DATA ALGORITHM: A NUMERICAL METHOD TO EXTRACT SHAPE INFORMATION FROM GRAY SCALE IMAGES *

R. Davoli and F. Tamburini †

Abstract
This paper presents a numerical algorithm which realizes a translation, rotation and scale invariant transform for single object gray scale images. This kind of algorithm is applicable to all those applications which need to deal with shape information, independent of the orientation, distance and position of an object. The algorithm we present takes its mathematical foundation from the log-polar transform [2]. These analytical results cannot be mapped directly into an algorithm: they involve generalized integrals that are hard to compute and the numerical implementation has to take account of discretization error which may propagate and lead to meaningless results. The aim of this paper is to present an algorithm which overcomes these problems.

1 Introduction

Many applications depend on object recognition features. Robotic applications need to recognize mechanical parts in order to perform some automatic actions on them. Other typical industrial applications for this kind of algorithm are those used to catalogue objects. In medicine, it is important to be able to recognize cells which have characteristics dependent on their shape. In X-Ray slides, echography and tomography opaque zones having a specific shape are symptoms for particular diseases. Other applications might be military (such as target discovery from satellite or aerial pictures and target driven weapons), astronomical (galaxy cataloguing) and so on. When an electronic system digitizes an object the distance, the position and orientation of the desired object in the picture are not generally known a priori. Our approach to this problem is to split the recognition process into two phases: phase one is a translation, rotation and scale invariant transform to eliminate these extraneous factors from the image, thereby leaving an easier job for the real pattern recognition algorithm (phase two).
What are the essential features of a transform algorithm for the first phase of this process?

- It has to maintain shape information. The result must be shape dependent.
- It has to be robust: additive noise, up to a reasonable level, as a result of digitization errors, must be masked out.

*Partial support for this work was provided by the the Italian National Research Council (CNR) under contract n. 92.00069.CT12.115.25585.

†Both authors are with Dipartimento di Matematica, Università di Bologna, Italy. E-mail addresses: {renzo,tamburin}@cs.unibo.it.

- It has to be flexible: most of the recognition applications today are tailored to specific problems, whereas this algorithm can be used for a wide range of recognition problems.

- Computation should be performed quickly: speed is a critical factor for some applications. To be quick, in this context, does not necessarily mean that the algorithm should have a low sequential computational complexity, but rather that the algorithm can run on several processor concurrently with no data exchange (except the initial feed, and the result output). By increasing the number of processors, the complexity of the algorithm's parallel elements can be reduced.

This paper is organized as follows: section 3 presents the analytical foundations; section 4 describes the structure and section 5 the implementation of the DATA algorithm. Section 6 presents some test results and the computational complexity of the algorithm is discussed in section 7. The appendix contains proofs of all the mathematical assertions.

2 Related works

Bi-dimensional shape recognition algorithms and methodologies are well documented in the bibliography. Much research has been carried out on this topic and has used many different approaches. Xu and Yang [6] used multidimensional orthogonal polynomials, He and Kundu [4], Sekita, Kurita and Otsu [5] used autoregressive models. Both these approaches are not that well suited for gray scale images: in fact these methods take as input the boundary lines of objects.
Other methods are based on the numerical Mellin transform of data. Zwicke and Kiss [8] used a Mellin transform to recognize the shapes of ships from radar signals, Altmann and Reitbock [1] used a scale and translation invariant transform. A complete recognition system is presented in Wechsler and Zimmerman [7]. Its pattern matching algorithm is an associative memory, and the pre-processing transform provides rotation and scale invariancy.
Not one of the above algorithms achieves a full translation, rotation and scale invariant transform. The critical point in exploiting such a feature is that the algorithm needs more than one transform step with a consequent propagation of computational errors.

3 Analytical foundations

The analytical foundation of this algorithm was first introduced by Casasent Psaltis [2]. Fourier analysis on data allows us to divide a signal up into a discrete or continuous sum of sinusoidal functions. The Fourier Transform (FT) of a function $f(x)$, takes the form:

$$F(u) = \mathcal{FT}\{f(x)\}(u) = \int_{-\infty}^{+\infty} f(x)e^{-jux}\,dx.$$

The Fourier transform is commonly used in image processing to filter an input image by decreasing or nulling some frequencies. In fact, a FT can be seen as a function which assigns to each frequency the amplitude and the phase of that frequency 'wave' that is part of the analyzed signal. Clearly, shifting a signal affects only the phase of the transformed result and not the magnitude: if we consider the exponential form of the complex function F, $F(u) = |F(u)|e^{j\varphi(u)}$, we can outline that a translation does not affect the FT magnitude: $|\mathcal{FT}\{f(x-a)\}(u)| = |F(u)e^{jua}| = |F(u)|$.
This property is also true for the Fourier bi-dimensional transform of images. The magnitude is also invariant for translation in this case.
The spectrum of a FT also has the following properties.

- It has its maximum at the origin $O = (0, 0)$ and this value is proportional to the total energy of the image.
- It maintains rotation: by transforming an image which has been rotated through an angle α, an image spectrum is obtained which is also rotated through the same angle α about O.
- It inverts scale operations: by transforming a scaled picture by a factor α, the image obtained is scaled by a factor $\frac{1}{\alpha}$.

So, FT magnitude is translation invariant and rotation and scale operations are mapped into equivalent operations centered about the origin of the frequency domain.
By converting rectangular (cartesian) co-ordinates into polar co-ordinates, rotations about O are clearly mapped into translation operations on the θ axis, while scale deformations from O are mapped to scale operations of the same ratio but on the radial direction only.
Mono-directional scale on the interval $[0, +\infty[$ can be mapped by a translation operation onto the interval $]-\infty, +\infty[$ by applying a Mellin transform, that uses logarithmic scaling ($\log \alpha x = \log \alpha + \log x$).
So by first using FT, translation operations can finally be masked out, by converting the result into polar co-ordinates and by log-scaling the radial co-ordinate, scale and rotation can also be interpreted as translations, which are then masked out by a second FT.
Several problems affect the method when implemented as an algorithm:

- the input of the second FT on the domain $]-\infty, +\infty[\times[0, 2\pi[$ is not a limited data set, i.e. non null points can be found at any distance from the origin. This input function is real, non negative, and reaches its global maximum at $x = -\infty$. As a result, all the integrals involved in the second FT are generalized integrals (hard to approximate with numerical methods).
- there has to be a trade off between discretization steps of data, accuracy and complexity. To clarify this concept, let us examine a poor implementation: a brute force approach to the problem carried out using Fast Fourier Transforms (FFTs), bitmap to bitmap polar co-ordinates and log scaling map conversion, leads to meaningless results because most of the output image depends on the values of a very few pixels all concentrated around the first FT origin. FFTs generate a uniform square grid of Fourier spectrum samples, whereas this application needs a very high density of samples near to the origin and fewer samples far from it.

4 The DATA algorithm

As in FFT theory, our algorithm takes as input an $N \times N$ square grid of values. However whereas the FFT computes the Fourier transform values on a different $N \times N$ uniform mesh, as we said, the DATA algorithm needs to value the FT on a greater number of points near to the origin and a fewer number of points far from it. In order to compute the FT on any point of the cartesian real plane, the input samples set $W = \{w_{h,k}\}_{h=0,\cdots,N\ k=0,\cdots,N}$ is extended to a function $f : \Re \times \Re \to \Re$ using the following definition:

$$f_W(x, y) = \begin{cases} w_{h,k} & \text{if } \frac{2\pi h}{N} \leq x < \frac{2\pi(h+1)}{N} \\ & \text{and } \frac{2\pi k}{N} \leq y < \frac{2\pi(k+1)}{N} \\ 0 & \text{otherwise} \end{cases}$$

Hence,

$$\mathcal{FT}\{f_W(x,y)\}(u,v) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f_W(x,y)\; e^{-j(xu+yv)}\, dx\, dy = K\sum_{k=0}^{N}\sum_{h=0}^{N} w_{h,k}\; e^{-\frac{j2\pi}{N}(ku+hv)}$$

A very good level of accuracy can be achieved by performing the first FT, rect-to-polar conversion and log-scaling in a single step. In this way there is no additional loss of accuracy due to multiple discretizations. The result of these transformation can be computed directly onto $(\rho,\, \theta)$ using the following formula:

$$LP_W(\rho,\theta) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f_W(x,y)\; e^{-j\; e^{\rho}(x\cos\theta+y\sin\theta)}\, dx\, dy = K\sum_{k=0}^{N}\sum_{h=0}^{N} w_{h,k}\; e^{-\frac{j2\pi}{N}e^{\rho}(h\cos\theta+k\sin\theta)}$$

This technique, however, is not sufficient to build a real algorithm which implements the transform we need. The output of this first numerical step is defined on a non-finite stripe of height 2π. The FT row computations involve generalized integrals (as seen above). To achieve greater accuracy and better performance, the analytical algorithm needs to be changed; a different kind of scale invariant transform has to be used. Another drawback in using or modifying the original method for scale invariancy is that no meaningful FT information on discrete data can be obtained beyond a square having the same size of the input grid. In fact, the value of the FT at each of these points, represents the amplitude and phase of a component having more than one oscillation per pixel. The numerical algorithm needs to 'cut' a finite interval of the stripe in a scale independent way; this chunk needs to contain the essential information of the shape and should not intersect the frequency overflow area.

All the rows of the LP transform output (on a discrete $N \times N$ grid of real positive input data) can be proven (see Appendix A) to be definitely monotonic as $\rho \to -\infty$. More precisely: $\forall\theta \in [0, 2\pi[\;\; LP_W(\rho,\theta)$ is monotonically decreasing on ρ, $\forall\rho \in \left]-\infty, \log\left(\frac{\sqrt{2}}{8}\right)\right[$. Consequently even the mean on θ (see note[1]): $\overline{LP}_W(\rho)$, is monotonically decreasing in the same interval. Defining (see Appendix A):

$$\Xi_W = \frac{\overline{LP}_W\left(\log\left(\frac{\sqrt{2}}{8}\right)\right)}{\overline{LP}_W(-\infty)}$$

and calling $\Xi = \max_{W\in IS}\; \Xi_W$, where IS is the set of all the images to be transformed, then clearly,

$$\forall W \in IS\; \exists!\rho_0 \in \left]-\infty, \log\left(\frac{\sqrt{2}}{8}\right)\right[\text{ such that } \frac{\overline{LP}_W(\rho_0)}{\overline{LP}_W(-\infty)} = \Xi$$

Even if sometimes it is not reasonable to process all the possible input images to compute the correct value for Ξ, it can be approximated experimentally by pre-processing a set of test images. In practice, by testing the algorithm on different images we have found that the cut parameter can be fixed as the maximum Ξ on all images in their larger size. Ξ, in fact, depends only on image size and contrast.

Once a size independent point on the stripe has been found it can be used as the origin and thereby the point from which to gather the shape information for further processing (see fig. 1).

Another parameter of the algorithm is the length of the selected radius interval. The shorter this interval is, the higher the scalability, i.e. similar objects of very different size can be recognized, however, by using a lower range of frequencies, the whole process takes into account only the most

[1] $\overline{LP}_W(\rho)$ is the mean value of the LP function computed on a finite set of points. For the purpose of this paragraph it can be considered either as a continuous or discrete mean.

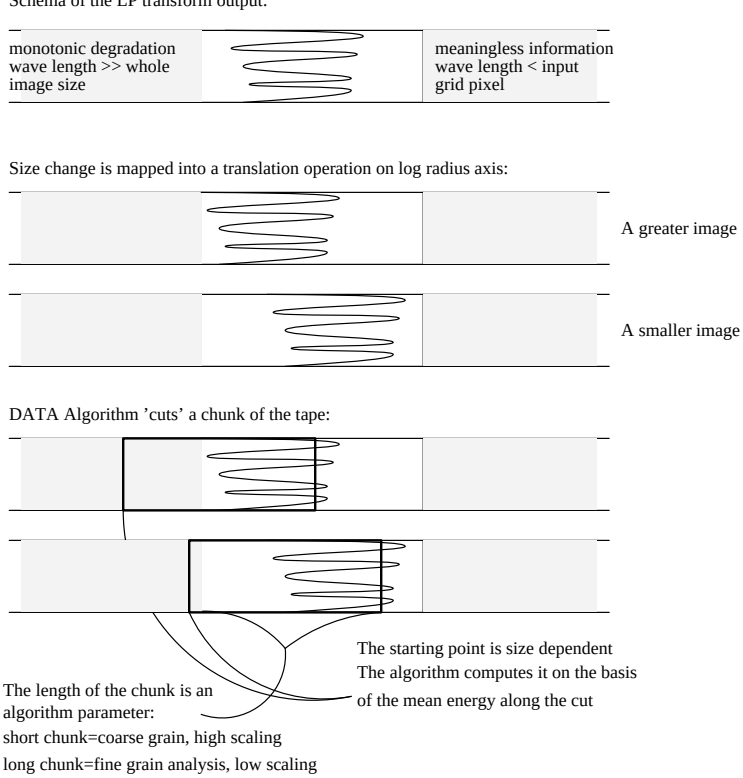

Figure 1:

general characteristics of the object's shape and small details are lost. On the other hand, by using long intervals it is also possible to take account of small details in the recognition process, and yet this leads to a lower degree of scalability. I.e. by scaling down the object, the small details may become smaller than the input sampling grid (see fig. 1). In this case, the algorithm tries to compute a FT outside a space having the same dimension of the input domain. As we pointed out before, this is not a well-defined operation.

5 Implementation of the DATA algorithm

The implementation of the DATA algorithm consists of three parts (see fig. 2):

- the starting point search on the main maximum of the FT in order to cut the unlimited stripe,
- FT, polar conversion and logarithmic scaling,
- FFT on image columns.

The first part uses the results obtained in appendix A to compute the starting point for logarithmic scaling. To achieve this, we have built a search algorithm that examines the module of the image's FT; for each step it computes the mean of the FT values on a circular region (see fig 3).
Starting with $\rho_0 = \log \frac{\sqrt{2}}{8}$, it searches for the value of ρ_0 that gives a mean, on the circular region, equal to $\Xi * F(0,0)$, where Ξ is a real value between 0 and 1, as defined in the previous paragraph, and $F(0,0)$ is the value of the FT at the origin (i.e. a value proportional to the image's total energy). Modifying ρ_0 according to the mean, the search algorithm can find the correct ρ_0, that will then be used to compute the logarithmic scaling.

DATATransform(in,out)
in input image
out transformed image
begin
/* *Logarithmic scaling starting point search* */
ρ_0 = $\log \frac{\sqrt{2}}{8}$
repeat
mean = 0
for each θ **from** 0 **to** π (N steps)
mean = mean + $\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \mathtt{in}[x,y] e^{-j\frac{2\pi}{N} e^{\rho_0}(x\cos\theta + y\sin\theta)}$
ρ_0 = $\mathcal{S}$(mean)
until mean = $\Xi * F(0,0)$

/* *Fourier transform, Polar conversion and L ogarithmic scaling* */
for each ρ **from** 0 **to** $N-1$
for each θ **from** 0 **to** π (N steps)
$LP_\rho[\theta]$ = $\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \mathtt{in}[x,y] e^{-j\frac{2\pi}{N} e^{\varphi(\rho)-\rho_0}(x\cos\theta + y\sin\theta)}$

/* *Fast Fourier Transform on LP columns* */
for each ρ **from** 0 **to** $N-1$
for each θ **from** 0 **to** π (N steps)
out[θ, ρ] = $(\mathcal{FT}\{LP_\rho\})[\theta]$
end

Figure 2: **The DATA-algorithm.**

To improve the performance we use an hybrid search algorithm, that starts with a linear search, when ρ_0 is far from the correct value, and then switches to a dichotomic search, as it approaches the final value, resulting in a high precision approximation. This task is synthesized in the function $\mathcal{S}$, that updates ρ_0 according to the search type.
The second part of the DATA-algorithm computes the FT, polar and logarithmic scaling in one step, using as the starting point of logarithmic scaling the ρ_0 value computed previously. The function φ rescales the $[0, N]$ interval onto an $[0, G]$ interval, where G is the length of the chunk as seen in fig 1.
Another device that we have used to improve the performance of this phase, and the previous one, is to compute the transform for θ from 0 to π only; since the FT exhibits conjugate symmetry, given that $|F(u,v)| = |F(-u,-v)|$, we can compute the LP transform for just half of total image, and rearrange the results to complete the entire transform.
The first two phases give a translation independent transformed image, and convert scale and rotation on the original image into translations.
A final FFT realizes the rotation and scale invariance for the original image.

6 Experimental results

We have used two sets of test images. The first set consists of five artificial images containing three squares in general random positions (which means that they have different translation, rotation and scale factors), a circle and a triangle. In fig 4 you can see these images and their corresponding transformed images.
To show the transformed images in all their detail, we have used, in all the figures shown in this paragraph, the classical logarithmic enhancement that reinforces the secondary maxima of FTs.
We can see immediately, by looking at figure 4, that although the square images are very different,

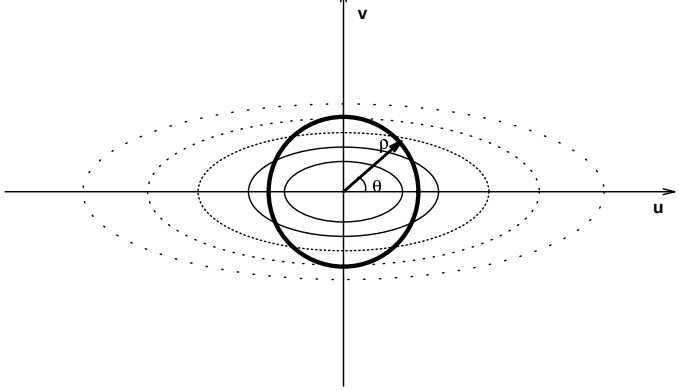

Figure 3: **This is a schematic view of the FT main maximum and the circular path for the initial point search.**

	Square 1	Square 2	Square 3	Circle
Square 2	3.31	*	*	*
Square 3	2.64	3.16	*	*
Circle	83.23	83.56	83.28	*
Triangle	135.46	134.73	134.61	123.87

Table 1: **This table shows the distance between the artificial test images**

in position rotation and scale, their transformed images are very similar and no obvious differences are observed, whereas the other objects' images have very different transforms.
In order to show the effectiveness of the procedure we have valued the euclidean distance between the transformed images. Table 1 shows that in this test different shapes are, on average, 37 times further than similar ones (the worst case is about 25).
In the second test set the algorithm processed four digitized gray scale images from two galaxies. Three of them are images of galaxy NGC1365 digitized in three different positions, orientation and scale, the fourth is an image of NGC3938. Again, we have included all the source and transformed images and a table containing the euclidean distances between the resulting outputs (see fig. 5 and table 2).
Gray scale images are managed as an effective way as binary images are. The distance between different instances of the same object due to discretization and computational errors are negligible when compared to the distance between two different objects.
In this section we have shown empirically that the DATA-algorithm can be seen as a discrete counterpart of the analytical transform, underlining its ability to extract shape information from images, removing all geometrical transform such as translation rotation and scaling.

7 Computational complexity

As we showed in the previous paragraphs we have introduced some changes in the mathematical transform process in order to overcome discretization problems. The FFT algorithm computes the FT on a square grid with a complexity bound of $O(N^2 \log N)$; these results can be achieved by using the variable separability, and by re-using common partial results. By computing the FT on a log-polar grid (as we do in the LP function) variables do not appear to be separable and there is no common partial result, then the scalar performance cannot be further improved. Even if $O(N^4)$ is

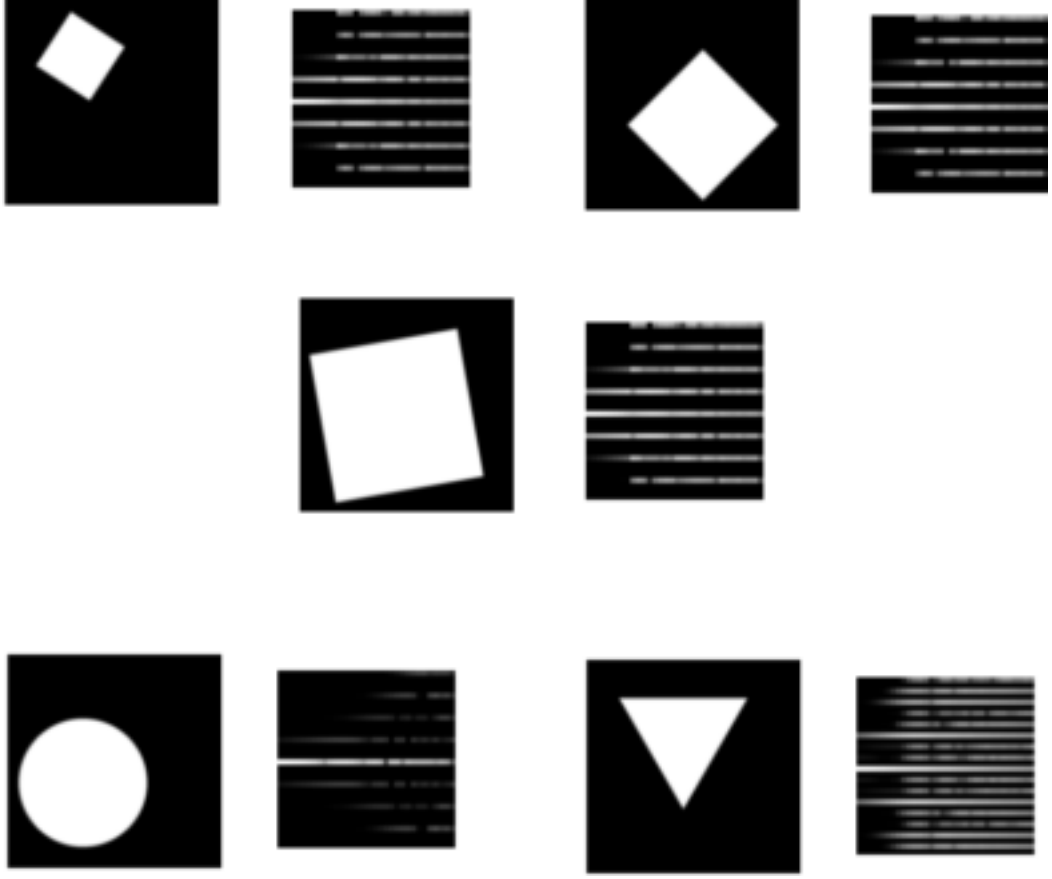

Figure 4: **Artificial test images and their transforms.**

a polynomial bound complexity, it can be quite difficult to handle high resolution picture or moving frames.
All the considerations above assume that the size of the input grid equals that of the output grid. In a common application even if the input is a high resolution picture, a lower size transformed image is sufficient for the recognition process. Where N is the input picture size and M the output picture size the complexity clearly reduces to $O(N^2M^2)$
The algorithm can also be parallelized to improve the real performance. Given that the LP transform computations for different output points do not share any of the partial results, this algorithm can easily be parallelized in a very effective way. In fact a speed up factor close to the theoretical maximum can be achieved by distributing the output point computations among the existing processors. The algorithm has no synchronization bottle necks except for the initial broadcast of the original image and the final collect of results.

8 Concluding remarks

In this paper we presented the DATA-algorithm, a translation, rotation and scale invariant transform. The result that we are obtaining on the test data set are very encouraging. Our future plans include an extension of the algorithm to operate as a complete recognition system, the study of specific applications and the incorporation of a color image processing feature.

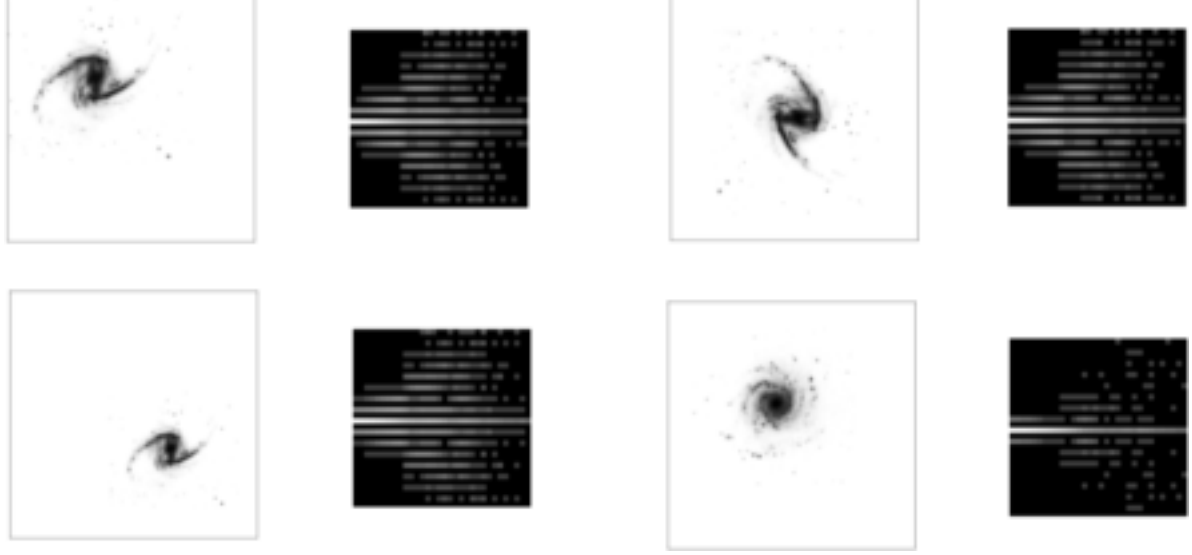

Figure 5: **Galaxies' gray scale images and their transforms.**

	NGC1365 a	NGC1365 b	NGC1365 c
NGC1365 b	6.63	*	*
NGC1365 c	6.78	6.92	*
NGC3938	92.94	93.27	94.08

Table 2: **This table shows the distance between the gray scale images of galaxies**

9 Acknowledgments

We wish to thanks prof. D. Bollini for his helpful advises and his encouragement throughout the project, dr. C. Pagnani for helping us to test the early version of the code and dr. L.A. Giachini for his technical support and suggestions.

A Appendix

Let us call an N-sampled monodimensional function $f : [a, b[\rightarrow \Re$, a function which is constant on each subinterval

$$I_k = [a + \frac{(b-a)}{N}k, a + \frac{(b-a)}{N}(k+1)[\qquad \forall k = 0..N-1.$$

This definition captures the abstraction of a digitized signal: the input function is a step function in which a single sample value is constant until the next sampling point.

A.1 Proposition:

The magnitude of the Fourier transform of an N-sampled real non-negative monodimensional function $f : [0, 2\pi[$ is monotonic decreasing in the interval $[0, 1/4[$.
Proof: Let's define $y_1, \ldots, y_N$ as:

$$y_k = f\left(a + \frac{b-a}{N}k\right) = f\left(\frac{2\pi}{N}k\right),$$

then

$$|\mathcal{FT}\{f(x)\}(u)|^2 = \left|\sum_{k=0}^{N-1} y_k\ e^{-j\frac{2\pi ku}{N}}\right|^2 = \left(\sum_{k=0}^{N-1} y_k \cos\left(\frac{2\pi ku}{N}\right)\right)^2 + \left(\sum_{k=0}^{N-1} y_k \sin\left(\frac{2\pi ku}{N}\right)\right)^2 =$$
$$= \sum_{k=0}^{N-1}\sum_{h=0}^{N-1} y_k y_h\ \left(\cos\left(\frac{2\pi ku}{N}\right)\cos\left(\frac{2\pi hu}{N}\right) + \sin\left(\frac{2\pi ku}{N}\right)\sin\left(\frac{2\pi hu}{N}\right)\right) =$$
$$= \sum_{k=0}^{N-1}\sum_{h=0}^{N-1} y_k y_h\ \cos\left(\frac{2\pi(k-h)u}{N}\right)$$

If $u < \frac{1}{4}$

$$\left|\frac{2\pi(k-h)u}{N}\right| < \frac{\pi}{2}.$$

Then each sum term

$$y_k y_h\ \cos\left(\frac{2\pi(k-h)u}{N}\right)$$

is positive and monotonic decreasing on u. ■

Let's also define an NxM-Sampled bidimensional function $f : [a,b[\times[c,d[\rightarrow \Re$, a function which is constant on each square subset of its domain

$$I_{k,h} = [a + \frac{(b-a)}{N}k, a + \frac{(b-a)}{N}(k+1)[\times[c + \frac{(d-c)}{M}h, c + \frac{(d-c)}{M}(h+1)[$$
$$\forall k = 0..N-1 \text{ and } h = 0..M-1.$$

A.2 Proposition:

The Norm of the Fourier transform of an NxN-Sampled real non-negative bidimensional function $f : [0,2\pi[\times[0,2\pi[$ is monotonic decreasing if computed on the segments $t\vec{v}$ with $t = [0, \frac{\sqrt{2}}{8}]$ and $\vec{v} : \|\vec{v}\| = 1$.
Proof: Let's define $y_{1,1}, y_{1,2}, \ldots, y_{N,N}$ as:

$$y_{k,h} = f(\frac{2\pi}{N}k, \frac{2\pi}{N}h),$$

then

$$|\mathcal{FT}\{f(xy)\}(u,v)|^2 = \left|\sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1} y_{k_1,k_2}\ e^{-j\frac{2\pi}{N}(uk_1+vk_2)}\right|^2 =$$
$$= \left(\sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1} y_{k_1,k_2}\cos\left(\frac{2\pi}{N}(uk_1+vk_2)\right)\right)^2 + \left(\sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1} y_{k_1,k_2}\sin\left(\frac{2\pi}{N}(uk_1+vk_2)\right)\right)^2 =$$
$$= \sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1}\sum_{h_1=0}^{N-1}\sum_{h_2=0}^{N-1} y_{k_1,k_2}\ y_{h_1,h_2}\ \cos\left(\frac{2\pi}{N}(uk_1+vk_2)\right)\cos\left(\frac{2\pi}{N}(uh_1+vh_2)\right) +$$
$$+ \sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1}\sum_{h_1=0}^{N-1}\sum_{h_2=0}^{N-1} y_{k_1,k_2}\ y_{h_1,h_2}\ \sin\left(\frac{2\pi}{N}(uk_1+vk_2)\right)\sin\left(\frac{2\pi}{N}(uh_1+vh_2)\right) =$$
$$= \sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1}\sum_{h_1=0}^{N-1}\sum_{h_2=0}^{N-1} y_{k_1,k_2}\ y_{h_1,h_2}\ \cos\left(\frac{2\pi}{N}(u(k_1-h_1)+v(k_2-h_2))\right) \leq$$

$$\leq \sum_{k_1=0}^{N-1}\sum_{k_2=0}^{N-1}\sum_{h_1=0}^{N-1}\sum_{h_2=0}^{N-1} y_{k_1,k_2}\ y_{h_1,h_2}\ \cos(2\pi(u+v))$$

The last inequality holds if $|k_1 - h_1| < N$ and $|k_2 - h_2| < N$.
If $|u+v| < \frac{1}{4}$

$$|2\pi(u+v)| < \frac{\pi}{2}$$

this is only true if $||(u,v)|| < \frac{\sqrt{2}}{8}$. ■

In this case the monotony of this function descends from the sum terms monotony.
The segments $t\vec{v}$ becames the segments $[0, \frac{\sqrt{2}}{8}[\times\{\arctan\frac{v_y}{v_x}\}$. when the spectral space is converted into polar co-ordinates. Then the vertical sum (continuous or discrete) and the vertical mean of the spectral values in polar co-ordinates become monotonic decreasing functions in the interval $[0, \frac{\sqrt{2}}{8}[$.
Being

$$\overline{LP}_W(\rho) = \frac{1}{N}\sum_{k=0}^{N-1} LP_W(\rho, \frac{2\pi k}{N}) = \frac{1}{N}\sum_{k=0}^{N-1} \mathcal{FT}\{f_W(x,y)\}(e^\rho \cos\frac{2\pi k}{N}, e^\rho \sin\frac{2\pi k}{N})$$

and extending the definition to $-\infty$ as

$$\overline{LP}_W(-\infty) = \mathcal{FT}\{f_W(x,y)\}(0,0)$$

Let's define the recognition factor Ξ_W of an $N \times N$-sampled bidimensional function $f_W : [0, 2\pi[\times[0, 2\pi[$ as:

$$\Xi_W = \frac{\overline{LP}_W(\log(\frac{\sqrt{2}}{8}))}{\overline{LP}_W(-\infty)} = \frac{\sum_{k=0}^{N-1}\mathcal{FT}\{f_W(x,y)\}(\frac{\sqrt{2}}{8}\cos\frac{2\pi k}{N}, \frac{\sqrt{2}}{8}\sin\frac{2\pi k}{N})}{N\mathcal{FT}\{f_W(x,y)\}(0,0)}$$

Ξ_W is the mean of the FT values on N sampling along the circle of radius $\frac{\sqrt{2}}{8}$ divided by the value of the FT at O (i.e. the total energy of the image).
The parameter Ξ_W is well defined on any non-null image and depends on the energy of the visual signal (size and contrast).
As we have already pointed out in paragraph 4, where IS is the set of all the images to be processed and $\Xi = \max_{W\in IS} \Xi_W$,

$$\forall W \in IS\ \exists!\rho_0 \in \left[-\infty, \log\left(\frac{\sqrt{2}}{8}\right)\right[\text{ such that } \frac{\overline{LP}_W(\rho_0)}{\overline{LP}_W(-\infty)} = \Xi$$

The value of ρ_0, which can be computed using a search algorithm, represents a size independent starting point for the following processing steps of the algorithm.
It can also be said that that the algorithm correctly computes its transform by fixing a Ξ value, for all W images having a Ξ_W less than or equal to Ξ.

References

[1] Altmann J. & Reitbock H. "A fast correlation method for scale and translation invariant pattern recognition." *IEEE Tr. Patt. Anal. Machine Intelligence*, Vol 6 n 1: 47–57, 1984.

[2] Casasent D. & Psaltis D. "Position, rotation and scale invariant optical correlations." *Applied Optics*, (15): 1793–1799, 1976.

[3] Frosini P. "Measuring shapes by size functions" *Intelligent Robots and Computer Vision X: Algorithms and Techniques, SPIE Proc.*, Vol. 1607: 122–133, 1991.

[4] He Y. & Kundu A. "2-D Shape classification using hidden Markov model." *IEEE Tr. Patt. Anal. Machine Intelligence*, Vol 13 n 11: 1172–1185, 1991.

[5] Sekita I., Kurita T. & N. Otsu "Complex autoregressive model for shape recognition." *IEEE Tr. Patt. Anal. Machine Intelligence*, Vol 14 n 4: 489–496, 1992.

[6] Xu J. & Yang Y-H. "Generalized multidimensional orthogonal polynomials with applications to shape analysis." *IEEE Tr. Patt. Anal. Machine Intelligence*, Vol 12 n 9: 906–913, 1990.

[7] Wechsler H. & Zimmerman G.L. "2-D invariant object recognition using distributed associative memory." *IEEE Tr. Patt. Anal. Machine Intelligence*, Vol 10 n 6: 811–821, 1988.

[8] Zwicke P.E. & Kiss I. jr. "A new implementation of the Mellin transform and its application to radar classification of ships." *IEEE Tr. Patt. Anal. Machine Intelligence*, Vol 5 n 2: 191–199, 1983.