Fast Algorithm for Detecting the Most Unusual Part of 2d and 3d Digital Images. Application to Large Medical Databases.

KOSTADIN KOROUTCHEV
Universidad Autónoma de Madrid
Escuela Politécnica Superior
Cantoblanco, Madrid, 28049
SPAIN
k.koroutchev@uam.es

ELKA KORUTCHEVA
Depto. de Física Fundamental, UNED,
c/ Senda del Rey 9, 28080 Madrid
SPAIN
and G.Nadjakov Inst. Solid State Physics
Bulgarian Academy of Sciences
Sofia, Bulgaria
elka@fisfun.uned.es

Abstract: In this paper we introduce a fast algorithm that can detect the most unusual part of a digital image. The most unusual part of a given shape is defined as a part of the image that has the maximal distance to all non intersecting shapes with the same form. The method is tested on two and three-dimensional images and have shown very good results without any predefined model. The results can be used to scan large image databases, as for example medical databases.

Key–Words: image processing, image statistics, image recognition

1 Introduction

In this paper we are trying to find the most unusual/rare part with predefined **shape** of a given image. If we consider an one-dimensional quasi-periodical image, as for example electrocardiogram (ECG), the most unusual parts with length about one second will be the parts that correspond to rhythm abnormalities [1]. Therefore they are of some interest. Considering two and three dimensional images, we can suppose that the most unusual part of the image can correspond to something interesting of the image.

Recently we have presented an algorithm that can detect the most rare part of a digital image, referring to two-dimensional images [2]. We have shown that this part of the image is defined as a part of the image that has the maximal distance to all non-intersecting shapes with the same form. In fact the algorithm can be used in more that two dimensions. The present paper is an extension of this method in the case of three-dimensional images.

Some related publications concerning the probabilistics models as well as the problem of object recognition can be find in Refs. [3, 4, 5, 6].

To state the problem, we need first of all a definition of the term "most unusual part". Let us chose some shape S within the image A, that could contain that part and let us denote the cut of the figure A with shape S and origin $\vec{r}$ by $A_S(\vec{\rho}; \vec{r})$, e.g.

$$A_S(\vec{\rho}; \vec{r}) \equiv S(\vec{\rho})A(\vec{\rho} + \vec{r}),$$

where $\vec{\rho}$ is the in-shape coordinate vector, $\vec{r}$ is the origin of the cut A_S and we used the characteristic function $S(.)$ of the shape S. We can suppose that the rarest part is the one that has the largest distance with the rest of the cuts with the same shape.

Namely, we can suppose that the most unusual part is located at the point $\vec{r}$, defined by:

$$\vec{r} = \arg\max_{\vec{r}} \min_{\vec{r'}:|\vec{r'}-\vec{r}|>\operatorname{diam}(S)} ||A_S(\vec{r}) - A_S(\vec{r'})||. \tag{1}$$

Here we assume that the shifts do not cross the border of the image. The norm $||.||$ is assumed to be L_2 norm[1]

Because the parts of an image that intersect significantly are similar, we do not allow the shapes located at r' and r to intersect, avoiding this by the restriction on $r' : |\vec{r'} - \vec{r}| > \operatorname{diam}(S)$.

The definition above can be interesting as a mathematical construction, but if we are looking for practical applications, it is too strict and does not correspond exactly to the intuitive notion of the interesting part as there can be several interesting parts. Therefore the correct definition will be to find the outliers of the distribution of the distances between the blocks $||.||$.

---

[1]Similar results are achieved with L_1 norm. The algorithm was not tested with L_{max} norm due to its extreme noise sensitivity. We use L_2 because of its relation with PSNR criteria that closely resembles the human subjective perception.

In d-dimensional space the figure with linear size N has N^d points and if $||S|| \ll ||A||$, in order to find deterministically the most unusual part, we need N^d operations. This is unacceptable even for large two dimensional images, not concerning 3D image databases. Therefore we are looking for an algorithm that provides an approximate solution of the problem and solves it within some probability limit in acceptable execution time.

As is defined above in Eq.(1), the problem is very similar to the problem of location of the nearest neighbor between the blocks. This problem has been studied in the literature, concerning Code Book and Fractal Compression [7]. However, the problem of finding $\vec{r}$ in the above equation, without specifying $\vec{r'}$, as we show in the present paper, can be solved by using probabilistic methods avoiding slow calculations.

2 The Method

2.1 Projections

The problem of estimating the minima of Eq. (1) is complicated because the block is multidimensional. Therefore we can try to simplify the problem by projecting the block $B \equiv A_S(\vec{r})$ in one dimension using some projection operator X. For this aim, we consider the following quantity:

$$b = |X.B_1 - X.B| = |X.(B_1 - B)|, \quad |X| = 1. \quad (2)$$

The dot product in the above equation is the sum over all ρ-s:

$$X.B \equiv \sum_{\vec{\rho}} X(\vec{\rho}) B(\vec{\rho}; \vec{r}). \quad (3)$$

If X is random, and uniformly distributed on the sphere of corresponding dimension, then the mean value of b is proportional to $|B_1 - B|$; $\langle b \rangle = c|B_1 - B|$ and the coefficient c depends only on the dimensionality of the block. However, when the dimension of the block increases, the two random vectors ($B_1 - B$ and X) are close to orthogonal and the typical projection is small. But if some block is far away from all the other blocks, then with some probability, the projection will be large. The method resembles that of Ref. [8] for finding nearest neighbor.

As mentioned above we must look for outliers in the distribution. This would be difficult in the case of many dimensions, but easier in the case of one dimensional projection.

We will regard only projections orthogonal to the vector with components proportional to $X_0(\rho) =$

Figure 1: The original test image. X-ray image of a person with ingested coin.
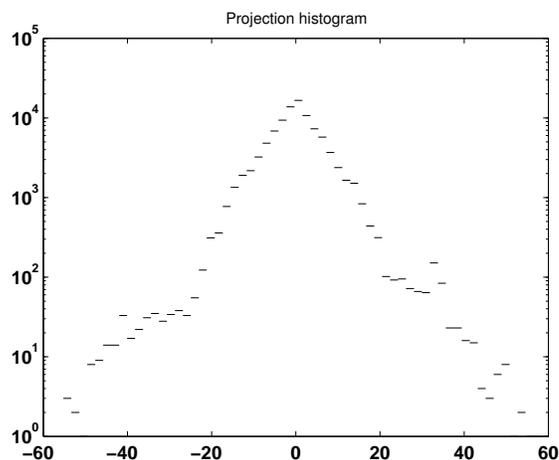
Figure 2: The distribution of the projection value for square shape with a size 48x48 pixels.

$1, \forall \rho$. The projection on the direction of X_0 is proportional to the mean brightness of the area and thus can be considered as not so important characteristics of the image. An alternative interpretation of the above statement is by considering all blocks that differ only by their brightness to be equivalent.

Mathematically the projections orthogonal to X_0 have the property:

$$\sum_{\vec{\rho}} X(\vec{\rho}) = 0. \quad (4)$$

The distribution of the values of the projections satisfying the property (4) is well known and universal [9] for the natural images. The same distribution
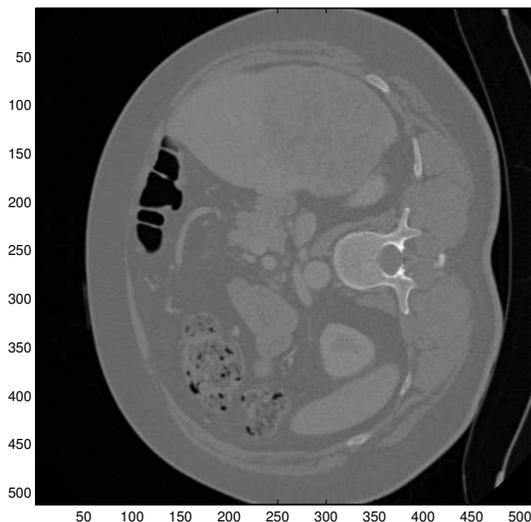
Figure 3: An intersection of the 3d-test image.
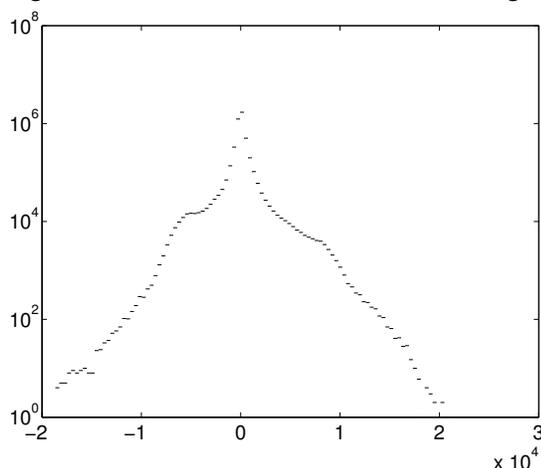
Figure 4: The distribution of the projection value for the corresponding 3d image.

seems to be valid for a vast majority of the images. The distribution of the projections derived for the X-ray image, shown in Fig. 1, is shown in Fig. 2.

In the case of a three-dimensional image, Fig. 3, the corresponding histogram, obtained by using the above method is shown in Fig. 4. One observes a higher asymmetry of the distribution of the projections in this case, compared to the same distribution of two-dimensional images, but qualitatively it is of the same type.

Roughly speaking, if the blocks are small enough, the distribution satisfies a power law distribution with exponential drop at the extremes. When the blocks are big enough, the exponential part is predominant.

If A_r and A_r' have similar projections, then they will belong to one and the same or to neighbors bins.

Therefore we can look for blocks that have a minimal number of similar and large projections. But these, due to the universality of the distribution, are exactly the blocks with large projection values.

As a first approximation, we can just consider the projections and score the points according to the bin they belongs to. The distribution can be described by only one parameter that, for convenience, can be chosen to be the standard deviation σ_X of the distribution of $X.B$.

The notion of "large value of the projection" will be different for different projections but will be always proportional to the standard deviation[2]. Therefore we can define a parameter a and score the blocks with $|X.B| > a\sigma_X$.

2.2 Algorithm

Resuming, in order to find the most unusual blocks of shape S in an image A we purpose the following Algorithm:

0. Construct a figure B with the same shape as A and with all pixels equal to zero.

1. Generate a random projection operator X, with carrier with shape S, zero mean and norm one.

2. Project all blocks (convolute the figure). We denote the resulting figure as C.

3. Calculate the standard derivation σ_X of the result of the convolution.

4. For all points of C with absolute values greater than $a\sigma_X$, increment the corresponding pixel in B.

Repeat steps 1-4 for M number of times.

5. Select the maximal values of B as the most singular part of the image.

As illustration we give separately in the Appendix the corresponding Matlab code.

The acceptable values of a are discussed in the next section. The number of iterations M can be fixed empirically or until the changes in B, normalized by that number, become insignificant. Following the algorithm, one can see that the time to perform it is proportional to $MN^d \log N$. The speed per image of size 1024×2048 on one and the same computer, with S, a square of size 56×56 points, is about 3 seconds compared to about an hour, using the direct search implementing the Eq. (1). [3]

---

[2]In general, the standard deviation will be larger for projections with larger low-frequency components. That is why we choose the criterion proportional to σ_X and not as an absolute value for all projections X.

[3]If the block is small enough, the convolution can be performed even faster in the space domain and it is possible to improve the execution time.

24x24   32x32

$a = 8$ $a = 10$

40x40   56x56

$a = 12$ $a = 16$
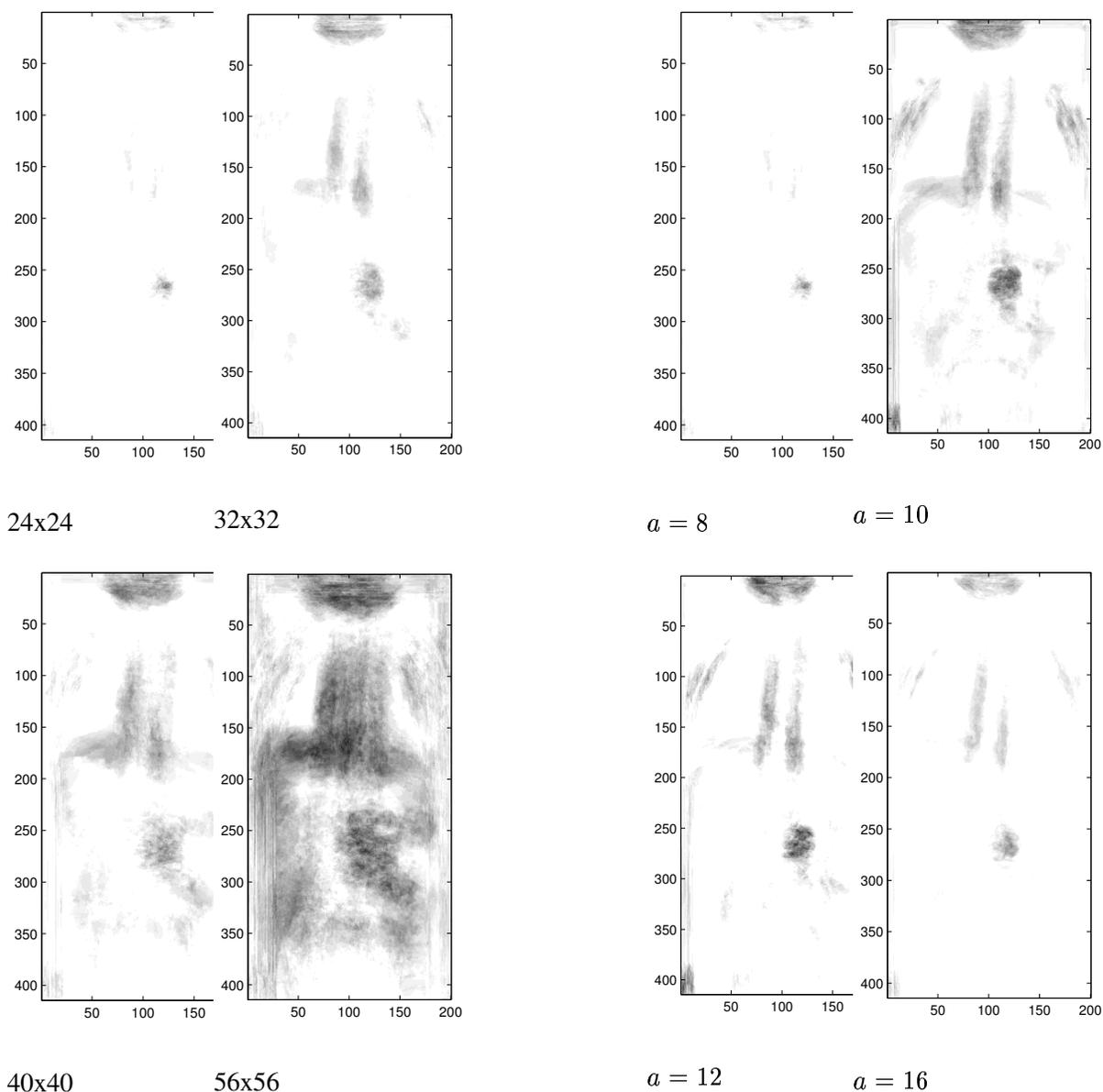
Figure 5: Score values for different size of the shape (24x24, 32x32, 40x40, 56x56). The value of the parameter a in all the cases is 12.

Figure 6: Score values for different parameter a ($a = 8,10,12,16$). The size of the shape is 24x24.

3 Empirical Assessment

Applying the algorithm above, we are looking for the most unusual part of the image in different settings. We also generalize the method in order to improve it and in to amplify the range of applications.

3.1 Two dimensional images

Some results are presented in Figs.5,6, where we used square shapes with different size, 30 projection operators and different values of a.

Because the distribution of the projections (Fig. 2) is universal, it is not surprising that the algorithm is operational for different images. We have tested it with some 100 medical Xray images and the results of the visual inspections were good.

It can be noted that the number of projection operators is not critical and can be kept relatively low and independent of the size of the block. Note that with significantly large blocks, the results can not be regarded as en edge detector. This empirical observation is not a trivial result at all, indicating that the degrees of freedom are relatively few, even with large

enough blocks, something that depends on the statistics of the images and can not be stated in general. With more than 20 projections we achieve satisfactory results, even for areas with more than 3000 pixels (some 10^5 in 3D). The increment of the number of the projections improves the quality, but with more than 30 projection practically no improvement can be observed.

A phenomenological argument can be given, observing that in the case of 30 projections, the pixels with maximal values are larger then 5. In order to distinguish a binary criteria (unusual/usual) this value is satisfactory large.

It is possible to look at that algorithm in a different way. Namely, if we are trying to reconstruct the figure by using some projection operators X_C (for example DCT as in JPEG), then the length of the code, one uses to code a component with distribution like Fig. 2, will be proportional to the logarithm of the probability of some value of the projection $X_C.A$. Therefore, what we are scoring is the block that has some component of the code larger than some length in bits (here we ignore the psychometric aspects of the coding). Effectively we score the blocks with longer coding, e.g. the ones that have lower probability of occurrence.

Using a smoothed version of the above algorithm in step 4, without adding only one or zero, but for example, penalizing the point with the square of the projection difference in respect to the current block divided by σ, and having in mind the universal distribution of the projection, one can compute the penalty function as a function of the value of the projection x, that results to be just $1/2 + x^2/2\sigma^2$. Summing over all projections, we can find that the probability of finding the best block is approximated given by $1/2[1 + \mathrm{erfc}(M(1/2 + x^2/2\sigma^2))]$ as a consequence of the Central Limit Theorem. The above estimation gives an idea why one need few projections to find the rarest block, in sense of the global distribution of the blocks, almost independently of the size of the block. The only dependence of the size of the blocks is given by σ^2 factor, that is proportional to its size. Further, the probability of error will drop better than exponentially with the increment of M.

The non-smoothed version performs somewhat better that the above estimation in the computer experiments.

3.2 Three dimensional images

Further we investigate how the algorithm works in 3D. As noticed in the previous section, the distribution of the projection is similar, but more irregular and asymmetric.
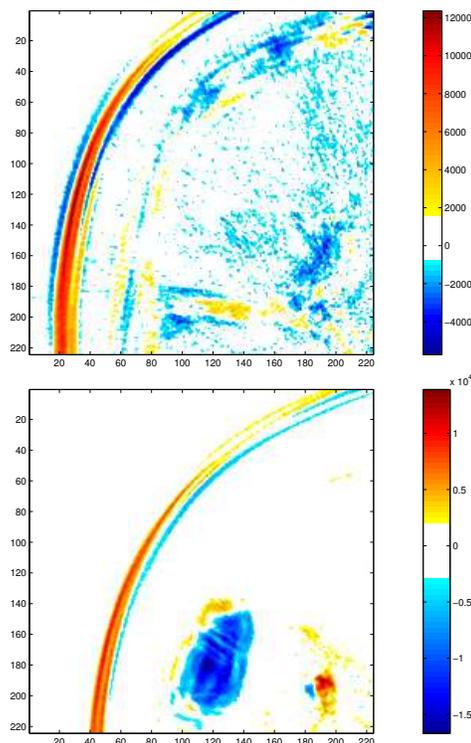
Figure 7: The upper panel shows the 3D structure, treated with 3D shapes. The lower panel shows the same structure treated with 2D shapes (at the same value of Z axes). The size of the block is 24×24 and $24 \times 24 \times 24$ correspondingly. The figures show that if the structures are clearly 3D the detection with 3D shape is better.

We have noticed that in 3D the method works better by using spherical shape, instead of cubic one. Most probably this is due to the fact that in the cube the most distant boundary voxel is $\sqrt{3}$ times further that the closest boundary voxel. This is significantly more that $\sqrt{2}$ as it is in the case of 2D pixels, although some "squaring" effect can be noted also in 2D (See Fig.9, $\gamma = 0.5$).

Comparing the quality of the method on two and three-dimensional images, (Fig. 7), one can say that when the structure is clearly three dimensional as the colon, the algorithm working in 3D separates this structure much better than working in 2D section. The irrelevance of the dimension for the algorithm is probably the main advantage in respect to other algorithms, as for example The Hough transform [10]. The maximum execution time scales with N as the number of pixels $MN^d \log_2 N^d$. Once again in 3D, as in the 2D case, M can be chosen very modest, about 30. The memory cost is four times the memory needed to save a single image, using naive FFT implementation
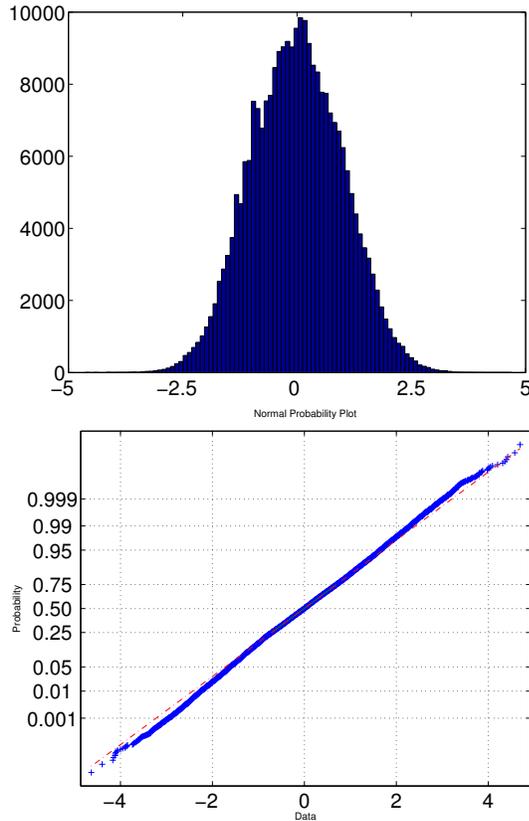
Figure 8: The upper panel shows the histogram of the normalized projection $X'_B = X_B/\sigma_B$. The lower one shows the quintile norm-plot. The distribution fits very well with the normal distribution.

of the convolution.

3.3   Contrast

However, there is an evident objection against the algorithm purposed. Namely, if some area of the image with high contrast is selected, then the projection is proportional to the contrast of that area. This will actually select the most contrast areas as the most unusual ones. Mathematically this is in fact so, but for practical reasons the dependence of the contrast could be eliminated or at least attenuated with similar argumentation as the one we have used for the brightness.

Namely, two images that differ only by their contrast could be considered as equivalent. To eliminate the influence of the contrast, the best is to normalize the projections using the contrast of the block. Let us regard as a contrast the standard deviation σ_B of the block B in question. Then the change in the algorithm is just evident: substitute each of the projections (3) with its normalized value $X'_B \equiv X_B/\sigma_B$. However, the distribution X'_B is no longer similar to that

shown in Fig. 4. The distribution is just normal [11]. To illustrate this, we represent the distribution and its quartile normal-plot in Fig.8 [4].

Using this normalization procedure makes the algorithm sensible to the noise, converting the flat noisy areas to the most unusual ones because of the randomness of the noise. Also the contrast, as an important characteristics, is better to be preserved in the normalized projection. Therefore it is much better not to eliminate the dependence of the contrast, but just to attenuate it. We found that using

$$X_B(\gamma) \equiv X_B/\sigma_B^{\gamma}$$

with different exponents γ-s serves well in order to give an appropriate weight of the contrast. When $\gamma = 0$ we have the case of uncorrected projections, while when $\gamma = 1$, the effect of the contrast is totally eliminated. Also γ can be assumed to be the tradeoff between the texture and the shape of the area. In Fig.9 we represent the results for different γ-s ($\gamma = 0, 0.5, 1, 1.5, 2$). We can see that different structures are highlighted dependent on the values of gamma. As expected, low values of γ (with relatively small shapes) accentuate the shape and high values of γ - the texture.

3.4   Conditioning

A case of special practical interest is to find the rarest (and the most similar) part of the image with respect to some database of images or with respect to a single image. Therefore, we are looking for the conditional probability of the occurrence of the blocks with respect to that database/image.

If we condition to one and the same test image A_p we can do it in a way, taking from the image A_p the patterns with a shape S in a random manner. That means to change step 1 of the algorithm to the following:

1c.   Select at random a point of A_p as an origin of the shape S. Use this area as a projection operator (normalizing and subtracting the mean brightness).

In this way we can answer both questions in the same time - What is the part of the image A with shape S that is most similar to the image A_p and what is the part of the image A most dissimilar to the parts of the image A_p. The first answer is related to pattern recognition problem. As an example, we are looking for a colon on the CT image shown in Fig.3, the test image can look like Fig.10.

---

[4]The distribution ought to be tested with caution because the low-pass filtering will flat the top of the distribution. Also the precision of the pixels ought to be at least 2 bytes in order to avoid rounding errors.

$\gamma = 0$

$\gamma = 0.5$

$\gamma = 1$

$\gamma = 2$
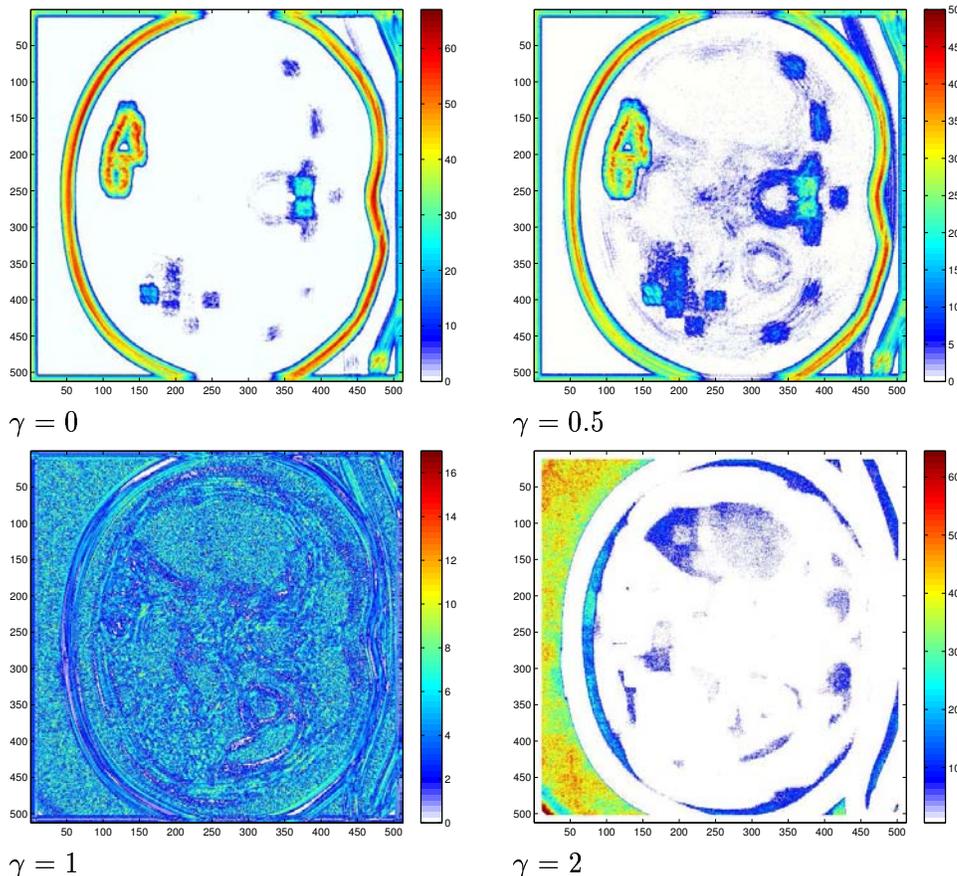
Figure 9: Different normalizations corresponding to the values of $\gamma = 0, 0.5, 1, 2$. The influence of the borders diminished and the influence of the texture increases.
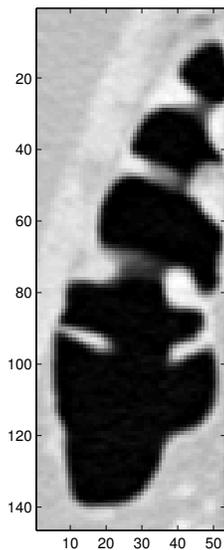
Figure 10: The figure of patterns we use to find similar and dissimilar parts of the image.

If the shape S is small, then the statistics would be more or less universal and we cannot expect that the result would be very specific to the image A_p. If se increase the size of the shape S the result will be more and more specific. If the size of the S is similar to the size of A_p we can expect highly specific response. In order to achieve satisfactory result we ought to use $\gamma \approx 1$, e.g. to eliminate the dependence on the contrast.

The results of the conditioning are shown in Fig.11.

We condition one image of the colon Fig. 3 to the other Fig.12. We find that the recognition is very good. It does not depend on the dimensionality of the image. The first two panels of Fig.11 with smallest S, 16×16 and 32×32, have strong mixture between negative and positive large projections (not shown in the figure). It accentuates more or less the borders, but with mixed sign of the projections. The last image has only positive correlations with the test image in the upper left corner. The position of the colon (See Fig.3) is detected correctly.

16×16

32×32

32×64

32×128
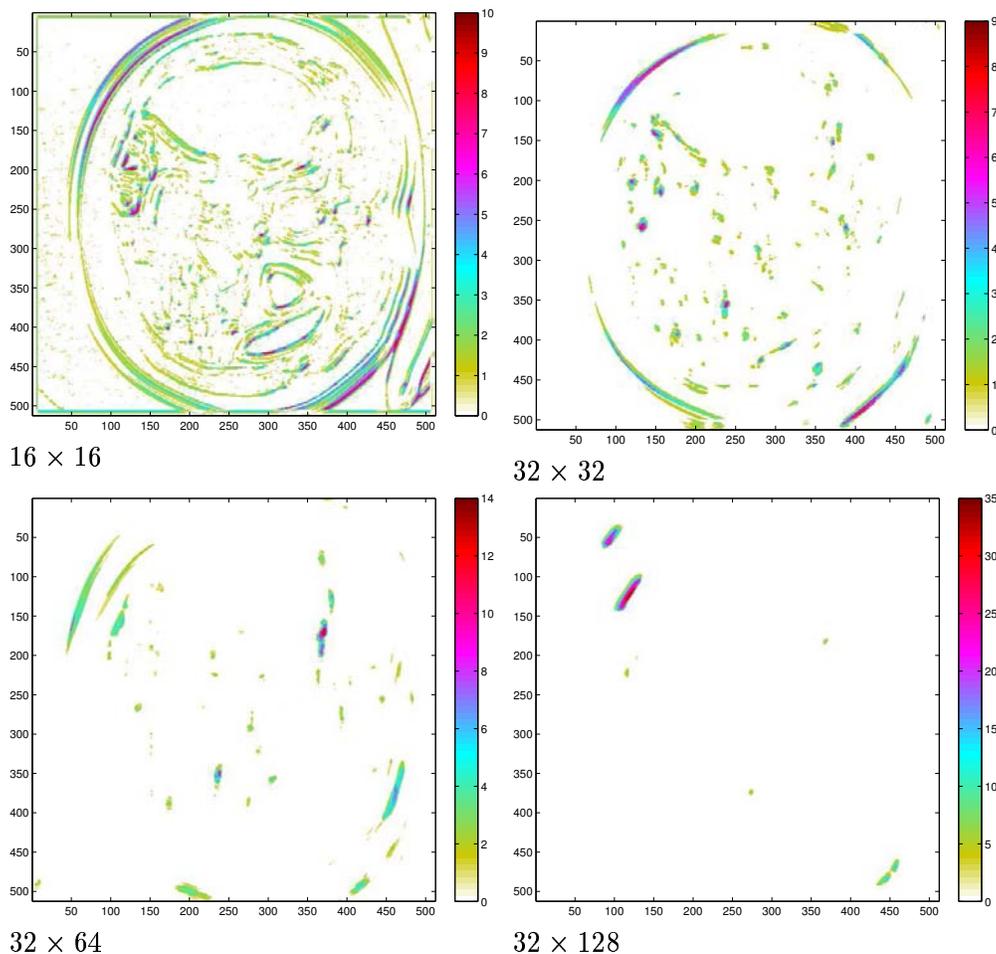
Figure 11: Finding the most similar part using the algorithm. The test image has size of 54×145. The size of the rectangle block S is shown for each image.

4 Conclusions

In this paper we present a method to find the most unusual (rare) part in two and higher dimensional images, when its shape is fixed, but in general arbitrary. The method is almost independent on the size of the shape in terms of the execution speed and time. It gives good results on experimental images without predefined model of the interesting event, working equally well for 2D and 3D images.

The algorithm uses finite and low dimensional space instead of the huge dimensional space of the image segments, where each point forms another dimension. Therefore, it can be used to scan image databases independently of its size, as for example medical databases.

We would also like to comment that the method requires to save only the large projections of images and this relies significantly the search in a large image dabase. Actually the storage requirements are very modest and this permits the searching in very large databases. This is suitable for medical purposes, because the quantity of even one digital X-ray device can provide hundreds of images per day.

Among the future applications of the present method, one could mention the achievement of experiments on different type of images and large image databases and experiments on acceleration of the network due to the special equivalence class construction.

We are currently working on the extension of the method reformulating it in terms of neural networks of Hebbian type. The preliminary results show a good quality and fast execution time for the localization of the most unusual parts of 2d and 3d images [12].

5 Appendix

The program we used is the following:

```
function [ba,bb,bu]=findrare(fname,
sx1,sx2,s1,iter)
if nargin < 5, iter = 30; end
if nargin < 4, s1 = 22; end
if nargin < 3, sx2 = 32; end
if nargin < 2, sx1 = 32; end
bu=imread(fname);
b=double(bu); b=b(:,:,1);
[i1 i2]=size(b);
bp(1:i1*i2)=b;
% it is more convinient to have
2D and 1D vector.
% normalize
b=b-mean(bp);
b=b/std(bp);
bp=bp-mean(bp);
bp=bp/std(bp);
for ii=1:iter
ii;
r1=rand(sx1:sx2);
% generate random projection
matrix
[i1 i2]= size(r1);
% normalize
r1p(1:i1*i2)=r1;
r1=r1-mean(r1p);
r1=r1/std(r1p);
% convolute
b1=conv2(b,r1);
% edge effects and size change.
[i1 i2]=size(b1);
b1p(1:i1*i2)=b1;
% collect only large
projections.
if ii==1
bips=(b1p>s1*2);
else
bips=bips+(b1p>s1*2);
end
bips=bips+(b1p<-s1*2);
end
% hist(b1p,60);
% postprocess and form some
auxiliar results.
[ia1 ia2]=size(b1);
b1res=reshape(bips,size(b1));
bvar=conv2(b,ones(sx1,sx2))\
/sx1/sx2;
bsqr=b .* b;
bvar=sqrt(conv2(bsqr,ones(sx1,sx2))\
/sx1/sx2-(bvar ./ bvar));
```

```
b3=0.5 * b1res ./
real(abs(bvar));
ba=b1res(sx1/2:ia1-sx1/2,sx2/2:\
ia2-sx2/2);
bb=b3(sx1/2:ia1-sx1/2,sx2/2:\
ia2-sx2/2);
end
```

References:

[1] E. Keogh, Lin J. and Fu A., HOT SAX: efficiently finding the most unusual time series subsequence, in the *Proceedings of 39th ACM Symposium on Theory of Computing*, 2007.

[2] K. Koroutchev and E. Korutcheva, Detecting the most unusual part of a digital image, *Combinational Image Analysis,* LNCS 4958, 2008, pp. 286-294.

[3] M. Benvenuti et al., Tracking of Moving Targets in Video Sequences, *WSEAS TRANSACTIONS on SYSTEMS*, Volume 4, No 4, 2005, pp. 359-368.

[4] Pei-Jun Lee, Real-Time Homecare Service System Based on Color Image Analysis, *WSEAS TRANSACTIONS on SYSTEMS*, Volume 6, No 5, 2007, pp. 1009-1018.

[5] S. Rahnamayan, Towards Incomplete Object Recognition, *WSEAS TRANSACTIONS on SYSTEMS*, Volume 4, No 10, 2005, pp. 1725-1734.

[6] K. Shihab, Probabilistic Graphical Models for Dynamic Systems, *WSEAS TRANSACTIONS on SYSTEMS*, Volume 4, No 6, 2005, pp. 830-839.

[7] Y. Fisher, *Fractal Image Compression*, ISBN 0387942114, Springer Verlag, 1995.

[8] P. Indyk, Uncertainty Principles, Extractors, and Explicit Embedding of L2 into L1, in the *Proceedings of 39th ACM Symposium on Theory of Computing*, 2007.

[9] D. Ruderman, The statistics of natural images, Network : *Computation in Neural Systems,* 5, 1994, pp.517-548.

[10] P. Hough et al., Method and Means for Recognizing Complex Patterns , U.S. Patent 3,069,654, 1962.

[11] K.Koroutchev, J.R.Dorronsoro, Factorization and structure of natural 4 x 4 patch densities, *Optical Engineering,* 45, 2006, pp. 127003-127009.

[12] K. Koroutchev and E. Korutcheva, in preparation.