XYZ Symposium (2008)
N. N. and N. N. (Editors)

CGI 2009 proceedings, ACM ISBN 978-60558-687-8, pp.39-44, 2009

RBF-based Image Restoration Utilising Auxiliary Points

J. Zapletal[1] and P. Vaněček[1] and V. Skala[1]

[1]Department of Computer Science and Engineering, University of West Bohemia, Pilsen, Czech Republic

Abstract

Utilisation of Radial Basis Functions (RBF) for reconstruction of damaged images became common technique nowadays. This paper deals with computation and utilisation of auxiliary points in order to further increase the ability of RBF to restore damaged areas in image. Our goal was to achieve the best possible results in acceptable time of computation. We put stress mainly on cases, where the image is heavily damaged e.g. by extreme noise. In these cases our new proposed approach achieved very usable results that even surpassed our expectations.

Categories and Subject Descriptors (according to ACM CCS): G.1.1 [Numerical Analysis]: Interpolation; I.4.5 [Image Processing and Computer Vision]: Reconstruction

1. Introduction

Techniques of reconstructing of corrupted artworks are as old as the art itself. Since centuries ago the need of reparation of damaged artistic artefacts was present, e.g., cracks in wall paintings or damaged statues. The goal of skilled restorers was to repair such object in a non-detectable way for non-familiar observer. With the evolution of photo/movie industry in the last few decades, really massive call for similar services have arisen from many purposes. Typically the need of retouching images (advertisement), eliminating of objects (political reasons), denoising of signals etc. came out. Utilisation of computer technology was just a matter of time.

1.1. Related work

The seminal work of Bertalmio et al. [BS00] was based on partial differential equations. Corrupted areas are filled by propagation of intensities from surrounding area in isophotes direction, calculated along mask border. Oliveira et al. [OBMC01] used Gauss convolution kernel for fast filling of small areas. Due to capabilities like ability to interpolate scattered data or evaluate value anywhere in any resolution, radial basis functions [CBC*01] became quite popular also in this area [SKU02], [US05], [WWW06]. They also take advantage of reconstructing of all characters of damage, such as inpainting or noise. This is in contrast with the other techniques, specialised at particular cases only. Use of RBF for images restoration was proposed by Savchenko et al. [SKU02] who used CSRBF. We decided to utilise and extend RBF method due to previous works [US05], [ZVS08]. In this paper we will propose new approach to RBF based image reconstruction that highly enhances results.

2. Image restoration

Prior to the reconstruction itself, the area for restoring should be specified. There is a possibility of automatic detection of damaged area but in this case supplemental information about the image should be provided. Since we focused on the act of restoration itself (not on detecting of damage), we will suppose that the area for reconstruction is already defined by black (corrupted) and white (valid pixels) mask. This presumption is commonly used in image reconstruction since the work of Bertalmio et al [BS00].

Let $\Omega = \{(x,y) \mid x = 0,\ldots,M-1, y = 0,\ldots,N-1\}$ be a rectangle image area with resolution $M \times N$ and the pixel value $h[x,y]$ is defined for $\mathbf{x} \in \Omega$. Damaged image Ω is a combination of correct pixels Ω_c and defective pixels Ω_d (mask) such as $\Omega = \Omega_c \cup \Omega_d$, $\Omega_c \cap \Omega_d = \emptyset$. We would like to find all values h for all $\mathbf{x} \in \Omega_d$.

3. Radial Basis Functions

RBF [CBC*01], [SKU02] is a circularly symmetric function around its centre. Smooth function $f(\mathbf{x})$ is interpolated from

CGI 2009 proceedings, ACM ISBN 978-60558-687-8, pp.39-44, 2009

a set of n scattered points using n RBFs for a pixel $\mathbf{x}$ as:

$$f(\mathbf{x}) = \sum_{i=1}^{n} \lambda_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) + P(\mathbf{x}) \tag{1}$$

where $\mathbf{x}_i$ are all valid pixels and λ_i are weights of RBFs. A polynomial $P(\mathbf{x})$, generally of degree m, is added to ensure solution stability together with additional conditions $\sum_{i=1}^{n} \lambda_i = 0,\ \sum_{i=1}^{n} \lambda_i x_i = 0,\ \sum_{i=1}^{n} \lambda_i y_i = 0$. In our approach $P(\mathbf{x}) = \gamma^T \mathbf{x} = \gamma_1 x + \gamma_2 y + \gamma_3$ was taken. This leads to a system of linear equations:

$$\left[\begin{array}{c|c} \mathbf{A} & \mathbf{P} \\ \hline \mathbf{P}^T & \mathbf{0} \end{array}\right] \left[\begin{array}{c} \lambda \\ \hline \gamma \end{array}\right] = \left[\begin{array}{c} \mathbf{h} \\ \hline \mathbf{0} \end{array}\right] \tag{2}$$

where $\mathbf{A}_{i,j} = \phi(r_{ij}) = \phi(\|\mathbf{x}_i - \mathbf{x}_j\|),\ i,j = 1,\ldots,n$, $\mathbf{P}$ is $n \times 3$ matrix of coordinates of valid pixels and finally $\mathbf{h}$ is the vector of pixel values h_i.

Selection of RBF significantly affects the result of interpolation (see submatrix $\mathbf{A}$ in 2). Popular RBFs are: Cubic r^3, Gauss $\varepsilon^{-(\varepsilon r)^2}$, Linear r, Multiquadric $\sqrt{1+(\varepsilon r)^2}$ and TPS $r^2 \log r$. The parameter ε furthermore defines behavior of the function, lowering ε value improves stability of linear system. We chose linear RBF due to our previous experiments, where this RBF demonstrated very good results.

We used a local reconstruction, i.e., only a local area of the image was used to reconstruct each defective pixel. For each damaged pixel we found this region by creating a virtual window, with that pixel in it's centre. All valid pixels from this window are then used in creating a linear system (2). If the number of valid pixels is not sufficient (see [ZVS08]), the restoration of the actual pixel is adjourned for future iteration of the image reconstruction. See Section 5.3 for further information of performed experiments or [ZVS08] for detailed analysis and experiment results.

4. Proposed approach

In the previous work, disturbing color artefacts often occurred. These artefacts arise in the situations where large continuous areas were restored. In this work we focused on elimination of these defects.

Let us have a look at situation on Fig. 1a. We created a virtual window from the neighbourhood of the pixel, we would like to restore. This window contains enough valid pixels to create and solve linear system (2) but the problem is, that these valid pixels are spread inappropriately - all of them are on the left side or under processed pixel. This causes converging of values of the interpolant to extreme values and consequently origin of artefacts, we are trying to avoid.

4.1. Auxiliary points

The situation we would like to achieve is uniform distribution of valid pixels in the neighbourhood. Because no more information about the image is provided, we have to 'construct' and add some auxiliary pixels in these empty regions. Prior it, we need to find out which regions are empty. This is done by splitting the window into quadrants (Fig. 1b) and exploring them. Then for each empty quadrant we need to obtain the auxiliary pixel and insert it into that quadrant.

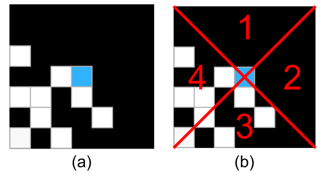


Figure 1: *a) Unbalanced distribution of valid pixels (white blocks) in the neighbourhood of the processed pixel (blue block). b) Splitting the neighbourhood into quadrants.*

Location of auxiliary point (pixel) is important - the closer it will be to the processed pixel, the more it will affect interpolated value. Since the auxiliary pixel can be constructed (i.e., it will not belong to real values of the image), we may increase an error in the image. Hence, this synthetic pixel shall be placed more distant then valid pixels in the window. Determining the proper position is impossible in practice so we reckoned on our experiments and stated empirically retrieved distance 5 pixels (for 7x7 virtual window) in direction of the axis of particular quadrant. Prior to computing and inserting auxiliary pixel we check it's coordinates in the empty quadrant. If there is valid pixel we will use it and no computation of auxiliary point will be performed. The reason is to modify the image as little as possible. If there is no valid pixel we have to estimate it. Following subsection deals with our approach how to estimate auxiliary points.

4.2. Triangulation based auxiliary pixel estimation

Involving only those pixels that occur in the immediate neighbourhood of the inserted one, will certainly obtain the best possible results. Our proposal is based on Delaunay triangulation and actually it does not create any synthetic pixel that will increase the error in the image.

Briefly, Delaunay triangulation is such type of triangulation where no point from the input set is present inside the circumcircle of any triangle of the triangulation.

The Delaunay triangulation is constructed from all valid pixels in the image prior to the image restoration. Then an empty quadrant is found and coordinates where the auxiliary pixel should be inserted are evaluated. Now, we localise the triangle including these coordinates and pick it's all 3 vertices, considering them as the auxiliary pixels. The reconstruction can consequently continue by creating linear system (from pixels found in the neighbourhood and auxiliary pixels) and solving it.

CGI 2009 proceedings, ACM ISBN 978-60558-687-8, pp.39-44, 2009

5. Experimental results

In this chapter we will discuss data we chose for experiments, then we will introduce the competitors for proposed method and finally compare results from experiments.

Majority of papers dealing with inpainting are considering only decent inpainting (overlayed text, scratches and cracks on the old photographs) where only a small regions of the image or images without large continuous damaged areas are required to restore [BS00], [OBMC01], [SKU02], [US05], [WWW06]. We focused on difficult cases but the experiments were performed on simpler ones too. We decided to compare with another algorithm from RBF family as well as with triangulation based methods since in these methods the interpolation is based on completely different principles.

5.1. Test images

The experiments were performed on wide range of input images and damage masks. We will present following representatives only, because of their character: Baboon for its high frequency content with high contrasts and Lena for containing large areas with mild transitions. Also following types of damage: rectangular grid (reconstruction is equal to scaling), noise (45% of the data is lost), extreme noise (95% damage), inpainted text and finally large continuous area of damage (see Fig. 2 respectively):

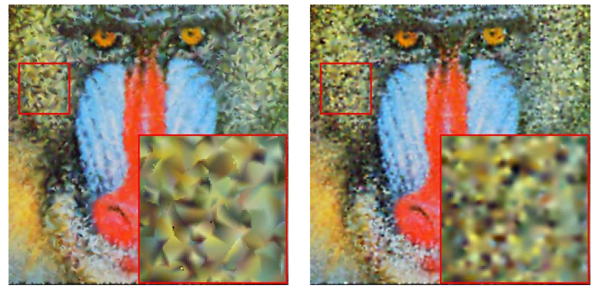
5.2. Competitors

We compared our novel technique with other RBF based algorithm proposed in [US05]. Novel method that utilise auxiliary points was verified on implementation of RBF algorithm introduced in [ZVS08]. Finally, we created Delaunay triangulation of valid pixels and interpolated pixels of each triangle both using bilinear interpolation and also Zienkiewicz's interpolation [Jan08].

5.3. Measurements

Experiments were performed on Pentium D 3.2 GHz, 4GB RAM, Windows Vista x64 in C#. For the evaluation we used established metrics PSNR and SSIM (see [WBSS04]), commonly used in image processing. Measurements were realised for each RGB component separately, presented value of each metric is average of these 3 values. RBF based measurements were achieved with following settings (see [ZVS08]): radius of neighbourhood equal to 3, LeftRight-TopBottom direction of the reconstruction, linear RBF and linear polynomial. All presented times are reconstruction times only, i.e., without triangulation (if needed), for images in 512×512 resolution. See Tab. 1 to get idea how much time took the creation of the triangulation for various masks.

In Fig. 3 we present results for Baboon image. Proposed method beats others in all variants of damage both in PSNR and SSIM comparison. The worst case was RBF algorithm with no auxiliary points and Noise95 mask. Fig. 4 shows this case, where disturbing artefacts are clearly visible. Novel RBF based method achieved better results also when comparing subjectively - bilinear and Zienkiewicz's interpolation created disturbing sharp edges of triangles (see Fig. 5). Disadvantage of RBF algorithm is computational time, where triangulation techniques performed much faster in most cases.



(a) (b)

Figure 5: *a) Zienkiewicz b) proposed RBF with auxiliary p.*

Another set of experiments was performed on Lena, where novel method achieved best results (see Fig. 6), also subjective perception is the best. Overall results are much smoother than triangulation based algorithms while computation times are comparable to previous experiments.

With Lena we utilised one more special mask - an adaptive triangulation (Fig. 7a). It was created to preserve the most important pixels (almost 2%), i.e., more triangles in detail areas whereas in large areas with mild transitions we need less triangles to save information in sufficient quality. This triangulation is than used as compressed representation of the image and consequently restored using some interpolation algorithm [DI04], [Jan08], [Koh07]. Reconstruction (decompression) of this adaptive triangulation best fits to the triangulation based approaches of course (input damaged image was intended for them), see results in Fig. 7. In RBF algorithms we utilise more pixels from the neighbourhood (i.e., we increase the error in the image in this specific case) - in contrast with triangulation techniques, where only vertices of appropriate triangles are utilised. But note, the idea of adaptive mask of damage is in conflict with the presumption that the input is not optimised for any method.

6. Conclusion and future work

We proposed a novel approach to RBF based image reconstruction. Principle of this technique is to insert auxiliary points into undefined areas. Computation of these points should be done while causing the error in the image as low as possible. We introduced approach how to obtain auxiliary points even without constructing synthetic pixels. We also

CGI 2009 proceedings, ACM ISBN 978-60558-687-8,

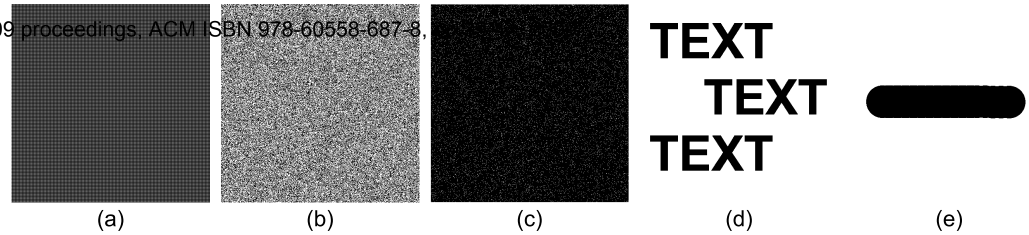


Figure 2: *Test masks of damage. a) Grid, b) Noise45, c) Noise95, d) Text and e) Wide.*

Table 1: *Triangulation times for our masks.*

	Adaptive	Noise95	Grid	Noise45	Text	Wide
pixels	3,975	11,789	65,536	142,043	223,506	229,224
time [ms]	93	250	3,656	13,360	26,921	28,735

performed experiments on different characters of damage and compared results with other algorithms that utilise different approach to interpolation. We were surprised by very good results we obtained, even in cases with extremely damaged inputs. Proposed method could be very helpful in these complicated cases.

Future research include restriction of triangulation on the selected areas only, GPU/CUDA acceleration and utilisation in image compression as suggested.

7. Acknowledgements

We thank to J. Skála for his triangulation library we used, J. Kohout and T. Janák for their application for interpolating of triangulation-represented digital images. We would like also to thank to all contributors to this publication anyway. Baboon and Lena images are courtesy of the Signal and Image Processing Institute at the University of Southern California.

This work was supported by the Ministry of Education of the Czech Republic, project VIRTUAL No.2C06002.

References

[BS00] BERTALMIO M., SAPIRO G.: Image inpainting. In *Proceedings of SIGGRAPH 2000* (New Orleans, USA, 2000), pp. 417–424.

[CBC*01] CARR J. C., BEATSON R. K., CHERRIE J. B., MITCHELL T. J., FRIGHT W. R., MCCALLUM B. C., EVANS T. R.: Reconstruction and representation of 3D objects with radial basis functions. In *SIGGRAPH 2001, Computer Graphics Proceedings* (2001), pp. 67–76.

[DI04] DEMARET L., ISKE A.: Advances in digital image compression by adaptive thinning. *Annals of the MCFA 3* (2004), 105–109.

[Jan08] JANÁK T.: On interpolation for triangulation-represented digital image. In *Proceedings of CESCG 2008* (Slovakia, 2008), pp. 59–66.

[Koh07] KOHOUT J.: On digital image representation by the delaunay triangulation. In *IEEE Pacific-Rim Symposium on Image and Video Technology (PSIVT 2007)* (2007), pp. 826–840.

[OBMC01] OLIVEIRA M. M., BOWEN B., MCKENNA R., CHANG Y.-S.: Fast digital image inpainting. In *Proceedings of the International Conference on Visualization, Imaging and Image Processing (VIIP 2001)* (Marbella, Spain, 2001), pp. 261–266.

[SKU02] SAVCHENKO V., KOJEKINE N., UNNO H.: A practical image retouching method. In *CW '02: Proceedings of the First International Symposium on Cyber Worlds* (Washington, DC, USA, 2002), pp. 480–487.

[US05] UHLÍŘ K., SKALA V.: Reconstruction of damaged images using radial basis functions. In *Proceedings of EUSIPCO 2005* (Bogazici University, 34342, Bebek, Istanbul, Turkey, 2005), p. 160.

[WBSS04] WANG Z., BOVIK A. C., SHEIKH H. R., SIMONCELLI E. P.: Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing 13*, 4 (April 2004), 600–612.

[WWW06] WEN P., WU X., WU C.: An interactive image inpainting method based on rbf networks. In *Advances in Neural Networks - ISNN 2006* (China, 2006), vol. 3972, pp. 629–637.

[ZVS08] ZAPLETAL J., VANĚČEK P., SKALA V.: Influence of essential parameters on the rbf based image reconstruction. In *Proceedings of SCCG 2008* (Slovakia, 2008), pp. 178–185.

CGI 2009 proceedings, ACM ISBN 978-60558-687-8, pp.39-44, 2009

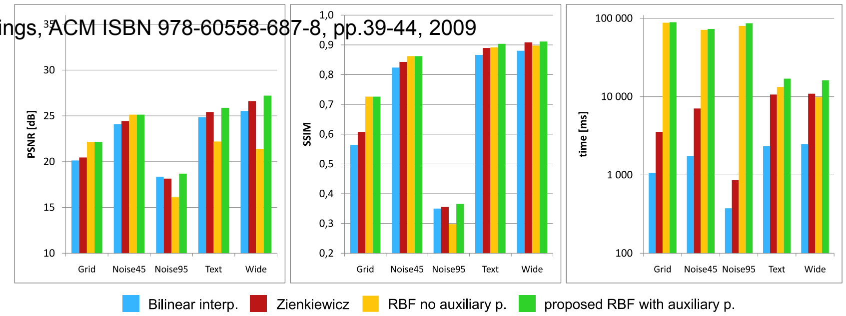
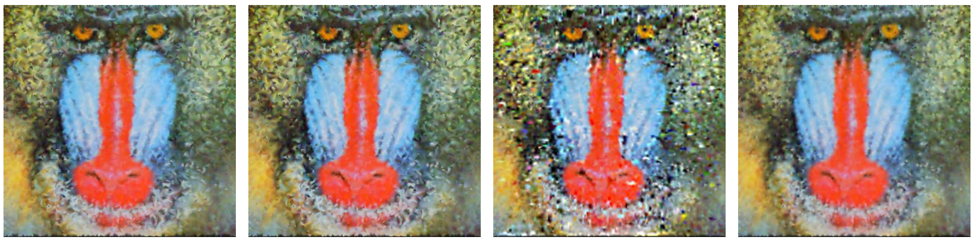


Figure 3: *Baboon. Left: PSNR graph, Middle: SSIM graph, Right: computing time.*



(a) (b) (c) (d)

Figure 4: *Results for Baboon image + Noise95 mask, a)Bilinear, b) Zienkiewicz c) RBF no auxiliary p. and d) proposed RBF with auxiliary p.*

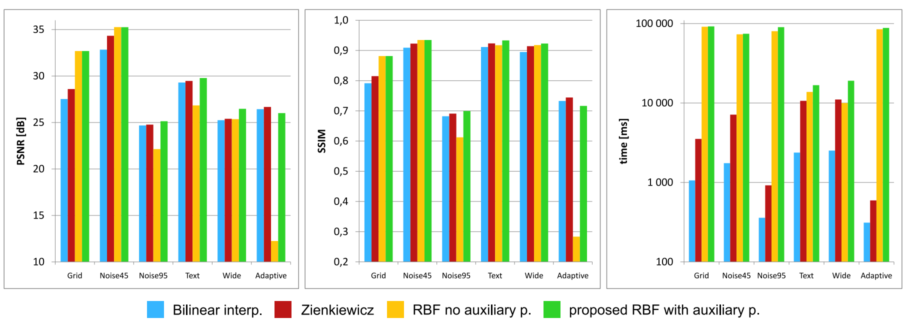
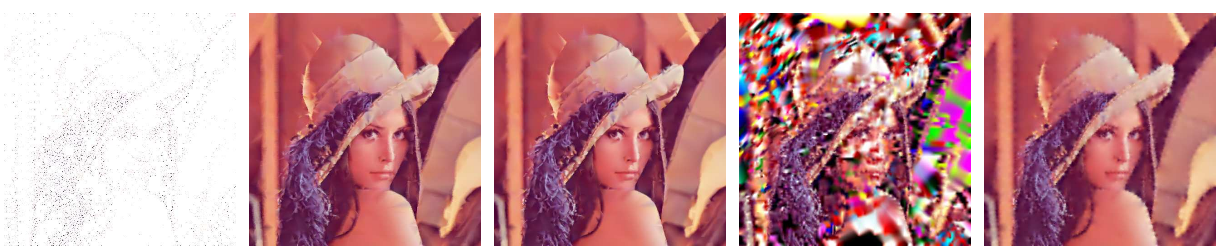


Figure 6: *Lena. Left: PSNR graph, Middle: SSIM graph, Right: computing time.*



(a) (b) (c) (d) (e)

Figure 7: *a) Image damaged by mask of adaptive triangulation (4000 vertices), b)Bilinear, c) Zienkiewicz d) RBF no auxiliary p. and e) proposed RBF with auxiliary p.*