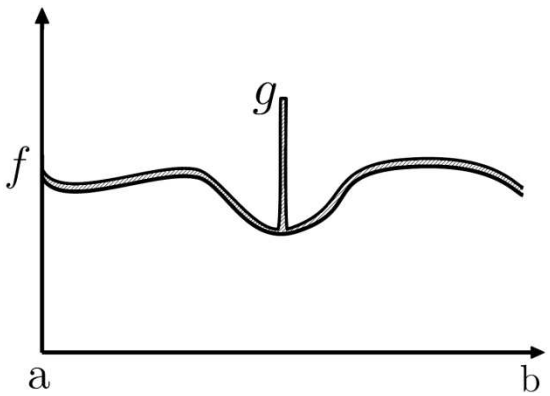
Neighborhoods generated by the precedent metrics:

One of the major interests of the "metric" tool resides in its associated topology, i.e. in the neighborhoods it generates. The shapes of such neighborhoods are totally different for the d_1 and d_∞ metrics. In fact, given a function f, each function g verifying $d_\infty(f, g) \leq \varepsilon$ satisfies $|f(x)-g(x)| \leq \varepsilon$ for every point x lying in the considered interval or region. It means that g belongs to a **"tolerance tube"** around f (Fig. 3-8- a)). This remark explains why d_∞ is called "uniform convergence metric". The same result holds for images, the tolerance tube becoming the volume located between the translated representative surfaces of f according to + ε and – ε.

When considering the "global" metrics $d_1,.., d_p$, an ε – neighbor of a given function f is totally different from a tube: it is an **unbounded set**! In fact, a function g belonging to the ε – neighbor of f may present at some point x an arbitrary large difference |f(x)-g(x)| and a very small area located between f and g (Fig. 3-8- b))



a)



b)

Fig. 3-8- a) The tolerance tube of f is represented by the hatched area

Fig.3-8- b) The difference between a function f and a function g lying in the ε – neighbor of f may be arbitrarily large.

3-2-3-2- Logarithmic metrics

Now let us present the logarithmic versions of all these metrics. Preliminary results on this subject have been exposed in [7], and the interested reader will refer to a detailed presentation ([8]).

The "global" logarithmic metric d1 $\overset{+}{\triangle}$

Given two grey level functions f and g, let us recall what represents their LAC (Logarithmic Additive Contrast) $C \overset{+}{\triangle}_{(x,y)}(f,g)$ at a point (x,y) of the spatial domain D. It is defined by the relation (cf. formula 7 bis):

$$C \overset{+}{\triangle}_{(x,y)}(f,g) = \mathrm{Max}(f(x,y), g(x,y)) \ \overset{-}{\triangle} \ \mathrm{Min}(f(x,y), g(x,y))$$

and is expressed according to:

$$C \overset{+}{\triangle}_{(x,y)}(f,g) = |f(x,y) - g(x,y)| \,/\, (1 - \mathrm{Min}(f(x,y), g(x,y)) / M)$$

Comment: a point of the spatial support D is generally noted x when no distinction between rows and columns is necessary, and (x,y) otherwise.

Now, a summation of such contrasts on all the elements of D or a region of D makes possible to exhibit a novel metric $d_1 \overset{+}{\triangle}$, tractable in the LIP context, and defined on the space I(D,[0,M[), either on the whole domain D or a region $R \subset D$:

$$d_1 \overset{+}{\triangle}_{D \text{ or } R}(f,g) = \iint_{D \text{ or } R} \mathrm{Max}(f(x,y), g(x,y)) \ \overset{-}{\triangle} \ \mathrm{Min}(f(x,y), g(x,y)) \, dxdy \quad (10)$$

Applied to digital images, such a metric becomes:

$$d_1 \overset{+}{\triangle}_{D \text{ or } R}(f,g) = [\ \sum \overset{+}{\triangle} \sum \overset{+}{\triangle}_{(i,j) \in D \text{ or } R} \mathrm{Max}(f(i,j), g(i,j)) \ \overset{-}{\triangle} \ \mathrm{Min}(f(i,j), g(i,j))\ ]$$

$$\times \text{ (area of a pixel)} \qquad (11)$$

where $\sum^{\triangle}$ represents the summation, in the LIP sense, of the contrasts between f and g at each point (i,j) of the considered region.

Remarks

8) The term "dxdy" in formula (10) becomes in digital version "area of a pixel" (cf (11))

9) The presence, in formula (11), of the "area of a pixel" permits to obtain a result independent of the numerization (sensor resolution) and to preserve the homogeneity of formula (10) which clearly represents a volume.

10) Formulas (10) and (11), respectively in continuous or digitized expression, estimate a "contrast volume" separating the representative surfaces of f and g.

11) A real difficulty arises if we need the visualization of such a distance. In fact, each contrast $\text{Max}(f(i,j), g(i,j)) \overset{\triangle}{-} \text{Min}(f(i,j), g(i,j))$ is a grey level and the sum of an arbitrary number of grey levels, although it remains a grey level, in general quickly approaches the limit value M. To solve this problem, we propose to replace this cumulative distance by an average contrast, taking into account the number of points present in the region of interest : if #R denotes the cardinal of R, we replace formula (11) by :

$$d_1 \overset{\triangle}{+}_R(f,g) = (1/\#R) \overset{\triangle}{\times} [\sum \overset{\triangle}{+} \sum \overset{\triangle}{+}_{(i,j)\in R} \text{Max}(f(i,j), g(i,j)) \overset{\triangle}{-} \text{Min}(f(i,j), g(i,j))] \qquad (12)$$

12) Formulae (11) and (12) are obviously compatible because the average contrast of (12) corresponds to the "contrast volume" of (11): in fact, computing an average contrast from the volume expression consists in dividing by "area(D)" whose value is (area of a pixel) multiplied by (#D)

The "atomic" metric d_∞ $\overset{\triangle}{+}$

We start from the expression, recalled below, of the "classical" atomic metric d_∞:

$$d_\infty(f, g) = \text{Sup}_{x\in R \text{ or } D} |f(x)-g(x)|$$

In the LIP context, if we refer to (5), the same formula with a LIP subtraction is not correct because the expression $|f(x) \overset{\triangle}{-} g(x)|$ is not always defined, according to the fact that f(x) is greater than g(x) or not. It must be replaced by the logarithmic difference between the maximum and the minimum of the pair (f(x), g(x)):

$$d_\infty \left(+\right) f, g) = \text{Sup}_{x \in R \text{ or } D} (\text{Max } (f(x), g(x)) \ \left(-\right) \ \text{Min } (f(x), g(x)) \qquad (13)$$

As for d_∞ (f, g), such a metric seems theoretically well adapted to industrial control and more precisely to point out possible defects, but it is very « sensible » because determined by one unique point, which may correspond to a very small, and then acceptable, defect. Given a reference function without defects, noted f and the same image with defects, noted g, it is possible that a unique pixel x_0 corresponds to the "Sup" value: in this case, only x_0 will be detected. In most cases, this kind of answer is not completely satisfactory because the defect size around x_0 may be larger than one pixel. Furthermore the method may ignore a number of other defects whose contrasts with the reference image are less than d_∞ (f, g). A possible answer to this problem is to perform a threshold t on the contrasts map between f and g, in order to know where g significantly differs from f and if x_0 is really an isolated point.

3-3- Logarithmic Multiplicative Contrast and associated metrics

3-3-1- Definition of a Logarithmic Multiplicative Contrast (LMC) in the LIP context

Now let us define a contrast notion based on the scalar multiplicative law of the LIP framework. This notion is completely different of existing contrasts. Nevertheless, we took care to give it a **physical meaning**, once more based on the transmittance law. It will be seen that this contrast, which is a real number instead of a grey level, presents the advantage to be more sensible near the white extremity of the grey scale than near the black one.

If f represents a grey level image and x and y two points of D, we define the logarithmic multiplicative contrast (LMC), noted $C \left(\times\right)_{(x,y)} (f)$ as the real number ratio of Max (f(x), f(y)) by Min (f(x), f(y)). In other words, it represents the number by which the brightest grey level must be multiplied, in the LIP sense, in order to obtain the darkest grey level:

$$C \left(\times\right)_{(x,y)} (f) \left(\times\right) \text{Min } (f(x), f(y)) = \text{Max } (f(x), f(y)) \qquad (14)$$

In the same way, the logarithmic multiplicative contrast may be defined for a pair of grey level functions (f,g) at each point x of D:

$$C \overset{\times}{\triangle}_{(x)} (f,g) \overset{\times}{\triangle} \text{Min}(f(x), g(x)) = \text{Max}(f(x), g(x)) \qquad (15)$$

Remarks

18 - In each of the precedent situations, the multiplicative contrast clearly corresponds to the number of times we must "add" (physically superpose) the « Min » grey level between the source and the sensor to obtain an attenuation equivalent to the « Max » grey level.

19 – In order to illustrate the better sensitivity of the Logarithmic Multiplicative Contrast for the bright part of the grey scale, let us consider a pair of grey levels f(x) and f(x) + k. We compute the Multiplicative Contrast between f(x) and f(x) + k when f(x) varies in the interval [0, 255-k], in a classical grey scale (0 = black). In Figure 3-9, the curves corresponding to these contrasts are represented for various values of the grey level difference k (here k = 30, 50, 100, 150, 200).

The non-linearity of the Logarithmic Multiplicative Contrast appears clearly, as well as its attenuation near the black extremity of the grey scale and its high sensitivity towards the white extremity.

Such a behavior must be compared to the LAC one, which overestimated the contrasts of dark points.

This situation present a very important advantage: it permits the processing of over-lighted images as well as low-lighted ones.

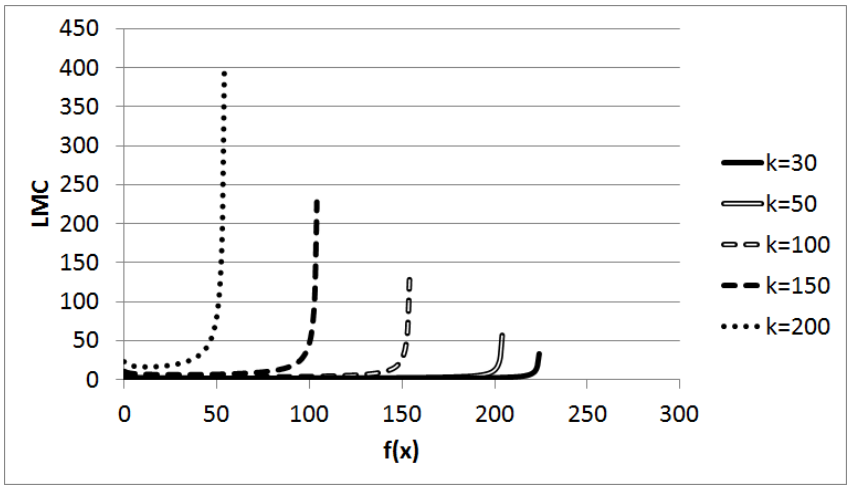


Fig.3-9: Curves representing the LMC between f(x) and f(x)+k for various values of k: k=30,50,100,150 et 200

20 - The additive contrast $C^{+}_{(x,y)}(f)$ was by definition a **grey level**. The multiplicative one is a **real number**, which reaches $+\infty$ if Min (f(x), f(y)) = 0 and Max (f(x), f(y)) $\neq$ 0. In the case where f(x) = f(y) = 0, we can decide to estimate the contrast to the unit value, in analogy with the general case of two identical grey levels.

To solve the problem of the possible infinite value, a simple and practical solution for digital images consists in **replacing each null grey level by the unit one**, which does not affect significantly nor the aspect neither the interpretation of the studied image.

In such conditions, all contrast values are finite and may be visualized as grey levels, after a normalization, if necessary: in fact, if we work in the classical situation of 8-bits images, we dispose of a grey scale of 256 (limited to 255) grey levels (from 1 to 255), with a nearly linear distribution. Thus the largest possible value $C^{\times}_{max}$ for the multiplicative contrast is reached for the pair of grey levels (1, 255), and satisfies:

$$(C^{\times}_{max}) \, ^{\times}(1) = 255 \qquad (16)$$

yielding $C^{\times}_{max} \approx 1416{,}79$

The normalized (lying in the interval [0, 255]) multiplicative contrast $C^{N\times}_{(x,y)}(f)$ of f at a pair (x,y) satisfies then:

$$C^{N\times}_{(x,y)}(f) = 255 \times (1/ C^{\times}_{max}).C^{\times}_{(x,y)}(f) \qquad (17)$$

sometimes denoted $C^{N\times}_{(f(x),t)}$ in the following when at less one of the considered grey levels is not defined by f but a threshold t for example.

In the same way, the contrast of a pair (f,g) of images at a point x is given by:

$$C\overset{N}{\triangle}_{(x)}(f,g) = 255 \times (1/ C\overset{}{\triangle}_{max}).C\overset{}{\triangle}_{(x)}(f,g) \qquad (18)$$

3-3-2- Examples of metrics associated to the Logarithmic Multiplicative Contrast

3-3-2-1- The "global" metric

In section 3-2-3-1, we have seen that on the space L^1 of integrable functions, it is classical to compute the distance $d_1(f,g)$ as the integral of the difference $|f(x)-g(x)|$ where x varies in the region of interest, which may be for example an interval of R. The same distance may be defined on images with a double integral on a subset of R^2...

Replacing in the integral the local distance $|f(x)-g(x)|$ by the logarithmic multiplicative contrast $C\overset{N}{\triangle}_{(x)}(f,g)$ and then cumulating such contrasts on the elements of D or a region of D, generates a novel metric $d\overset{N}{\triangle}_{1}$ defined on the space I(D,[0,M[), either on the whole domain D or a region $R \subset D$. Such a metric is tractable in the LIP framework and is expressed according to:

$$d\overset{N}{\triangle}_{1,D \text{ or } R}(f,g) = \iint_{D \text{ or } R} C\overset{N}{\triangle}_{(x)}(f,g)\, drdc \qquad (19)$$

$$= \iint_{D \text{ or } R} 255 \times (1/ C\overset{}{\triangle}_{max}).\, C\overset{}{\triangle}_{(x)}(f,g)\, drdc$$

where the coordinates of a point x in terms of rows and columns are noted (r,c)

Applied to digital images, such a metric becomes:

$$d\overset{N}{\triangle}_{1,D \text{ or } R}(f,g) = \left[\sum \sum_{(r,c)\in D \text{ or } R} C\overset{N}{\triangle}_{(x)}(f,g) \right] \times (\text{area of a pixel}) \qquad (20)$$

Remarks

23- The presence, in formula (20), of the "area of a pixel" permits to obtain a result independent of the digitization scale.

24- Formulas (19) and (20), respectively in continuous or numerical expression, estimate a "multiplicative contrast volume" separating the representative surfaces of f and g.

25- Obviously, the size (number of pixels, or "cardinal" noted #) of the considered region R plays a role in the distance estimation. To suppress it, we can divide the distance by the region area i.e. (cardinal(R) x (area of a pixel), and thus obtain an averaged value noted

$d^{N,A}_{1,D\text{ or }R}(f,g)$ and defined by:

$$d^{N,A}_{1,D\text{ or }R}(f,g) = (1/\#R).\ [\ \sum\ \sum_{(r,c)\in D\text{ or }R}\ C^{N}_{(x)}(f,g)] \qquad (21)$$

3-3-2-2- The « atomic » metric

When applied to a subset $D \subset R^2$ or a region R of D, remember the « uniform convergence metric » is defined by:

$$d_\infty(f, g) = \mathrm{Sup}_{x\in R\text{ or }D}\,|f(x)-g(x)|$$

If we aim at transferring this formulation in the context of multiplicative contrast, the expression |f(x)-g(x)| must be replaced by $C^{N}_{(x)}(f,g)$ and thus:

$$d^{N}_{\infty}(f, g) = \mathrm{Sup}_{x\in R\text{ or }D}\ \ C^{N}_{(x)}(f,g) \qquad (22)$$

Remark 26: the two precedent metrics are comparable to those associated to the LAC in the sense that d^{N}_{1} and d^{N}_{∞} have respectively a "global" or "atomic" behavior. The first one evaluates the resemblance of an image g to a given image f through the belonging of g to unbounded neighbors of f and the second one through the belonging of g to "tolerance tubes" of f.

The main difference between the two metrics d^{N}_{1} and d^{N}_{∞} on one hand and the pair d_{1} and d_{∞} on the other hand is linked to their physical meanings which imply a better sensitivity of the multiplicative metrics d^{N}_{1} and d^{N}_{∞} on the white part of the grey scale. Thus they will produce their best results for over-lighted images or on the light-grey part of an image.

As aforementioned for "additive" metrics, it may be useful to introduce an intermediate solution.

3-3-2-3- The "intermediate" metric

When applied to industrial control, the atomic metric seems theoretically well adapted to defects detection, and also to biomedical applications when small objects have to be detected. The problem is the extreme « sensitivity » of this metric because it is determined by one unique point.

This is the reason why we introduce an "intermediate" definition between the "diffuse" distance $d^{N}_{1,D \text{ or } R}$ and the "atomic" one d^{N}_{∞}, noted $d^{N}_{1, supR}$.

It consists first in choosing a subset (region) R of the domain D, then to compute the distance $d^{N}_{1, R}(f,g)$ for each position of R inside D, and finally to define:

$$d^{N}_{1, supR}(f,g) = Sup_{R \subset D}\ d^{N}_{1, R}(f,g) \qquad (23)$$

Remarks

27 - As the atomic one, the intermediate metric obviously detects one position of the region R inside D corresponding to the largest distance between f and g. If this distance is not acceptable, the controlled product must be rejected.

28 - In order to detect the set of all the defects, acceptable or not, we propose to choose a particular pixel inside the region R, for example the gravity center c. For each x of D, denote Rx the region R when its gravity center c is superposed to x, and compute the distance $d_{N_{1, Rx}}(f,g)$ which becomes the grey level of x. We obtain a grey level image. If we apply to it a threshold corresponding to the "acceptable error" ε, all the pixels presenting a grey level greater than ε must be interpreted as defects.

3-4- Applications of these Metrics notions

The problem of defects detection in industrial control has been aforementioned. Now some applications are going to be presented in order to illustrate the interest of the introduced metrics.

Scene modifications

Given a reference image f of a certain scene, and a current image g of the same scene, a recurrent question is: what differs between f and g?

Such a question covers various domains:

- Comparison of two satellite images of the same site at different times, the aim being to put in evidence new buildings or roads, expansion of towns, and also modifications of agriculture areas…
- Robots safety, the aim being to detect the entrance of a person in a forbidden area.
- Industrial or military sites surveillance.
- Automated crossroads supervision
- ….

The metric tools can obviously answer such problems.

Pattern recognition and target tracking

When possessing a reference image f of a region (object) of interest, and given a current image g, the addressed question is to decide if, inside g, some region very similar to f

exists or not. Such a similarity may be estimated by the distance between f and a considered sub-region of g.

It is also possible to extend this method to a video: the target of interest is detected on the first image of the sequence and this target's image is used as searched region on the second image, and so on. The tolerance accepted on the considered metric permits small variations of the target in terms of shape, orientation and grey levels, allowing the tracking of the target.

Metrics as correlation tools

The notion of correlation is a fundamental tool in the field of statistics. It is also commonly used in image processing in order to perform the optimal superposition of two images, in particular to compensate sensor motions.

The role of metrics as correlation tools is evident: given two images, one may be moved (translation, rotation…) in reference to the other until the correlation becomes optimal, that is to say when a certain distance between them reaches a minimal value.

A perfect example of this approach has been given in the precedent section, when a given target has been moved of the reference image in order to detect the locations corresponding to a minimal distance.

Characterization of pseudo-periodic textures

The concept of "texture" is rather difficult to define with rigorous terms, in the sense of mathematics for example. In this section, our ambition is not to go into this subject in depth, so we will limit us to propose some new tools to study images presenting a pseudo-periodic texture. For examples of such images, see Fig. 3.10.

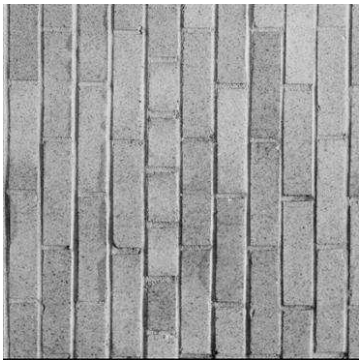


Fig. 3.10 Initial pseudo-periodic image: bricks wall

A recurrent question about pseudo-periodic images is to estimate a period in some direction. One of the major techniques to answer this question consists, given an image f, in computing a covariogram (notion introducted by Matheron in [9]) of function f in some direction. To do that, f is translated related to itself in the given direction. For each value of the translation vector h, the "correlation" between f and its translated f_h is computed. Then the results are presented as a curve.

Example of covariogram is presented in Fig. 3.11: it has been applied to the image of Fig. 3.10 on which the correlations are computed for horizontal translations with the distance $d_1 \triangleq (f, f_h)$. The locations where this distance presents a relative minimum value correspond to local optimal correlations.

Note that if the first minimal value is reached for a translation vector h_o, the expected estimation of the period of f is precisely given by h_o.

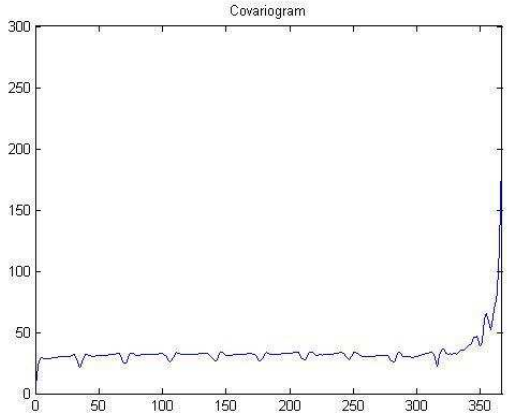


Fig. 3.11 Covariogram of Fig.3.10 showing the average "period" of bricks.

Remarks

13- For images presenting complex textures, the covariogram curve may be noisy. In such conditions, the automated extraction of the estimated period is rather difficult. An efficient way for de-noising the covariogram consists in applying a classification algorithm (k-means for example) to restrict the number of grey levels present on the initial image f.

14- As previously said, the advantages of applying logarithmic metrics are multiple: they are perfectly adapted to images acquired in transmitted light, they are defined in a

framework consistent with human vision, and they are weakly dependant of illumination changing (Fig. 3.12), unlike classical metrics as d_1. Images of Fig. 3.12 have been presented by Inam Ul Haq (cf. [10]).

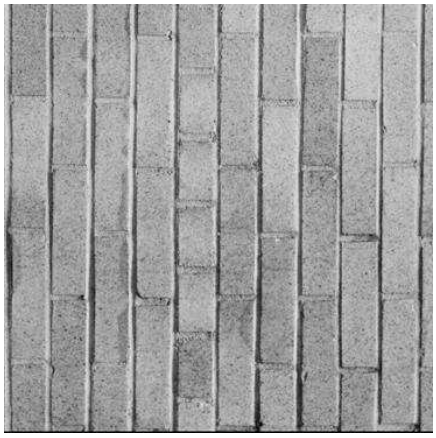


Fig. 3.12 a) Initial image f

Fig. 3.12 b) The same under low-lighting conditions.

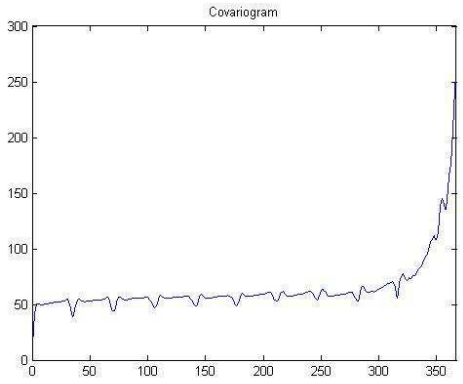


Fig. 3.12 c) The covariogram of a) computed with d_1 ⊞

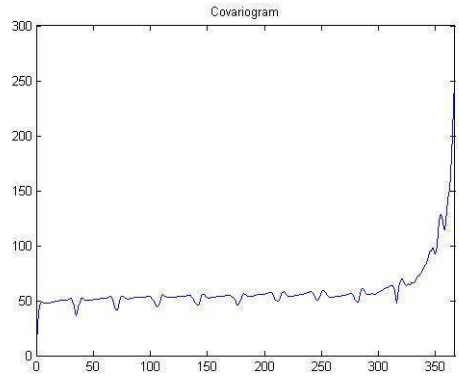


Fig. 3.12 d) The covariogram of b) computed with d_1 ⊞

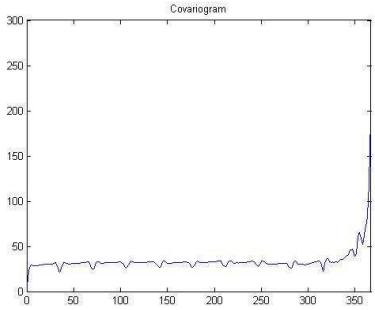


Fig. 3.12 e) The covariogram of a) computed with d_1

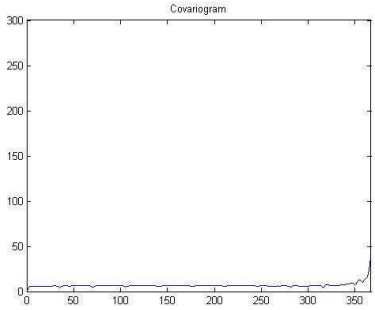


Fig. 3.12 f) The covariogram of b) computed with d_1

Bibliography of chapter 3:

[1] Jourlin M., Pinoli J.C., 1988, "A model for logarithmic image processing", J. Microsc., 149, 21-35

[2] Jourlin M., Pinoli J.C., 1995, "Image dynamic range enhancement and stabilization in the context of the logarithmic image processing model", Signal Process. 41 (2), 225-237

[3] Jourlin M., Pinoli J.C., 2001, "Logarithmic Image Processing: The mathematical and physical framework for the representation and processing of transmitted images", Advances in imaging and electron physics, 115, 129-196

[4] Brailean et al, 1991, "Evaluating the em algorithm using a human visual fidelity criterion", in: Proceedings of the International Conference on Acoustics. Speech and Signal Processing, pp. 2957–2960

[5] Pumo B., Dhorne T., 1998, "Modèle logarithmique de régression optique. Application à l'identification des fonctions de tranmittance", Revue de Statistique Appliquée, 46, 3, 65-75

[6] Jourlin M., Pinoli J.C., Zeboudj R., 1989, "Contrast definition and contour detection for logarithmic images", J. Microsc., 156, 33-40

[7] Carré M., Jourlin M., Bouabdellah M., 2011, "Additive contrast and associated metrics", ICS XIII (13th International Congress of Stereology, Pékin.

[8] Jourlin, M., Carré, M., Breugnot, J. and Bouabdellah, M., 2012, "Logarithmic image processing: Additive contrast, multiplicative contrast, and associated metrics", *Advances in Imaging and Electron Physics, P. W. Hawkes, Ed. Elsevier*, **171 (7),** 357 - 406.

[9] Matheron G., 1970, "La théorie des variables régionalisées et ses applications", Les Cahiers du Centre de Morphologie Mathématique, Fasc. ***5,*** Paris, E.N.S.M.P. 212 **p.**

[10] Inam Ul Haq M. and Jourlin, M., 2011, " Contribution of Logarithmic Tools To Texture Analysis", ICS XIII (13th International Congress of Stereology, Pékin.

Résumé Chapitre 3 :

Ce chapitre est entièrement consacré à des rappels sur le Modèle LIP et les derniers outils développés dans ce Modèle, en particulier de nouvelles notions de contraste (additif, multiplicatif) et les métriques associées à ces contrastes. L'intérêt du chapitre est double : on sait que le Modèle LIP est compatible avec la vision humaine, et les métriques logarithmiques apparaitront comme des outils capables de piloter la construction de covariogrammes. Des exemples de telles constructions sont donnés à la fin de cette partie, montrant en particulier la robustesse de l'approche proposée à des situations d'images sous-éclairées et/ou présentant des dérives d'éclairement.

Chapter 4 – Percolation, Fast Marching, Propagation front

4-1- Percolation

4-1-1- Principle of percolation

4-1-2- Application to image processing

4-1-3- Application to texture evaluation

4-2- Fast Marching Model

Bibliography of the Chapter

Chapter 4 – Percolation, Fast Marching, Propagation front

4-1- Percolation

4-1-1- Principle of percolation:

Let us begin with a definition of percolation.

The slow passage of a liquid through a filtering medium is called percolation. Thus percolation appears as a physical critical process to describe the transition of a system from one state to another.

It can also be interpreted as a phenomenon of transmission of "information" through a set of interconnected sites and links that can relay information to neighboring sites based on some status or properties these sites present.

Broadbent and Hammersley (1957) [1] introduced the „percolation model' to study how the anti-gas masks of the soldiers become in effective [2]. Consider a porous stone putted in a container of water. So what is the probability that the centre of the stone will be wetted? It is explained as following.

Let $L^2(Z^2, E^2)$, where Z represents vertices and E represents edges, be the plane square lattice and let p be some probability, which simply means that $0 \leq p \leq 1$.

We examine each edge of L^2 , and declare this edge to be „*open'* with probability „*p*' and „*closed*' otherwise, independently of all other edges. The edges of L^2 represent the inner passages of the stone, and the parameter p is the proportion of passages which are broad enough to allow water to pass along them. We think of the stone as being modeled by a large, finite subsection of L^2 (see Figure 4.1), perhaps those vertices and edges of L^2 contained in some specified connected sub-graph of L^2. By sinking the stone, a vertex x inside the stone is wetted if and only if there exists a path in L^2 from x to some vertex on the boundary of the stone, using open edges only.

Percolation theory is concerned primarily with the existence of such „open paths'. By deleting the closed edges only random sub-graphs of L^2 are left. The structure of the sub-graphs of L^2 is studied to point out the relation between the structure and the numerical value of the initial probability p. Importantly the interior passages of the stone are in notable

amount according to the overall size of the stone. So the probability that a vertex near the centre of the stone becomes wet because of the water diffusion from its surface, is similar to the probability that the vertex is the end-vertex of an infinite path of open edges in L^2. Thus it can be said that the large scale diffusion of water into the stone is due to the infinite connected clusters of open edges.

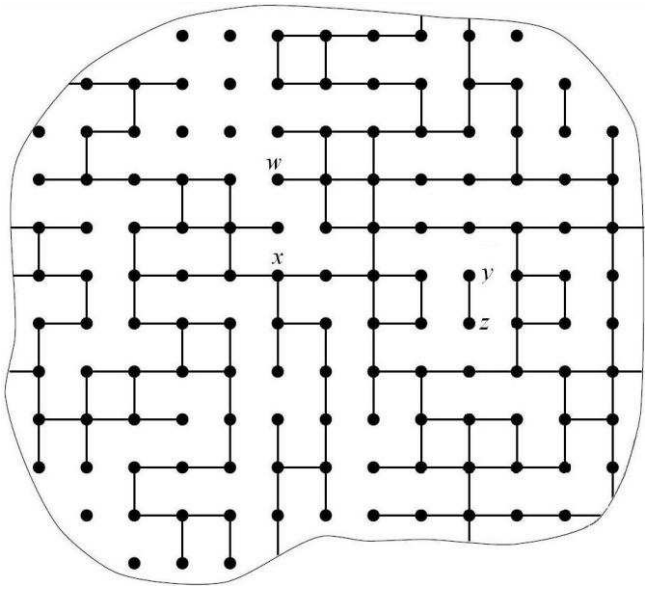


Figure 4.1. Structure of 2-D porous stone. Open edges are indicated by lines. Clearly vertex w and x can be wetted while y and z will remain dry while the stone is sunk in the water.

Percolation can be (see Fig. 4.2.):

i) **Bond Percolation:** A percolation which considers the lattice edges as the relevant entities.

Consider 2-dimensional lattice plane L^2, where set of edges E^2 connect sites x, y = ($x_1, x_2, x_3, \ldots\ldots x_n$, $y_1, y_2, y_3, \ldots\ldots, y_n$) located at the vertices Z^2 for which the distance, defined as

$$d\ x, y\ =\ \sum_{i=1}^{n} x_i - y_i$$

does not exceed the maximum value. Thus the edges connect the adjacent vertices. It is noteworthy that an edge is open with probability p and closed otherwise independent of all other edges.

ii) **Site Percolation:** A percolation which considers the lattice vertices as the relevant entities. In site percolation, each vertex of L^2 is open with probability p

and closed with probability $1-p$ and is independent of other vertices. A path is *open* if and only if all the vertices are open.

Every bond percolation problem can be treated as a site percolation (graph varies) while the vice versa is false.

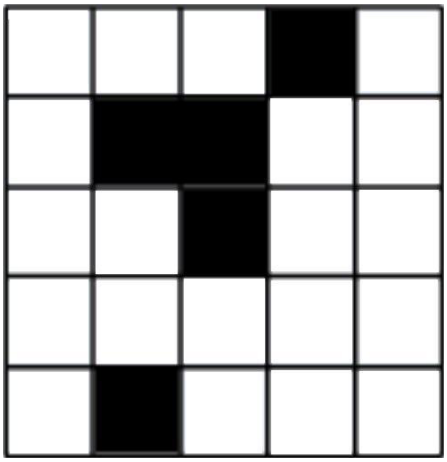


Figure 4.2a)Site Percolation

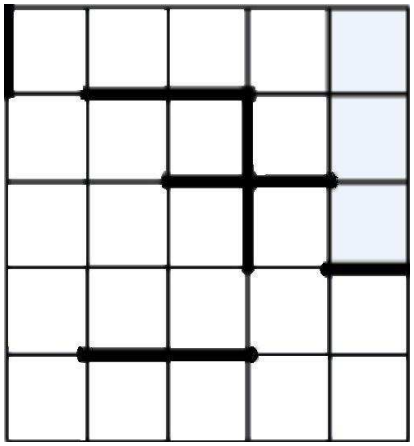


Figure 4.2 b)Bond Percolation

Now let us look at some applications of percolation:

There are plenty applications of percolation in many diverse situations. We can give as examples:

1). Network stability and fragility. Percolation theory can be used to determine the number of random nodes to be removed from a network so that its connectivity is lost (cf D. Callaway et al ([3])).

2). Oil fields: Percolation theory is used in predicting oil or gas distribution in the porous rocks of oil fields (cf. D. Stoffer ([4])). P R King et al applied percolation theory to estimate the time for a fluid injected into an oilfield to breakthrough into a production well [5]

3). Conductors: Consider a regular body built with tiny particles of two materials, A (conductor) and B (insulator). Assuming that electric field is applied over the opposite sides of the body, we suppose that each particle of the body is chosen at random to be either of type A or type B. In such conditions, a percolating cluster of neighboring particles of type A between the two electrodes can result in producing finite resistance phenomenon, while the absence of such cluster will result in infinite resistance. This is dependent on the ratio of type

A particles, represented by probability p_a. There exists a critical probability $p_c \approx 50$, such that $p_a > p_c$ the resistance is finite otherwise infinite [2].

4-1-2- Application to image processing:

Bond percolation based model for image segmentation is used by Iftekhar H. et al [6], [7]. The terms pixel and site are used interchangeably if an image is defined on a square lattice with one-to-one correspondence between the pixels and the sites. A threshold is defined, and a bond is assumed to exist between a site and its nearest pixels until the threshold is not crossed.

To form a bond-percolation lattice for an image I with size NxM, the following two parameters are defined

$$p \triangleq \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} e^{-\frac{\Delta I(i,j)}{\Delta I}}$$

and

$$f(i,j,l,m) \triangleq \frac{e^{-\frac{\Delta I(i,j)}{\Delta I}} + e^{-\frac{\Delta I(l,m)}{\Delta I}}}{2}$$

where $N \times M$ is the image size, $\Delta I(i,j)$ is the magnitude of the gradient at pixel (i,j) and ΔI is the average gradient magnitude. So here the probability p can be considered as the threshold. Being calculated the values of p and f the corresponding bond-percolating lattice is formed by finding a bond to be present between a pixel i,j and its neighbor l,m if and only if $f < p$.

From the definition of p and (i,j,l,m), it's clear that the bond exists between neighboring pixels at i,j l,m if they have high gradients like contrast edges or textures areas. Yamaguchi and Hashimoto [8] used a method of scalable window processing based on percolation model that takes into account the connectivity between the focal pixels and the neighborhood. This method is proved to be effective for various tasks like crack detection, noise reduction and edge detection.

4-1-3- Application to texture evaluation:

In our knowledge, it is not classical to use the notion of percolation to evaluate and classify textured images.

Nevertheless, we can refer to the PhD thesis of François MAYET and Gabriel FRICOUT, whose references are listed in Chapter 5 of the present work, in which we propose a first experiment aiming at classifying randomly textured images. The method consists of associating to percolation trajectories inside a studied image some parameters like length, fractality…and performing well-known classifications algorithms on the retained parameters.

4-2- Fast Marching Model

Fast Marching Method (FMM) is introduced by J.A.Sethian [9]. It is a numerical technique designed to track the evolution of Interface. FMM is very efficient for certain specialized front problems. It handles the problems in which separating interfaces develop sharp corners and cusps, change topology and become truly intricate.

Consider an interface, separating two regions from one another, and a speed F which indicates the motion of each point of the interface. In the figure 4.3., a green dotted line separates the inner region and the external surrounding region (yellow), and at each point of the dotted green line the speed F is given (represented by vectors orthogonal to the green line). Moreover, let us suppose that the speed F is always positive, which means that the front always moves outwards.

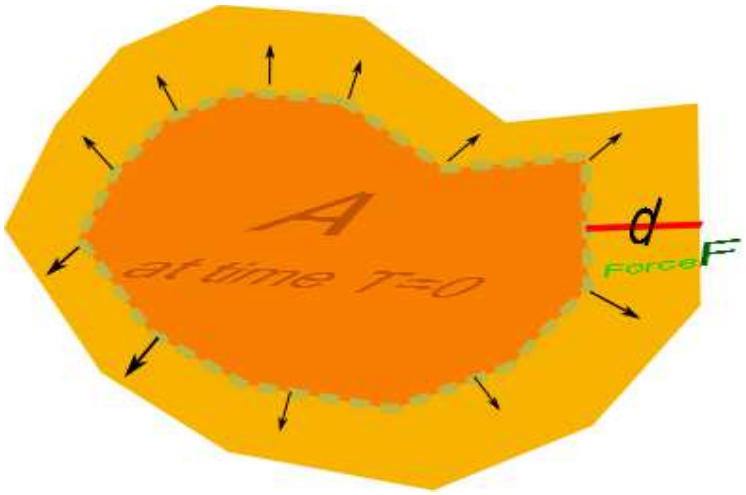


Fig 4.3. Representation of a propagation front

Suppose that body A, at time T=0, is bounded by dotted line, and is tending to expand in the outside direction in the yellow area. It can be thought of any phenomena like water waves (which should be circular and regular) or earth quack expanding its region of influence based on certain different factors.

Rather than follow the interface itself, the Fast Marching Method makes use of stationary approach to the problem. At first glance, this sounds counter-intuitive; we are going to trade a moving boundary problem for one in which nothing moves at all! To see how this is done, imagine a grid laid down on top of the problem:

Fast Marching method uses stationary approach to the problem. This apparently sounds quite unreasonable to use the moving boundary problem in a case where there is nothing moving. Consider the following example of Fig 4.4:

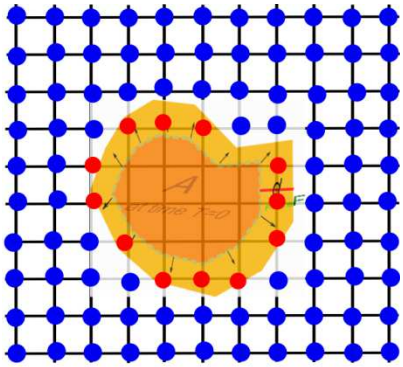


Fig 4.4 Applying a regular square grid over the propagation situation

Let there is a sensor and timer at each grid point (red and blue where red are more probable to be effected by the propagating front). The time T at each grid point is noted, when the front crosses it. Thus a function T can be defined, which associates the time $T(x_i, y_j)$ to each grid point (x_i, y_j).

As an example, suppose the initial disturbance is a circle propagating outwards. This circular region, in the following example, has outwards propagation tendency and may it cross the blue points. The following cone -shaped surface in figure 4.5.b) is given by the function $T(x_i, y_j)$, and it is noteworthy that this surface has an important property: it intersects the $xy-$ plane exactly where the curve is initially. This conned shaped surface looks expending with time T if the surface propagates. In other words at any value of times T, it is possible to

determine the set of points reached at. Hence Figure 5 b gives the arrival time so it is **arrival time surface**.

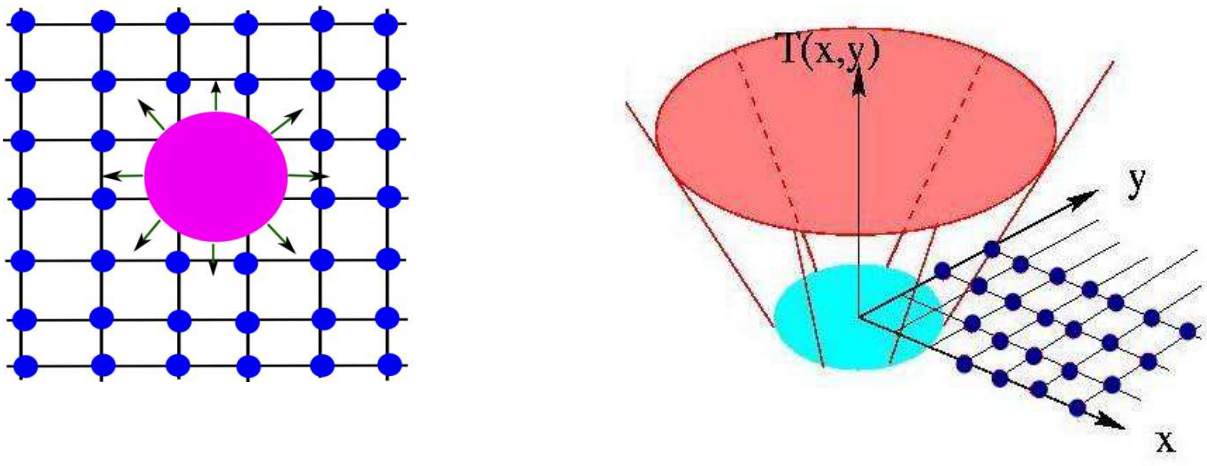


Fig 4.5. a) and b) Example of circular propagation and arrival time surface.

It is also called binary value formulation as let us suppose the initial position of the front be the boundary for this arrival time surface T(x,y) that we would like to find. The actual concept is to find something that not stationary according to time T, thus redefining the stationary problem in which the arrival surface carries the information about what is non-stationary position.

Thus the boundary value formulation is given by the Eikonal Equation as given below:

$$\nabla T \; F = 1.$$

where ∇T is the gradient of T.

Using an approximation to the gradient, the *Eikonal* equation may be discretized as:

$$[\max(D_{ij}^{-x}T,0)^2 + \min(D_{ij}^{+x}T,0)^2 + \max(D_{ij}^{-y}T,0)^2 + \min(D_{ij}^{+y}T,0)^2 +] = 1/F^2,$$

where $T(x,y,0)=0$,

$D_{ij}^{-x}T=(T_{ij} - T_i\text{-}1,j)/(x_i - x_i\text{-}1)$, $D_{ij}^{+x}T=(T_i+1,j - T_{ij})/(x_i+1 - x_i)$,

$D_{ij}^{-y}T=(T_{ij} - T_{i,j}\text{-}1)/(y_j - y_j\text{-}1)$ and $D_{ij}^{+y}T=(T_{i,j}+1 - T_{ij})/(y_j+1 - y_j)$.

A less diffusive approximation to the gradient is (this will be implemented):

$$[\max(\max(D_{ij}^{-x}T,0), -\min(D_{ij}^{+x}T,0))^2 + \max(\max(D_{ij}^{-y}T,0), -\min(D_{ij}^{+y}T,0))2\ ] = 1/F^2.$$

This equation is implemented over grid points which are of the following three types.

- **Alive Points** (black points) are points where values of T are known.
- **Trial Points** (green) are the closer points to alive points curve and the propagation is to be computed for it, and its set is called *narrow band*. The points from narrow band join the alive points as propagation takes place and the narrow band spreads.
- **Far Away Points** (blue points) are points which are outside of the narrow band and could be the part of it ie can be transformed into trail points later.

It's important to observe that the propagation occurs from smaller to greater values of T. Figures 4.6 *a) – f)* explain this idea: in figure 4.6 *a)*, the black point (alive) represents the initial curve; in figure 4.6 *b),* the value of T is computed in the neighborhood of black point; this neighborhood is converted from far away (blue) to trial points (green); in figure 4.6 *c)* the trial point with smallest value of T is chosen (for example "A"); in figure 4.6 *d),* values of T are computed in the neighbors of point A, converting them from far away to trial. In figure 4.6 *e),* the trial point with smallest value of T is chosen (for example, "B"); in figure 4.6 *f),* the neighbors of B are converted from far away to trial. And so on.

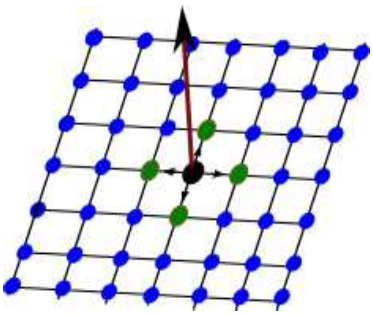


Figure 4.6: a) Initiating from alive point (black)

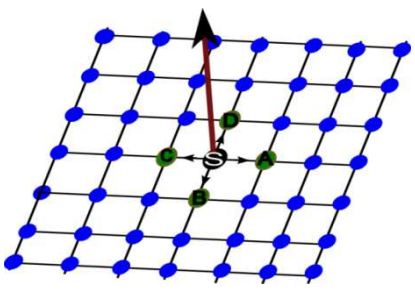


Figure 4.6 b) Updating „Upwind „ and compute possible values for neighbors

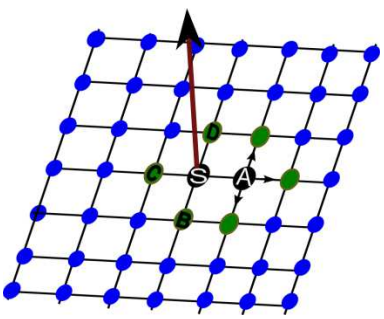


Figure 4.6c) Choose the minimum value of (here "A").

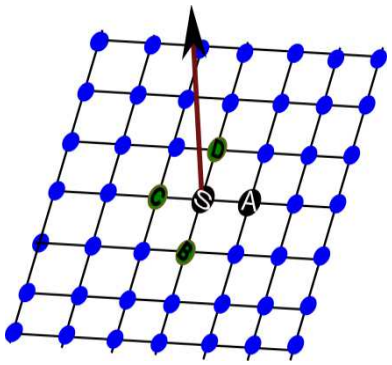


Figure 4.6 d) Compute values of T around of point A, converting them from far away to trial.

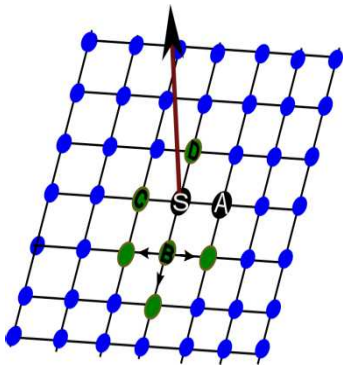


Figure 4.6e) Choose the smallest value of T is chosen ("B" here);

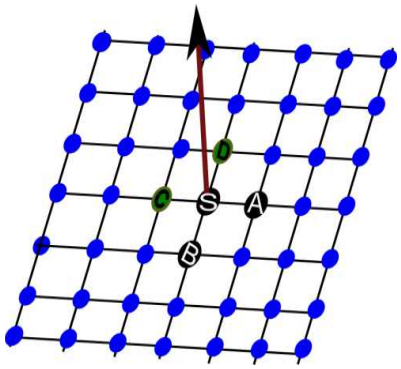


Figure 4.6f)Update the neighbors of B by converting from far away to trial

It is important to note that to update a point only the adjacent points with minimum value are taken into account or more precisely the 4-neighboradjacency with smaller value of T.

Basic Algorithm

The basic algorithm to fast marching method is:

- Initialization Step

1. Alive Points:

   Let A be the set of all grid points $\{i_A, j_A\}$ that represents the initial curve;

2. Narrow Band:

   Let *NarrowBand* be the set of all grid points neighbors of A.

3. Far Away Points:

   Let *FarAway* be the set of all others grid points $\{i,j\}$. Set $T_{i,j} = \infty$ for all points in FarAway.

- Marching Forwards

1. Begin Loop: Let (i_{min}, j_{min}) be the point in NarrowBand with the smallest value for T.
2. Add the point (i_{min}, j_{min}) to A; remove it from NarrowBand.
3. Tag as neighbors any points (i_{min}-1, j_{min}), (i_{min}+1, j_{min}), (i_{min}, j_{min}-1), (i_{min}, j_{min}+1) that are either in NarrowBand or FarAway. If the neighbor is in FarAway, remove it from that list and add it to the set NarrowBand.
4. Recompute the values of T at all neighbors according to discrete Eikonal equation, selecting the largest possible solution to the quadratic equation.
5. Return to top of Loop.

Initialization

Given an initial implicit curve $C = \{f(x,y)=0\}$ the initialization step is performed by evaluating the function f in all grid points (i,j). The curve C occurs between grid points where the function f changes the signal. The values of $f(i\Delta x, j\Delta y)$, where Δx and Δy are respectively the horizontal and vertical spacing of a regular grid, will be used to calculate initial T values for alive and narrow band points.

First all grid points T values are set to ∞ and they are classified as far away. The value of the function f is calculated in each grid point (i,j). If $f(i\Delta x, j\Delta y) \leq 0$ and $f>0$ for any neighbor $(i+1,j)$, $(i-1,j)$, $(i,j+1)$, $(i,j-1)$, then (i,j) is classified as alive and its neighbors where $f>0$ are classified as trial. The T value may be computed as follows:

Suppose $f((i+1)\Delta x, j\Delta y) > 0$, then $T_{rig} = \Delta x \, f(i\Delta x, j\Delta y) / [f(i\Delta x, j\Delta y) - f((i+1)\Delta x, j\Delta y)]$.

Supose $f((i-1)\Delta x, j\Delta y) > 0$, then $T_{lef} = \Delta x \, f(i\Delta x, j\Delta y) / [f(i\Delta x, j\Delta y) - f((i-1)\Delta x, j\Delta y)]$.

Supose $f(i\Delta x, (j+1)\Delta y) > 0$, then $T_{top} = \Delta y \, f(i\Delta x, j\Delta y) / [f(i\Delta x, j\Delta y) - f(i\Delta x, (j+1)\Delta y)]$.

Supose $f(i\Delta x, (j-1)\Delta y) > 0$, then $T_{bot} = \Delta y \, f(i\Delta x, j\Delta y) / [f(i\Delta x, j\Delta y) - f(i\Delta x, (j-1)\Delta y)]$.

The resulting T_{ij} value may be computed as

$T_{ij} = \max(T_{rig}, T_{lef}, T_{top}, T_{bot})$.

As T_{rig}, T_{lef}, T_{top} and T_{bot} have negative values, T_{ij} is the smaller absolute value among them. Figure 4.7 *a)* shows this calculation. That's obvious that the situation where the four neighbors are positive is an exception, where the initial front is a single point.

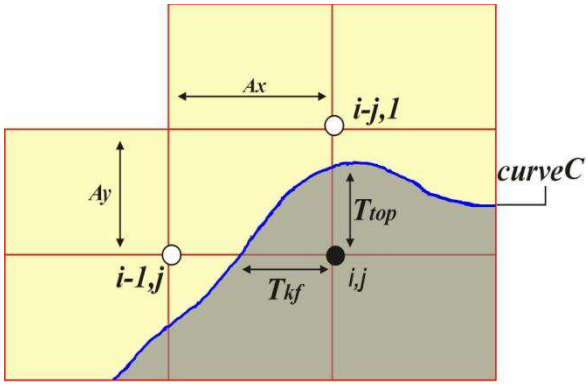


Fig 4.7 a) Calculating T(i,j)

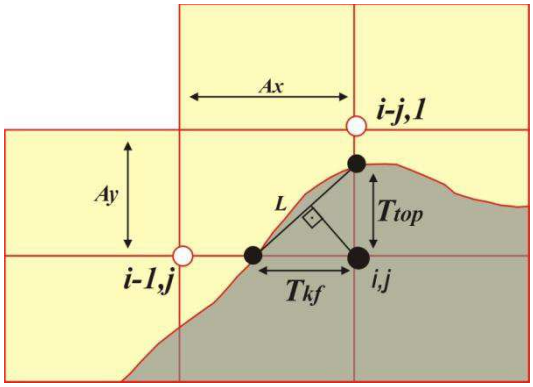


Fig 4.7 b:Smallest values of T(i,j) is calculated by using from line L.

The values of the trial points are computed in the same way.

A more precise way to calculate the T_{ij} value is shown in Fig 4.7 b, where the values T_{lef} and T_{top} are used to define a line L. The value T_{ij} is computed as the distance of point (i,j) to line L. That's important remember that in the evolution considered above the speed F>0, so the front will evolve outward. If the front would evolve inwards, it wouldn't be considered the signed distance in for the initial alive points.

The above computation considers that we have an implicit function f. In many situations we don't have such function, but only a set of values defined in the grid points (i,j), and given an iso-value c (that represents the curve C) we'd like to propagate from points with value c. In this case the initialization process is similar to the case where we have $C = \{f(x,y)=c\}$.

Marching Forward

The step 4 of the marching forward algorithm tells that the values of T at all neighbors of (i_{min},j_{min}) are recomputed according to discrete Eikonal equation

$$[\max(\max(D_{ij}^{-x}T,0), -\min(D_{ij}^{+x}T,0))^2 + \max(\max(D_{ij}^{-y}T,0), -\min(D_{ij}^{+y}T,0))2\ ] = 1/F^2,$$

selecting the largest possible solution to the quadratic equation. The upwind difference of such equation indicates that information propagates from smaller values to bigger values of T. For simplification consider $1/F = g$.

In Figure 4.8, consider that the value T_X of the point X must be computed. Consider that the T_X value is computed from point A (containing the smallest T value among points B, C and D).

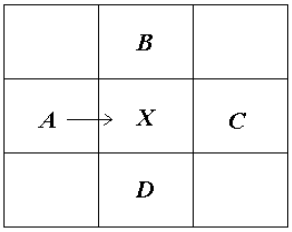


Fig 4.8 : A contains the smallest value T (out of A,B,C,D).

The above equation says to consider only the biggest between the forward and backward differences (in both directions). So in such computation we only use the smallest between T_B and T_D values and the smaller between T_A and T_C values. There are basically two situations to be considered in computing the T_X value:

1. If $T_A + g \geq min\ (T_B, T_D)$ (for simplification, consider $min\ (T_B, T_D) = T_B$)

In this case T_X value may be computed from equation:

$$(T_X - T_A)^2 + (T_X - T_B)^2 = g^2 .$$

Because of condition $T_A + g \geq T_B$, the discriminant has a non-negative value.

To prove that the T values are always crescent, we should find that $T_X \geq T_A$, as bellow:

The solution of equation is $T_X = [T_A + T_B + (2g^2 - (T_B - T_A)^2)^{1/2}] / 2$.

As $g \geq (T_B - T_A)$, so $T_X \geq [T_A + T_B + (2(T_B - T_A)^2 - (T_B - T_A)^2)^{1/2}] / 2$, and this results that $T_X \geq T_B$. As T_A is the smallest among T_B, T_C, and T_D, so $T_X \geq T_A$, what proves the monoticity of T.

2. If $T_A + g \leq min\ (T_B, T_D)$

In this case the discriminant has a negative value and T_X value is computed from:

$T_X = T_A + g$. As $g>0$, so $T_X \geq T_A$.

Realize that in both cases above if the T value of any point B,C, D is far away, it won't be considered in the calculation of T_X.

Kimmel and Sethian ([10]) give a technique for computing geodesics on manifolds using the Fast Marching Method. Tao X. et al used the FFM to extract weighted geodesic curves that have desired global properties [11] and experimented it on a simulated image. The simulated image is a gray level image with some bright structures (Fig 4.9). In the image we specified the end points of the desired curve. If the speed function is set to be the intensity value, the resulting curve will run through regions with high intensity values as much as possible. This is the upper curve in the figure. If instead, the lower curve is desired, without constraints on the global appearance of the curve, we need to specify some intermediate points (circles in the figure).

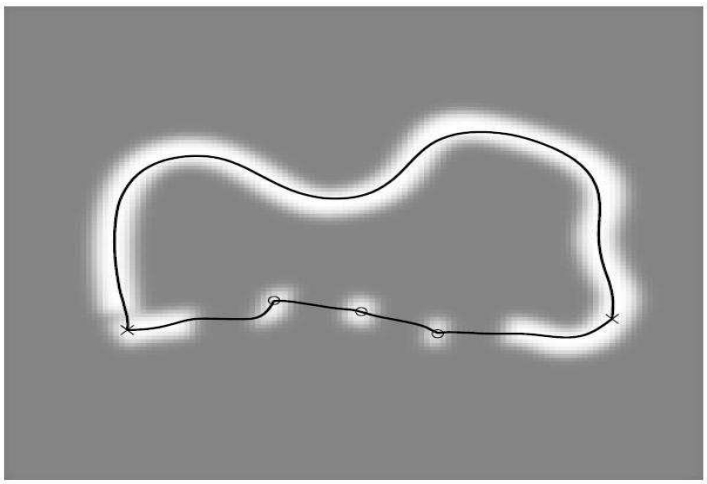


Fig 4. 9 simulated image is a gray level image with some bright structures to extract weighted geodesic curves that have desired global properties as discussed above.

N. Forcadel et al used the Generalized Fast Marching method, an extension of FMM, for image segmentation ([12]). The Fast Marching Method (FMM) is useful when tracking a propagating front that evolves with a constant sign speed, the GFMM offers more general assumptions on the velocity and in particular no sign restrictions. A second order version of

the Fast Marching Method on triangulated domains to compute shape offsets on machine parts is given by Sethian and Vladimirsky in [13].

Bibliography of chapter 4:

1. S. R. Broadbent and J. M. Hammersley (1957). Percolation processes. Mathematical Proceedings of the Cambridge Philosophical Society, 53, pp 629-641.
2. G. Grimmett: Percolation, Second édition, Springer-Ver lag, Berlin, 1999,
3. Duncan Callaway, M. E. J. Newman, Steven H.Strogatz, and Duncan J. Watts, Network robustness and fragility: Percolation on random graphs, Phys. Rev. Lett. 85, 5468-5471 (2000).
4. Stauffer D., Arahony A., Introduction to Percolation Theory, Taylor & Francis, 2001
5. King PR, Buldyrev SV, Dokholyan NV, et al, Using percolation theory to predict oil field performance, Meeting on Horizons in Complex Systems, 2002, Pages:103-108
6. Hussain I, Reed TR. Bond percolation-based Gibbs-Markov random fields for image segmentation. *IEEE Signal Processing Letters*, 2(8):145–147, August 1995.
7. Hussain I, Reed TR A bond percolation-based model for image segmentation. IEEE Trans Image Process. 1997; 6(12):1698-704.
8. Yamaguchi T, Hashimoto S. Image processing based on percolation model. *IEICE Transactions on Information and Systems* 2006; E89-D(7):2044–2052.
9. JA Sethian, Advancing interfaces: level set and fast marching methods, ICIAM, 1999.
10. Kimmel, R., and Sethian, J.A., Fast Marching Methods on Triangulated Domains, Proc. Nat. Acad. Sci., 95, pp. 8341-8435, 1998.
11. Tao X, Davatzikos C, Prince JL. Using The Fast Marching Method To Extract Curves With Given Global Properties. MICCAI 2005, LNCS 3750, pp. 870–877, 2005.
12. N. Forcadel, et al, Generalized Fast Marching Method : Applications to Image Segmentation, Numerical Algorithms, 48(1-3) :189-211, 2008.
13. Sethian, J.A., and Vladimirsky, A., Extensions of Fast Marching Methods: Higher Order, Static Hamilton-Jacobi Equations and Control, submitted for publication, Proc. Nat. Acad. Sci., Oct. 1999.

Résumé Chapitre 4 :

Dans le chapitre 4, nous introduisons les outils qui vont nous permettre d'aborder l'étude des images à texture aléatoire. La notion de percolation est rappelée de manière détaillée, ainsi que la construction de front de propagation qui en découle et le concept de fast marching qui est un moyen de sa mise en œuvre.

Chapter 5 – Application of percolation to textured images classification

Chapter 5 – Application of percolation to textured images classification

5-1- Introduction:

Considering that the concept of percolation is not very familiar in image processing, let us recall a possible general definition of it.

The free encyclopedia "Wikipedia" gives a very interesting definition of "Percolation" of which we are going to quote here:

Percolation is [...] a threshold phenomenon, associated with the transmission of "information" through a network of sites and links, depending on their condition, to or not to relay information to neighboring sites. The term percolation comes from the similar phenomenon like the passage [...] of water through a percolator, the filter of a coffee machine, to make coffee.

Our approach is an attempt to answer the following questions.

- Is it possible to study the propagation of a fluid through a percolating medium to extract sufficient features which allow identifying the medium?
- In other words, consider in image processing (as we will apply this idea in it):

Is it possible to extract sufficient parameters of the percolation trajectories to characterize and identify a texture?

Such an approach has been previously used by F. Mayet ([1]) and suggested by G. Fricout ([2]) in their respective theses. Nevertheless it seems that a very few literature is dedicated to this idea.

The following Figure 5-1 a), b), c) shows the approach we have presented considering that the percolation is directed from the top to the bottom of the image (this direction is identical to a gravitational field in the case of physics).

Figure 5-1 a) represents the considered initial image, which is extracted from the very classical Brodatz database of textured images.

We apply to this image a percolation algorithm (explained in detail later) from top (first line of the image) to bottom (last line of the image).

Thus, each pixel of the first line moves inside the image according to a percolation trajectory. Some of these trajectories are superimposed to the initial image, which is shown on Fig 5.1.b).

Finally, these trajectories are represented separately on Fig 5-1 c) to facilitate their readability.

Fig 5-1 a) Original Image from Brodatz database

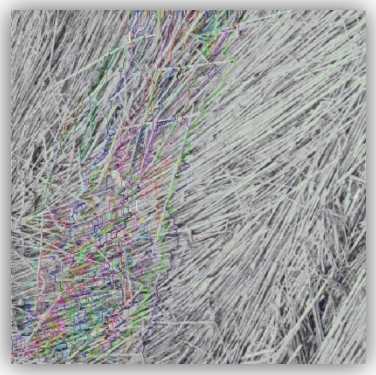


Fig 5-1 b) Original image and its percolation trajectories

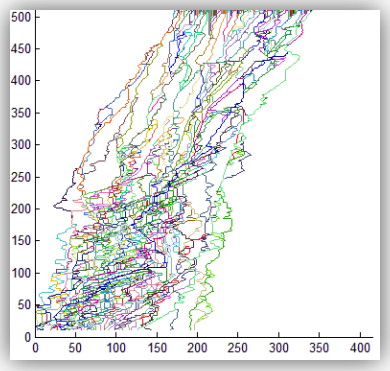


Fig 5-1 c)Percolation trajectories only

Fig 5-1 Percolation trajectories computed on a textured image

5-2- Implementation of percolation:

This section has been realized in collaboration with Bassam Abdallah, which was spending his End Studies Project in the Laboratory Hubert Curien and is now a PhD student in the Centre de Morphologie Mathématique (Ecole des Mines de Paris). We were particularly interested in the characterization of the so-called random textures, considering that for pseudo-periodic textures, many interesting tools have been developed and are functional at the moment.

Initially, we select a number of points on the starting line (in our study the first line of the image, but any other line can be chosen as starting line). In order to initialize the percolation process, we have chosen to consider middle/second third of the points as focus

points in the first line to avoid problems of percolation towards unintended edges which could distort the results (see Fig 5-2 below).

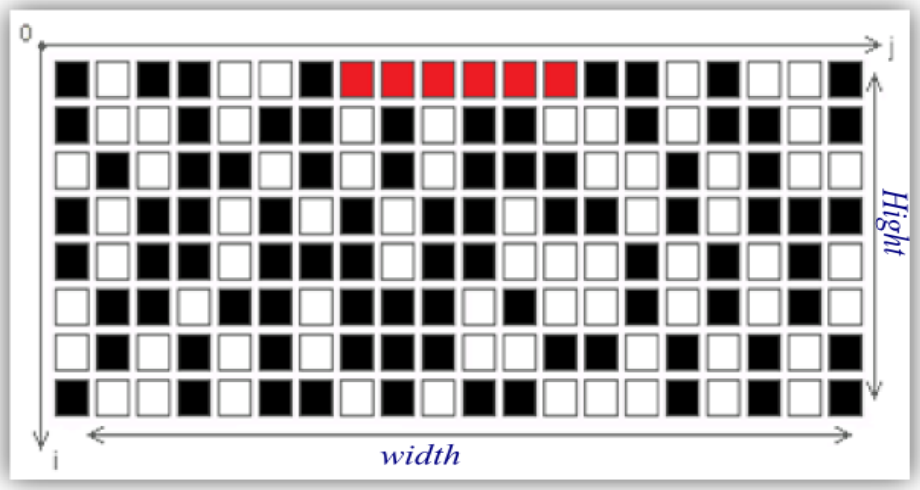


Fig 5-2 Red points are selected as starting points for percolation in a binary Image.

Once the starting points are selected, we perform the rest of the process until the end line of the image is reached.

We select one by one each of the current (red) points as we are interested in a portion of its neighborhood V8 (green in Fig 5-3 given here). Since we are working on a binary image (another version of percolation grayscale was also implemented but we do not present it here) we will try to follow the contours of extracted patterns in the preprocessing. It is simply to follow as possible pixel value 1 (white as in Fig 5-3).

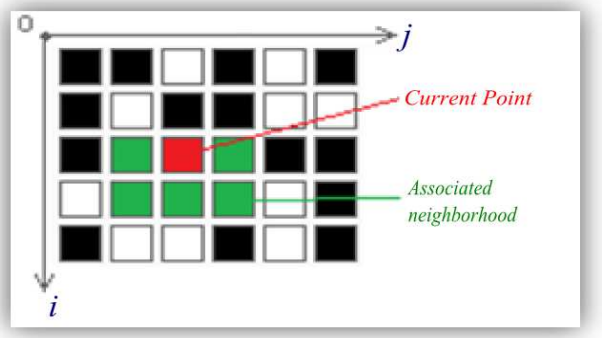


Fig 5-3 Useful part (in green) of the neighborhood of the studied pixel (in red)

For each of these neighboring points, a parameter is computed (which can be driven by a grayscale LUT), which here is simply the value of the pixel. Then we will select the neighboring point to maximize the selected parameter (in our case the lightest point).

Note 1: In order to overcome the problem of multiple points with the same parameter value, we set up a random selection from among the best candidate.

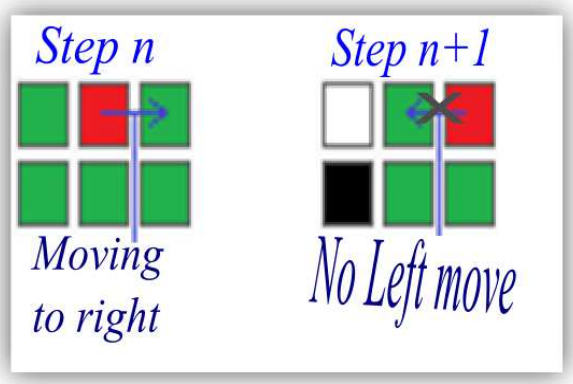


Fig 5-4 Inertia rule in order to avoid oscillations

Note 2: To avoid oscillations (see Fig 5-4), as soon some/one direction is chosen (on the line from the current point) to avoid/prohibit the other direction. Indeed this is understandable in terms of inertia in physics: if some liquid chooses one direction at a step n it will not come back on itself in step $n+1$, in our model we allow the other direction once the coordinate of the current point i are increased.

Note 3: In the case where all the neighboring pixels are zero we must perform a vertical movement downwards (this avoids having to create random movements that would disturb to extract parameters (see the following section 5-3: "Percolation curves and parameters extraction")

At the end of the process, the trajectory of each initial pixel has been built (cf. Fig 5-5).

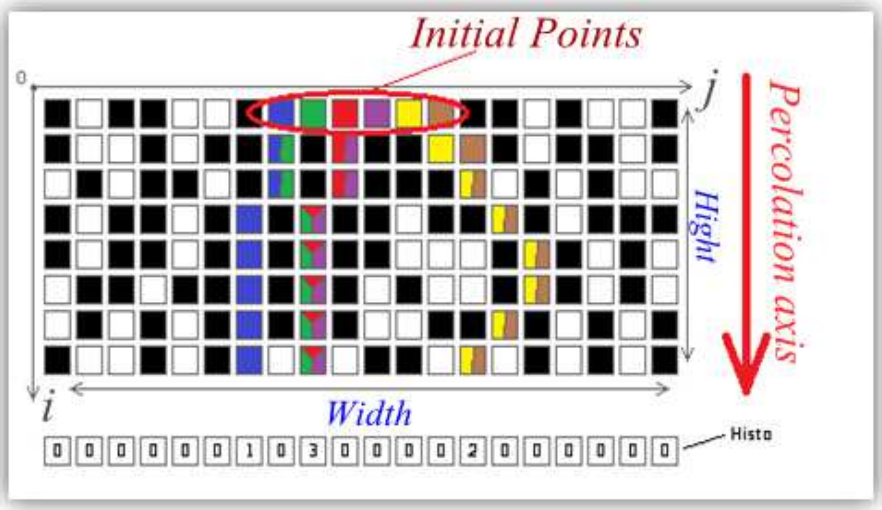


Fig 5-5 Result of the percolation of the example presented in Fig 5-2

5-3- Percolation curves and parameters extraction

Once the percolation process is over, that is to say, when each of the starting points arrives to the bottom of the image, we obtain a multitude of trajectories (see Fig 5-1 c)) that we must now examine.

We first tried a multitude of parameters that can be used in the study of these trajectories. After study of the influence of these, we have selected the most significant parameters in a PCA, which allows us to calculate the weights of the various chosen parameters. This study was conducted with the Brodatz base of textured images. The parameters selected are those producing the best separation of different textures and at the same time bringing together the best textures of the same nature.

Our approach is exploratory and does not aim at deeply justify the selection of retained parameters. It is the reason why we have limited ourselves to the extraction of 9 parameters, listed hereafter.

Histogram:

This variable is not used directly as a parameter, but the parameter MeanHisto is derived from it. This variable is a vector (1 x m), where m is the image size, in the direction perpendicular to the percolation axis (in this case the width). Variable *Histo* at each index j

represents the number of trajectories arriving in column j (in the case of a vertical percolation).

Mean Histogram:

This parameter represents the average offset d (that is to say, on the axis orthogonal to the axis of the percolation) of trajectories. It is calculated by taking the weighted average of the columns (the weight of a column j corresponding to the number of curves approaching this column's end). Thereafter we subtract the average of the column values initially chosen (variable Windows). Thus we obtain the average displacement through our connected curves (percolation candidates) from the departure point to the end point.

In addition, as we want to have a variable independent of the initial image size, these results are normalized i.e. divided by the width *m* of the image. If the size of initial image I was n * m, the following result holds:

$$MeanHisto = \frac{1}{m} * \sum_{j=1}^{m} (Histo\ j\ * j - Window\ j\ * j)$$

The other selected parameters are:

L: probability at a given moment, a droplet at the position (i, j) arrives at (i, j-1) at next step.

R: probability at a given moment, a droplet at the position (i, j) arrives at (i, j +1) at next step.

BR: probability at a given moment, a droplet at the position (i, j) arrives at (i +1, j +1) at next step.

BL: probability at a given moment, a droplet at the position (i, j) arrives at (i +1, j-1) at next step.

B: probability at a given moment, a droplet at the position (i, j) arrives at (i +1, j) at next step.

MeanFractality: Average fractal dimension calculated for each observed droplet trajectory. The fractal dimension of droplets trajectories is computed using the method "boxes", hereafter recalled.

Recalls about Fractal Dimension

The state or condition of being Fractal is called fractality.

Fractals are self-similar structures where the whole has the same shape as its parts e.g., broccoli (Fig 5-6a) or the Koch snowflake (Fig 5-6b).

a) Broccoli

b) Koch Snowflake

5-6: Fractal examples

Mandelbrot ([3], [4]) has defined fractals as the object having self-similarity. Fractals are geometrical objects with fractional dimensions. Fractality also addresses extremely irregular curves or shape for which any suitably chosen part is similar in shape to a given larger or smaller part when magnified or reduced to the same size.

"Fractal is defined as an object or a quantity which depicts the self-similarity in certain senses on various or all scales". But it is important to note that there should be some similarity of structure, at least, instead of demonstrating the exact similarity among the sub-objects inside the object. And each complete sub-object has almost all the information like the whole object.

Since the geometrical objects/shapes live in spaces with dimensions greater than or equal to that of objects so Fractals are the objects (shapes or curves) having dimensions in fraction".

Following are few methods of calculating fractal dimensions.

Hausdorff - Besicovitch dimension:

Before Mandelbrot, Hausdorff and Besicovitch have defined fractality and fractal dimension. The mathematical definition of Hausdorff or Hausdorff-Besicovitch dimension (Hausdorff [5], Besicovitch [6], [7], [8]) is based on the following definition of α-dimensional measure.

Let (E, d) be a metric space and ε a real number strictly positive. Consider an overlapping of E by a finite union of subsets $A_1 \ldots A_n$ of E, whose diameter $\delta(A_i)$ is less than ε. In such conditions, for a given ε and a given exponentiation α, the measure of the smallest overlapping is noted $m_{\varepsilon,\alpha}(E)$:

$$m_{\varepsilon,\alpha}(E) = \text{Inf} \{ \sum [\delta(A_i)]^{\alpha}, \text{ for } E \subset \text{union}(A_i) \text{ with } \delta(A_i) \leq \varepsilon \}$$

The α-dimensional measure $m_{\alpha}(E)$ of E is the limit value of $m_{\varepsilon,\alpha}(E)$ when ε tends to 0:

$$m_{\alpha}(E) = \lim_{\varepsilon \to 0} m_{\varepsilon,\alpha}(E)$$

Let (E, d) be a metric space. Let m_α be its α-dimensional measure. The Hausdorff dimension is the unique value $dim_H E$ such that

$$\begin{aligned} m_\alpha\ E &= \infty \ if\ 0 \leq \alpha < dim_H E \\ m_\alpha\ E &= 0 \ if\ dim_H E < \alpha < \infty \end{aligned} \qquad \ldots (4.1)$$

The Hausdorff dimension $dim_H E$ appears thus as the value for which the α - dimensional measure makes a "jump" from zero to infinity. This definition is difficult to implement, hence other methods which does not involve the notion of measure have been developed.

Minkowski- Bouligand Dimension:

"Minkowski sausage" of an object E is the set of points located at a distance less than ε of it, which means the dilation of E by a ball of radius ε.

Let $A(\varepsilon)$ be the area of this sausage. The Minkowski- Bouligand Dimension $\Delta_{MB}\ E$ (Bouligand [9], [10], Minkowski [11]) is defined according to

$$\Delta_{MB}\ E = \lim_{\varepsilon \to 0} \frac{\log \frac{A\ \varepsilon}{\varepsilon^2}}{\log \varepsilon} \qquad \ldots (4.2)$$

and can be deduced from the slope of a *log-log plot as given in the figure 5-7*

.

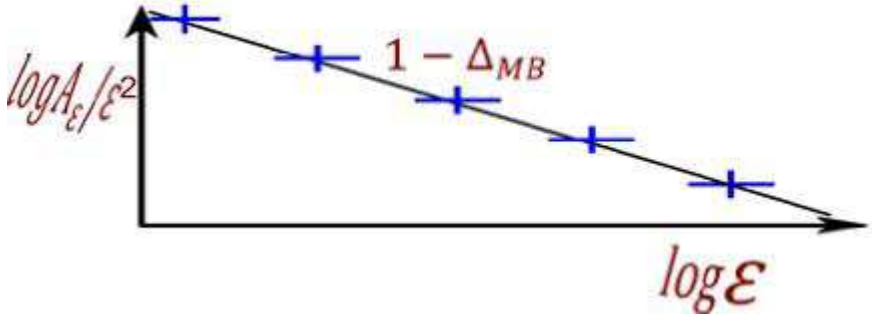


Fig 5-7: Estimation of Minkowski- Bouligand Dimension.

Dimensions by Box counting:

This method is also called grid method. The spatial domain is covered with a square grid consisting of k^2 contiguous boxes of size $\varepsilon = \frac{1}{k}$ (Figure 5-8).

Let $N(\varepsilon)$ be the number of boxes intercepted by the studied object. The dimension of the boxes is given by

$$\Delta_B = \lim_{\varepsilon \to 0} \frac{\log N(\varepsilon)}{\log \varepsilon} \qquad \ldots (4.3)$$

It can be demonstrated that the union of intercepted boxes constitutes an approximation of the Minkowski sausage (Bouligand [10]) and thus this method is identical to that of Bouligand-Minkowski (Tricot [12]).

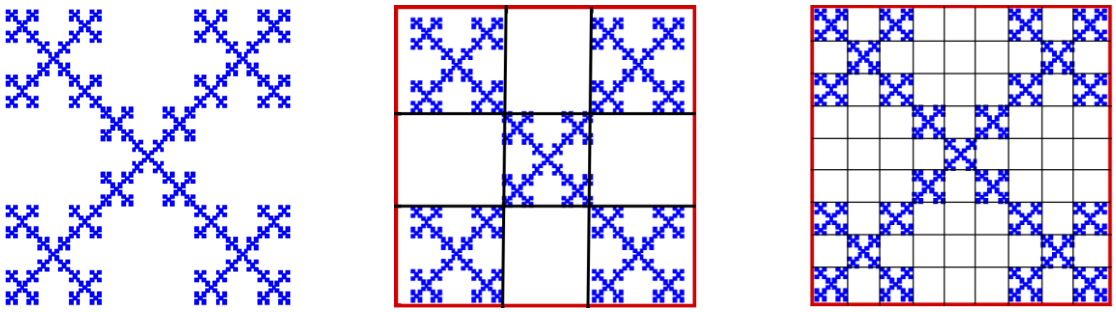


Fig a) :Object in one box. *Fig b): Object in* 3×3 *boxes.* *Fig c): Object in* 9×9 *boxes.*

Fig 5-8 Covering a same object by $k \times k$ *contiguous boxes.*

5-4- Results:

With the previous settings, a K-NN classifier has been programmed on an image database of 72 images divided into 6 different types of random textures. More precisely, we

started with 6 different textures from Brodatz database (Fig 5-9) and divided each of them into 12 sub-images, resulting in 72 textures.

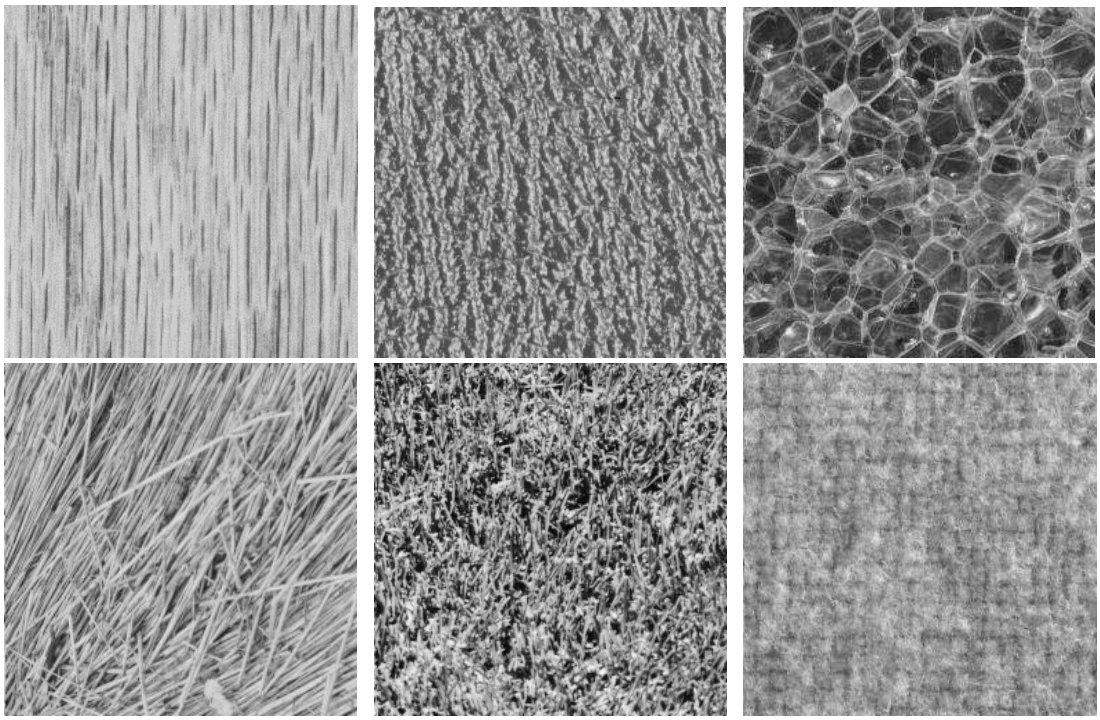


Fig 5-9 The six initial textures

Finally, to assess the robustness of our method, a study of the performance of the classifier has been realized with the addition of noise: on each of the 72 images, we added 16 different levels of Gaussian noise. In such conditions, we will see how different textures behave according to the increase in the noise level.

Remark: a cross-validation method has been performed to choose the best K in the classification, determining K = 3.

Let us present the results for the first 3 textures.

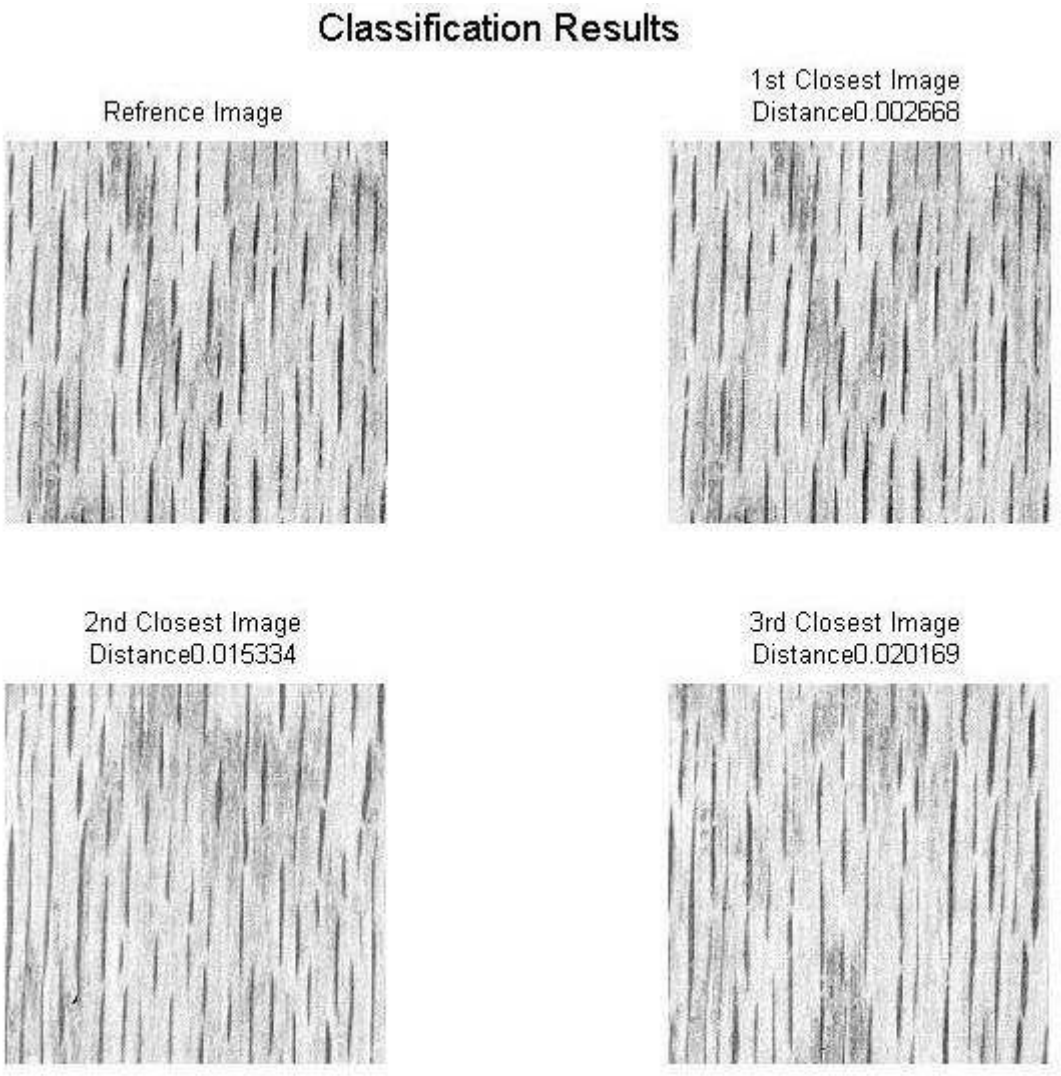


Fig 5-10 Three closest images for texture of type 1.

On Fig 5-10, the three closest images are of the same family. The obtained distances are very small (depend on the noise level). Moreover, we observe that for the increase in the average value of the Gaussian noise randomly assigned on each pixel of the images did not have much influence on the performance of classification (cf. Fig 5-10), which remains above 90% (cf. Fig 5-11).

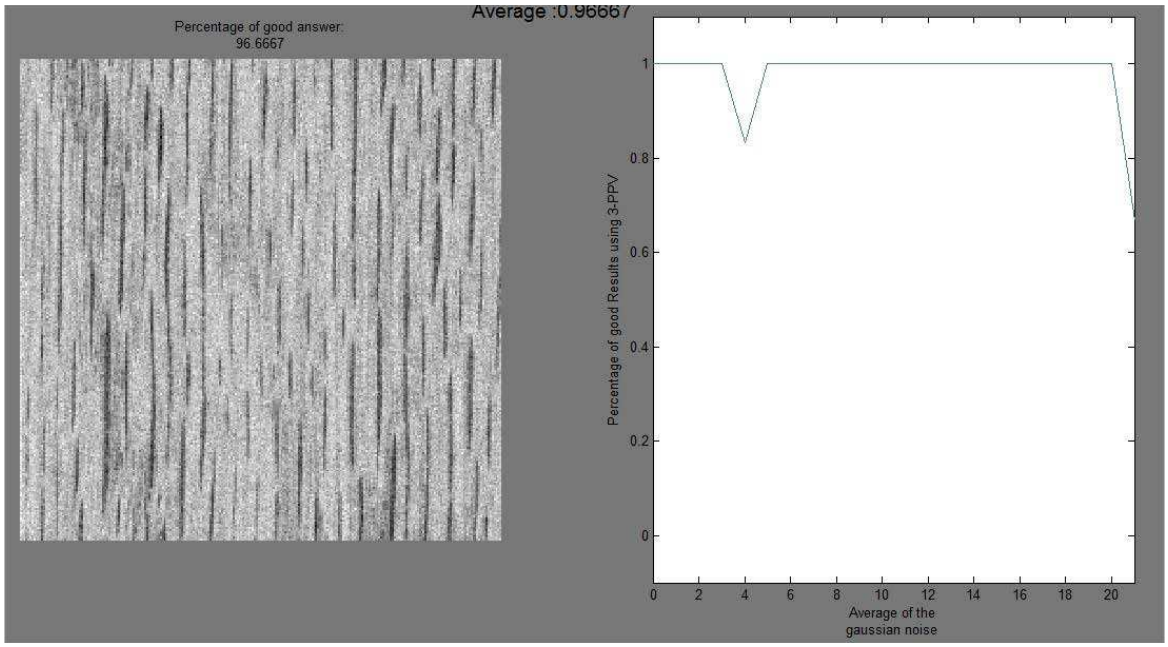


Fig 5-11 Percentage of good answers according to the level of Gaussian noise (texture 1).

For texture of type 2, however, we observe a small degradation of good results after adding noise (cf. Fig 5-12). We begin to observe confusions between this texture and other textures in the database.

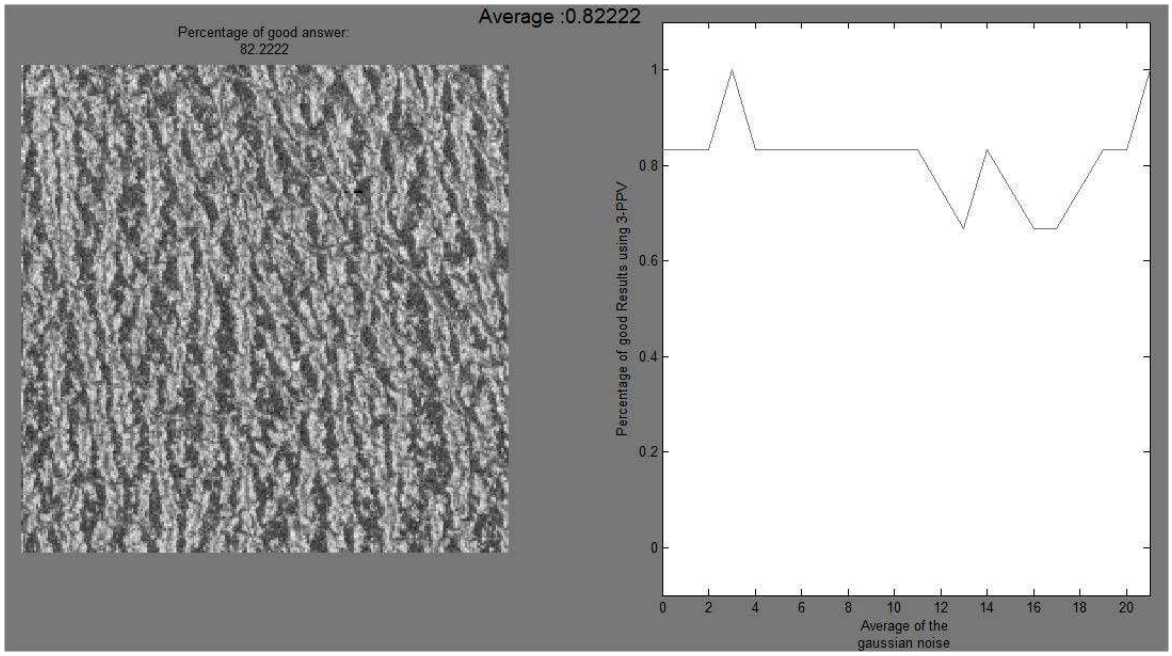


Fig 5-12 Percentage of good results according to the level of Gaussian noise (texture 2).

The texture of type 3, a typical "texture chameleon", has been chosen from image database and shows that more confusions with other textures are observed, even if noise is not added (cf. Fig 5-13).

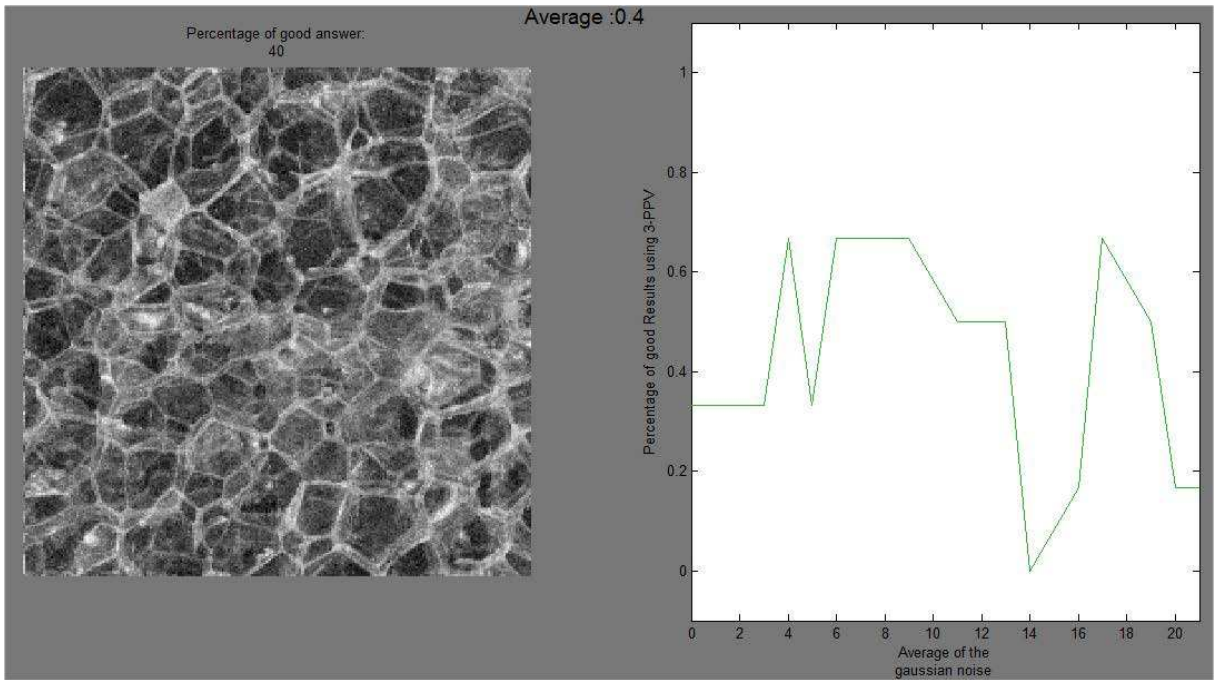


Fig 5-13 Percentage of good results according to the level of Gaussian noise (texture 3).

5-5- Conclusion and perspectives:

The proposed method is novel and the obtained results are promising. Nevertheless, it would need exploratory deeper work.

For example, increasing the number of extracted parameters would improve the performances of the method for texture classification.

Bibliography of chapter 5 :

[1] F. Mayet, 2001, " Classification d'images par analyse fractale et multifractale", Thèse de l'Université Jean Monnet (Saint-Etienne).

[2] G. Fricout, 2004, "Propriétés Morphologiques et Optiques des Surfaces Rugueuses", Thèse de l'Ecole des Mines de Paris.

[3] B. Mandelbrot, 1977, "Fractals : Form, Chance and Dimension", W. H. Freeman and Company.

[4] B. Mandelbrot, 1982, " The Fractal Geometry of Nature ", New-York, W. H. Freeman and Company.

[5] F. Hausdorff, 1919, " Dimension Und Äusseres Mass" , Mathematishe Annalen, Vol. 79, pp. 157-179.

[6] A. S. Besicovitch, 1933, "On the Sets of Fractional Dimensions: On the Sum of Digits of Real Numbers Represented in the Dyadic System" , Mathematishe Annalen, Vol. 110, pp. 321-329.

[7] A. S. Besicovitch, 1934, "Sets of Fractional Dimensions: On a Rational Approximation to Real Number" , Journal of the London Mathematical Society, Vol. 9, pp. 126-137.

[8] A. S. Besicovitch and H.D. Ursell, 1937, "Sets of Fractional Dimensions: On Dimensional Numbers of Some Continuous Curves", Journal of the London Mathematical Society, Vol. 12, pp. 18-25.

[9] M. G. Bouligand , 1928, "Ensembles Impropres et Nombres Dimensionnels" , *Bulletin des Sciences Mathématiques,* Volume 2, N°52, pp : 320-344.

[10] M. G. Bouligand, 1929, "Sur la notion d'ordre de mesure d'un ensemble plan" , *Bulletin des Sciences Mathématiques,* Volume 2, N°52, pp : 185-192.

[11] H. Minkowski, 1901, "Über die Begriffe Länge. Oberfläche und Volumen" , Jahr. Deut. Math., Vol. 9, pp. 115-121.

[12] C. Tricot, 1993, "Courbes et Dimensions" , Springer Verlag.

Résumé Chapitre 5 :

Cette partie est une étude exploratoire sur la possibilité de classer des images texturées au moyen d'un nombre limité de paramètres issus de trajectoires de percolation. Sur une image donnée, ces trajectoires sont construites à partir de points initiaux (appartenant à la première ligne de l'image par exemple), de façon à optimiser une « fonction de coût » liée aux variations de niveaux de gris observées sur les pixels successifs de la trajectoire. A cette dernière, des paramètres sont associés, liés aux changements de direction d'une part, ou à des caractéristiques globales d'autre part, comme la dimension fractale des trajectoires.

Des résultats de classification sont présentés sur un ensemble d'images issues de la base de Brodatz, chacune de ces images ayant été affectée d'un bruit Gaussien croissant. Il est montré que notre approche est très peu sensible au bruit et qu'elle donne des résultats prometteurs sur certaines images. Pour d'autres cas, il apparait nécessaire de prendre en compte d'autres paramètres issus des trajectoires.

Chapter 6 – Conclusion and perspectives.

- **General Conclusion**
- **Conclusion (Français)**
- **Highlights on our contribution**

General conclusion:

At the beginning of the present work, we decided to study the possible contributions of the logarithmic Image Processing (LIP) framework to the processing and interpretation of textured images.

In fact, the concept of texture is not so easy to define, but everybody admits that it is strongly connected to the Human Visual System. Knowing that the LIP Model has been demonstrated as consistent with human vision, it appeared us interesting to create logarithmic tools dedicated to texture evaluation.

In a first time, we have worked to adapt the Haralick parameters in the LIP framework, but most of them necessitate the notion of logarithmic product of two grey levels, which is not rigorously defined. Although some parameters have been defined in a logarithmic version, we have not obtained sufficient results with this approach and we have decided to not present it in the present manuscript.

Concerning the classical method of co-occurrence matrices, their definition is not particularly suitable to the LIP context. It's the reason why we have focused on the covariogram approach, which can be driven by various notions of logarithmic metrics. Such metrics play the role of "correlation" tools, with the advantage to take into account the human vision. Moreover, the logarithmic tools are weakly dependent of illumination variations, and thus produce results rather independent of such variations.

The two last Chapters propose a new approach consisting of considering the grey levels of an image as phases of a medium. According to its grey level, each phase permits to simulate the percolation of a liquid through the medium, thus defining percolation trajectories.

Each propagation of the liquid from a line to another is considered as easy or not, in link with the crossed grey levels. Such a method permits the creation of a "cost" function which modifies the "time" for progressing from a point to another.

Moreover, this cost function may be computed in the LIP framework, with the possible advantage to take into account the Human Visual System.

Finally, the trajectories allow the computation of various parameters including the fractal dimension. On the basis of such parameters, a first experience on random textures classification has been realized, proving the pertinence of the idea.

Obviously, the proposed approach is only a first step and necessitates complementary works to result in a reliable classification method.

Conclusion:

Au début du présent travail, nous avons décidé d'étudier les possibles contributions du Modèle LIP au traitement et à l'interprétation des images textures.

En fait, le concept de texture n'est pas facile à définir, mais il est clair qu'il est fortement lié au Système Visuel Humain. Sachant que le Modèle LIP est compatible avec la vision humaine, il nous a semblé intéressant de créer des outils logarithmiques dédiés à l'évaluation de la texture.

Dans un premier temps, nous avons tenté d'adapter les paramètres de Haralick au contexte LIP, mais la plupart d'entre eux utilisent le produit de deux niveaux de gris, qui n'est pas rigoureusement défini. Bien que certains paramètres aient été définis en version logarithmique, nous n'avons pas obtenu des résultats suffisants avec cette approche et nous ne l'avons pas présentée dans le présent manuscrit.

Concernant la classique méthode des matrices de co-occurrence, leur définition n'est pas particulièrement adaptée au contexte LIP. C'est pourquoi nous nous sommes concentrés sur la notion de covariogramme, qui peut être pilotée par diverses métriques logarithmiques. Ces métriques jouent le rôle d'outils de "corrélation", avec l'avantage de prendre en compte la vision humaine. De plus, les outils LIP sont peu dépendants des conditions d'éclairement et fournissent donc des résultats robustes si celles-ci varient.

Les deux derniers Chapitres proposent une nouvelle approche consistant à considérer les niveaux de gris d'une image comme les phases d'un milieu. Chaque phase permet de simuler la percolation d'un liquide dans le milieu, définissant ainsi des trajectoires de percolation. Chaque propagation d'un pixel à un autre est considérée comme facile ou non, en fonction des niveaux de gris traversés. Une « fonction de coût » est créée, qui modifie le « temps » de propagation d'un point à l'autre. De plus, la fonction de coût peut être calculée dans le contexte LIP, pour prendre en compte la vision humaine.

Finalement, on associe divers paramètres à ces trajectoires, incluant la dimension fractale. Sur la base de ces paramètres, une première expérience de classification de textures aléatoires a été réalisée, prouvant la pertinence de l'idée.

Evidemment, notre étude est seulement une première approche et nécessitera des développements complémentaires pour aboutir à une méthode de classification fiable.

Highlight on our contributions:

As announced at the manuscript beginning, the aim of our work was to propose new tools for texture analysis, in particular in link with the LIP Model, which presents the advantage to be consistent with Human Vision.

Our main contributions are:

- For pseudo-periodic textures, the introduction in the Covariogram approach of logarithmic metrics based on new contrast notions.
- For random textures, a percolation approach producing trajectories more or less tortuous according to the image context. To such trajectories, n selected parameters are computed and the classification of textures is performed on these parameters values.
- The adaptation of Haralick parameters to the LIP framework, but this approach has not produced the expected results: it necessitates the use of a logarithmic product of grey levels and such a definition is not clearly justified.

International Congress

M. INAM UL HAQ and M. JOURLIN, "*Contribution of Logarithmic Tools to Texture Analysis*", XIIIth International Congress of Stereology (ICS XIII), Pékin, Oct. 2011

National Congress.

M. INAM UL HAQ, B. ABDALLAH et M. JOURLIN, "*Tentative de classification de textures aléatoires par percolation logarithmique*", 36ème Journée ISS France, Ecole des Mines de Paris, Fév. 2013.

Publication

After the French ISS (International Society for Stereology) Congress, a publication will be proposed to the review « Image Analysis and Stereology » (May 2013) with title "*Random Textures Classification thanks to Logarithmic Percolation"* (same authors).