Genetic Normalized Convolution

Giulia Albanese¹, Marco Cipolla², and Cesare Valenti²

Dipartimento di Scienze dell'Informazione Università di Bologna, via Mura Anteo Zamboni 7, 40127, Italy galbanes@cs.unibo.it

Dipartimento di Matematica e Informatica Università di Palermo, via Archirafi 34, 90123, Italy {cvalenti,mcipolla}@math.unipa.it

Abstract. Normalized convolution techniques operate on very few samples of a given digital signal and add missing information, trough spatial interpolation. From a practical viewpoint, they make use of data really available and approximate the assumed values of the missing information. The quality of the final result is generally better than that obtained by traditional filling methods as, for example, bilinear or bicubic interpolations. Usually, the position of the samples is assumed to be random and due to transmission errors of the signal. Vice versa, we want to apply normalized convolution to compress data. In this case, we need to arrange a higher density of samples in proximity of zones which contain details, with respect to less significant, uniform parts of the image. This paper describes an evolutionary approach to evaluate the position of certain samples, in order to reconstruct better images, according to a subjective definition of visual quality. An extensive analysis on real data was carried out to verify the correctness of the proposed methodology.

1 Introduction

Normalized convolution is a signal processing method that allows to reconstruct an image when just a few pixels are available due to the presence, for example, of noise or instrumental error. These pixels, also called certain samples, are assumed to have an uniform random distribution. Adaptive normalized convolution consists in a pipeline of normalized convolutions to improve the overall quality of the resulting reconstruction, though it requires a lot of computation.

Aim of this paper is the robustness evaluation of normalized convolution for data coding and compression purposes. In this case, we want to establish the correct amount of samples, together with their positions, needed to reconstruct images having an high perceived quality. In particular, we present a genetic algorithm able to locate these samples, according to an attentive model based on edges and centers of symmetry that usually correspond to details in the scene. A recent metrics has been applied to measure the subjective quality of the solution, trough the analysis of luminance, contrast and structures.

We experimentally verified on a database of real images that our method outperforms classic normalized convolution. Moreover, it can be used as a preprocessing step to enhance adaptive normalized convolution, too.

G. Maino and G.L. Foresti (Eds.): ICIAP 2011, Part I, LNCS 6978, pp. 670–679, 2011.

[©] Springer-Verlag Berlin Heidelberg 2011

Section 2 briefly describes both normalized convolution and its adaptive variant. Section 3.1 sketches phase congruency and radial symmetry transforms to detect regions of interest in images. The structural similarity metrics is reported in section 3.4 as part of the fitness function of our genetic algorithm, introduced in the rest of section 3. Experiments and conclusions are reported in sections 4 and 5.

2 Normalized Convolution

Normalized convolution represents an important tool in the digital signal processing field. It was described for the first time by Knutsson and Westin [1] who pointed out the opportunity to provide also a confidence measure of the available samples. Actually, a map should indicate the presence degree of a sample in a given position. In particular, a binary map would indicate just the absence or presence of the signal. The underlying theory is simple and its implementation appears quite fast.

Let S be the positive map that represents certain samples of a digital image I. If we indicate by $\{S \cdot I\}$ the pixelwise product of S and I and by $\{K * I\}$ the usual convolution with a kernel K, then normalized convolution is defined by

$$NC(I, S, K) = \{K * S \cdot I\} / \{K * S\}$$

In other words, to reconstruct the whole image I from its samples specified in S, we just have to weight $\{K*S\cdot I\}$ by the confidence $\{K*S\}$ of the results generated.

The kernel, centered in the origin, is usually defined as a Gaussian-like surface

$$K_{x,y} = \begin{cases} r^{-2} \cos^2\left(\frac{\pi r}{2r_{\text{max}}}\right) & \text{if } r < r_{\text{max}} \\ 0 & \text{otherwise} \end{cases}$$

where $r = \sqrt{(x^2 + y^2)}$. To avoid over-smoothing the output image, K should be big enough to contain just some pixels of the input signal. Vice versa, if the distance between the nearest samples in S is greater than the size $r_{\rm max}$ of K, then the reconstructed image will contain gaps. Without a priori information, $r_{\rm max}$ is automatically set to the minimal distance among the available samples to reduce artifact effects along edges: at least one pixel lies always within the radius. It is noteworthy that fast implementations can be developed for both convolving images with very few samples and computing the distance among these samples.

A variant of this algorithm is known as adaptive normalized convolution and it modulates both the size and the shape of the kernel K, according to the position of certain samples [2,3]. In this case, implementing an optimized and efficient custom convolution routine can be quite difficult. Indeed, a different filter can be arranged for each pixel of the output image and an estimate of the gradient of the whole image is used to determine this proper kernel. Obviously, this gradient itself is just an approximation since it has to be computed from available

samples specified by S. This preprocessing step, known as derivative of normalized convolution or normalized differential convolution, requires a considerable amount of computational time. Actually, we do not to use the adaptive normalized convolution technique because its performances usually does not justify the enhancement of the final output image, which we aim to achieve by better positioning the samples in S. Anyway, adaptive normalized convolution can be applied with the sample positions returned by our method.

3 Genetic Normalized Convolution

Genetic algorithms have been already applied to the inpainting problem (i.e. automatic completing missing areas and spatiotemporal restoration by using image samples) [4], but, to our knowledge, this is the first description of a method that combines normalized convolution and an evolutionary approach. We have not modified this former algorithm which will be used to code and compress data. We implemented a genetic algorithm to locate the best position of certain samples. We desire to hold all details in the image by assigning most information to them. In order to decide if such parts of the image should be considered as a regions of interest, we used models suggested by the Gestalt theory. Anyway, enough samples still have to be devoted to uniform zones, which are usually due to the background. The following section regards basic algorithms helpful to identify important details in images.

3.1 Regions of Interest

Regions of interest detectors are usually applied to selectively process images, to locate peculiar features and to simulate active vision systems. Many definitions of regions of interest do exist and provide complementary information as edges, corners, blobs and symmetries. In particular, we will sketch a fast radial symmetry transform and an effective edge detector. These methods are the basis for a combined detector to manage the positions of certain pixels needed by the normalized convolution process.

A variety of algorithms to locate centers of radial symmetry in digital images have been described in the literature. Some of these methods require custom hardware or do complex filtering within big windows. Often, the size of symmetric zones is unknown, thus multiresolution pyramidal approaches were proposed too [5]. A few methods reduce the computational time by letting the gradient of the image drive the analysis in some way [6]. In particular, we used the fast radial symmetry transform [7] because it returns in real time an output coherent with present attentive human models. The key idea is to use an accumulation map, as for the Hough transform, which highlights the contribution of the gradient vector field. If a point is a center of symmetry, then it receives a degree proportional to the number of gradient vectors heading to it (see figure 1a). Actually, the radius of the scanning window, that is the length of the vectors to consider, should be set by the user, but satisfactory results can be obtained by merging

together the outputs corresponding to a set of different radii (figure 2b shows an example with radius from 1 to 11 with an incremental step equal to 2).

In [8] it was presented an algorithm to locate both corners and edges from phase congruency information. This is particularly robust against changes in illumination and contrast because contours have many of their frequency components in the same phase, as in figure 1b. Moreover, these edges present a single response as line features, are less prone to the presence of noise and are largely independent of the local contrast (see figure 2c). These observations are due to the fact that this algorithm is not based on a first-derivative operator (on the contrary of Canny or Sobel filters) which usually exalts also the contribution of negligible details. The whole method is computationally intensive, but fortunately we need to apply both phase congruency and the above symmetry detector just at the beginning of our genetic algorithm to create a regions of interest mask that comprises both edges and centers of radial symmetries.

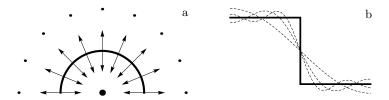


Fig. 1. (a) dots represent the degree of radial symmetry pointed by gradient vectors. (b) Fourier components, as dashed lines, share the same zero crossing of a step edge.

3.2 Population

Given an image I, our algorithm creates a corresponding mask ROI that measures the pixels' 'importance'. This mask is the superimposition, normalized in [0,1], of edges and centers, obtained trough the methods described in section 3.1. To reduce the effect of pointlike noise and to spread the regions of interest, a Gaussian convolution, with radius equal to 19 and standard deviation equal to 6.3, is performed. The pixelwise product of the resulting mask with its thresholded version (on the average value), puts in evidence all details [9,10,11] (figure 2d).

Each genome A of the population encodes co different indices with values in $[1, n^2]$, where the image has exactly $n \times n$ pixels and co samples will be used to reconstruct it. Even if a convenient coding would be the use of whole permutations of n^2 elements, we limited the genome dimension to co, thus reducing drastically the storage complexity of the algorithm. Indeed, we want to remark that usually $co \ll n^2$.

The initial set of ni random individuals have fd genes constrained on details highlighted by the mask ROI. This trick is not necessary and does not follow

any evolutionary strategy, but we verified that it reduces the number ng of generations required to reach a stable solution. It must be noted that our algorithm does not constrain the number of samples on details during the evolution of the population. The cardinality ni of the population, the quantity fd of details and further parameters will be considered in section 4.

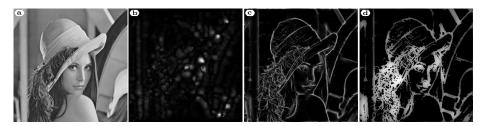


Fig. 2. (a) input image. (b) radial symmetries. (c) phase congruency edges. (d) regions of interest map.

3.3 Selection and Genetic Operators

During the development of our algorithm, we considered a variety of selection methods (e.g. roulette wheel, ranking and random) [12] to pick the chromosomes for crossover. The final version of the system includes only a 4-way tournament, without duplication of already selected individuals, because other strategies returned poor results. An elitist selection is also applied to assure that the worst chromosome will be eliminated and the best one be present as two copies in the next population too. Although this approach reduces the variability of the hereditary characteristics, we ascertained that it increases the result accuracy.

Since our individuals store just part of permutations, we developed ad hoc tools instead of using standard genetic operators for permutations (e.g. CX, OX and PMX). Let us indicate with $A \cup B$ and $A \cap B$ respectively the union and intersection sets of genes owned by two parents A and B. Two offsprings C and D are obtained by joining the indices in $A \cap B$ with half of the indices in $(A \cup B) \setminus (A \cap B)$, randomly chosen (see figure 3). This operation is not subject to any probability because, due to data structure optimizations, we always force the same percentage pc of chromosomes to undergo crossover.

```
A = (\underline{26} \ 11 \ \underline{30} \ \underline{9} \ \underline{15} \ 24 \ 14 \ 27 \ 5 \ 34)
B = (21 \ 6 \ \underline{30} \ \underline{15} \ \underline{9} \ 29 \ 19 \ \underline{26} \ 10 \ 16)
P((A \bigcup B) \setminus (A \cap B)) = (29 \ 16 \ 10 \ 19 \ 27 \ 21 \ | \ 6 \ 11 \ 34 \ 14 \ 5 \ 24)
C = (\underline{26} \ 29 \ \underline{30} \ \underline{9} \ \underline{15} \ 16 \ 10 \ 19 \ 27 \ 21)
D = (6 \ 11 \ \underline{30} \ \underline{15} \ \underline{9} \ 34 \ 14 \ \underline{26} \ 5 \ 24)
```

Fig. 3. A crossover example. Common genes in A and B are underlined. P indicates a random permutation of the indices.

In the case of complete permutations, mutation is accomplished usually by swapping a pair of random indices. We cannot apply this method because, while the order of the indices in an individual is irrelevant, we must ensure to replace a gene already coded with another one that is not in the genome yet. This operation corresponds to move just a sample to another position. We introduced the parameter mg to guarantee that an adequate number of genes are modified to significantly change the content of the image. Moving mg samples in a completely random fashion does not improve the overall quality of the reconstructed image, because, on average, the same number of samples remain in the same zones of the image. To slightly change the aspect of the scene (see figure 4), we verified that a better strategy consists in moving mg samples within their own neighborhoods of radius 3. As for crossover, this operation is not subject to any probability because we force always the same percentage mg of genes to change and the same percentage pm of chromosomes to undergo mutation.

$$A = (26\ 11\ 30\ \underline{9}\ 15\ 24\ 14\ \underline{27}\ 5\ 34)$$

 $B = (26\ 11\ 30\ \underline{8}\ 15\ 24\ 14\ \underline{29}\ 5\ 34)$

Fig. 4. A chromosome A and its mutated version B. Small perturbations, underlined here, usually occur.

3.4 Quality Metrics and Fitness Function

Many metrics were defined to evaluate the similarity between images. For example, mean square error and its variant peak signal-to-noise ratio are widely applied to quantify pixelwise differences between two images, though this approach does not satisfy a real human perception system. That is, it can be proved that very altered versions of the same image still present almost identical PSNR values. It must be noted that the bigger this metrics is, the better the image is perceived. Generally speaking, a medium quality image should have a PSNR value not smaller than 30dB.

A recent subjective metrics, known as structural similarity, was introduced in [13] to compare local patterns of pixel intensities, after luminance and contrast normalization trough a z-score function. This method assumes that the main structures in the scene are independent from luminosity, which should be isolated. The similarity index between two images with ℓ gray levels, means μ and standard deviations σ is given by

$$SSIM(I_1, I_2) = \frac{(2\mu_1\mu_2 - \kappa_\mu)(2\sigma_{12} + \kappa_\sigma)}{(\mu_1^2 + \mu_2^2 + \kappa_\mu)(\sigma_1^2 + \sigma_2^2 + \kappa_\sigma)}$$

where σ_{12} represents the covariance of the images and the constants $\kappa_{\mu} = \varepsilon_{\mu} (\ell - 1)^2$ and $\kappa_{\sigma} = \varepsilon_{\sigma} (\ell - 1)^2$, with $0 < \varepsilon_{\mu} \ll 1$ and $0 < \varepsilon_{\sigma} \ll 1$, avoid instability when $\mu_1^2 + \mu_2^2$ or $\sigma_1^2 + \sigma_2^2$ is close to 0.

Again, the bigger the value of this metrics is, the better the image is perceived. Assessments derived from SSIM keep count of the whole image and are closer to subjective judgments. Sometimes, PSNR provides very different values in the case of images which can be considered almost identical and, therefore, with similar SSIM values. For the sake of completeness, we compare the reference image and the reconstructed one by both SSIM and PSNR.

Our fitness function f to evaluate a chromosome A is just the inner product, normalized to the number of pixels, between the regions of interest mask (see section 3.1) and the SSIM value of the original image I and its reconstructed version, obtained trough the map of samples coded in A

$$f(A) = \langle ROI(I), SSIM(I, NC(I, S(A), K)) \rangle / n^2$$

Better individuals correspond to higher values of f. Despite its simplicity, this formula provided enough variability to produce good results for all images we considered. The population evolves until a stable solution is found or merely the maximum number of generation is reached.

4 Experimental Setup

We verified the performance of our methodology on a database of gray scale images, freely available on the Internet. We studied 50 images that contain

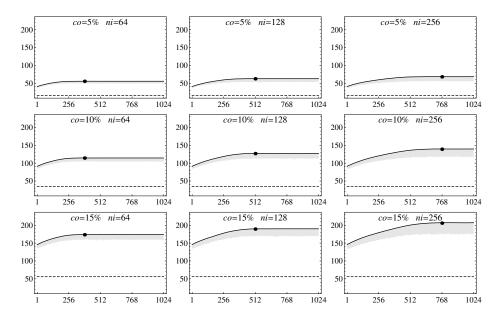


Fig. 5. Average fitness trend versus number of generations, with respect to number ni of individuals and percentage co of confident samples. The values of f lie within the gray band. Average fitness due to random samples is reported as dashed line. Suggested number of generations is indicated by the point.

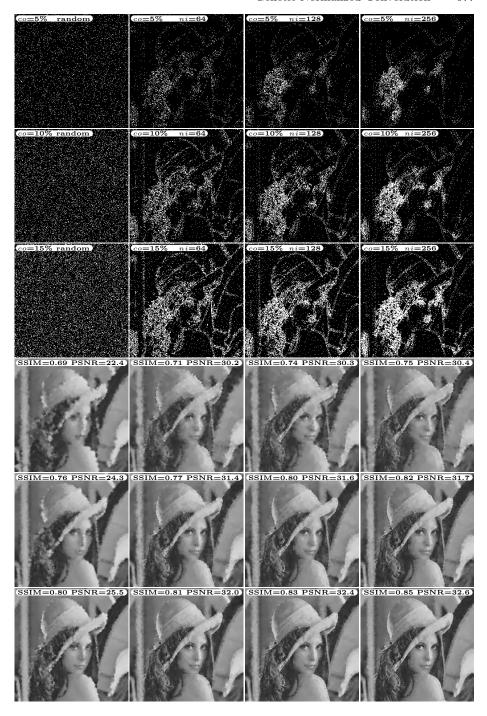


Fig. 6. Comparison among random samples and samples arranged by our algorithm

structured and textured scenes with 256×256 or 512×512 pixels. They were compared with the reconstructed versions by the SSIM quality metrics. Plots in figure 5 show the average fitness during the genetic evolution on the entire database. The gray band represents the minimum and maximum values of f. The average value is close to the best one (i.e. the maximum), thus most chromosomes really converge to the solution found by our method. Further parameters, as the percentage co of certain samples and the number ni of individuals are also reported. The average fitness of an image reconstructed from samples in completely random positions is depicted as a dashed line. Reconstruction examples are shown in figure 6. It is quite evident that our genetic algorithm is able to locate samples that improve the performance of the usual normalized convolution, regardless the values of ni and co.

Various experiments were carried out to fine tune the percentages $mg \in \{0.10, 0.40\}$ of genes to mutate, $fd \in \{0.10, 0.40\}$ of details to maintain, $pc \in \{0.70, 0.95\}$ and $pm \in \{0.05, 0.30\}$ of chromosomes for crossover and mutation, respectively. As a rule of thumb, mg = 0.25, fd = 0.15, pc = 0.95, pm = 0.05 allow to reduce the run time and to achieve satisfactory final images. For all these values, we studied the results corresponding to an incremental step equal to 0.05 and a number ni of individuals equal to 64, 128 and 256. In the case of lots of chromosomes, a better solution is expected but many generations are needed to let it stabilize. The following sets of suitable parameters, indicated trough a point in the plots of figure 5, can be used independently of the number co of certain samples (as percentage of the image size): ni = 64 and ng = 384, ni = 128 and ng = 512, ni = 256 and ng = 768.

5 Results and Conclusions

Normalized convolution is an important tool to reconstruct images from few samples and it is particularly useful when part of the information is lost due to noise or when a failure occurs during data transmission. Usually, these samples are assumed to be randomly distributed, in a uniform fashion. This paper considers normalized convolution for data coding and compression. In this case, we are interested in the reconstruction from a given number of certain samples or below an error threshold. A genetic algorithm has been presented to locate the best positions of the samples and an extensive experiment was carried out to verify its robustness on real images. In particular, an attentive model, based on phase congruency and radial symmetry, was used to create a proper initial population, while a structural similarity metrics was adopted to measure the perceived quality of the solution. It is noteworthy that this image can be slightly improved by using the usual adaptive normalized convolution on the samples chosen by our genetic algorithm, in spite of the increased amount of computations.

We developed our system in the interpreted high level MatLab language. This means that a few minutes are still necessary to elaborate a 256×256 pixels image, though a C/MPI compiled version of the program should be able to reach a feasible solution very quickly. We are already working on an improved, parallel

version of our genetic algorithm to get benefit from new multicore CPUs, available on present computers. We experimentally verified that the time complexity is roughly linear with respect to both the number of individuals and the size of the image to reconstruct. On the other hand, the number of certain samples does not cause any overload, due to the reduced quantity of genes coded by the chromosomes.

References

- Knutsson, H., Westin, C.: Normalised and differential convolution. In: IEEE Proc. CVPR on Computer Society Conf., vol. 15-17, pp. 515-523 (1993)
- Pham, T.Q., van Vliet, L.J.: Normalized averaging using adaptive applicability functions with applications in image reconstruction from sparsely and randomly sampled data. In: Bigun, J., Gustavsson, T. (eds.) SCIA 2003. LNCS, vol. 2749, pp. 485–492. Springer, Heidelberg (2003)
- 3. Pham, T.Q., van Vliet, L.J., Schutte, K.: Robust Fusion of Irregularly Sampled Data using Adaptive Normalized Convolution. Journal Applied Signal Processing (2006)
- Kim, E.Y., Jung, K.: Object Detection and Removal Using Genetic Algorithms. In: Zhang, C., W. Guesgen, H., Yeap, W.-K. (eds.) PRICAI 2004. LNCS (LNAI), vol. 3157, pp. 411–421. Springer, Heidelberg (2004)
- 5. Di Gesù, V., Valenti, C.: Detection of regions of interest via the Pyramid Discrete Symmetry Transform. In: Proceedings of International Workshop on Theoretical Foundations of Computer Vision (1997)
- Reisfeld, D., Wolfson, H., Yeshurun, Y.: Context Free Attentional Operators: The Generalized Symmetry Transform. Int'l J. Computer Vision 14, 119–130 (1995)
- 7. Loy, G., Zelinsky, A.: Fast Radial Symmetry for Detecting Points of Interest. IEEE Trans. on Pattern Analysis and Machine intelligence 25(8) (2003)
- Kovesi, P.: Phase Congruency Detects Corners and Edges. In: The Australian Pattern Recognition Society Conference: DICTA Sydney 2003, pp. 309–318 (2003)
- 9. Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the 7th International Conference on Computer Vision, pp. 1150–1157 (1999)
- Lindeberg, T.: Feature detection with automatic scale selection. International Journal of Computer Vision 30(2), 79–116 (1998)
- 11. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. International Journal on Computer Vision 60(1), 63–86 (2004)
- 12. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer, Heidelberg (1996)
- 13. Wang, Z., Bovik, A.C., Sheikn, H.R., Simoncelli, E.P.: Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Transactions on Image Processing 13(4), 600–612 (2004)