

RECONSTRUCTION OF DAMAGED IMAGES USING RADIAL BASIS FUNCTIONS[‡]

KAREL UHLIR[§] AND VACLAV SKALA

Abstract. The Radial Basis Function method (RBF) can be used not only for reconstruction of surface from scattered data but for reconstruction of damaged images, filling gaps and for restoring missing data in images, too. The basic idea of reconstruction algorithm with RBF and very interesting results from reconstruction of images damaged by noise are presented. Feasibility of the RBF method for image processing is demonstrated.

Keywords. radial basis functions, interpolation, reconstruction, image processing

AMS subject classification. 15A30, 65D15, 41A10, 68U10

1. Introduction. One of the most interesting problems is how to reconstruct damaged or incomplete images as well as possible. This problem is referred to in many papers [6, 1]. The main question is: "What value was in a corrupted position and how can I restore it?". One of the conditions for solving this is to have as much information as possible from the original image. Then methods exist that use this and try to reconstruct information in gaps [1]. The amount of retained information from the original picture is very important and the quality of the result depends on it.

The Radial Basis Function method (RBF) is based on the variational implicit functions principle and can be used for interpolation of scattered data, see section 3 for details. The possibility of missing data restoration (image inpainting) by the RBF method was mentioned in Kojekine & Savchenko [7]. They used this method for surface retouching and marginally for image inpainting as well. They used compactly supported radial basis functions (CSRBF)[3] for reconstruction and octree data structure for representation of the parts for reconstruction. The advantage of this method is that the linear system is sparse and can be solved easily [9]. The drawback of this approach is in errors which can be obtained with an improper selection of the radius of support of the CSRBF functions see Fig. 11 and Table 2.

In our work we address a global basis function for image reconstruction and reconstruction with constant "window" size. A description of the method can be found below in section 4.

2. Problem definition. Let us assume that we have an image Ω with resolution $M \times N$ with 256 grey levels of pixel p .

$$(1) \quad \Omega = \Omega_k + \Omega_c, \quad \Omega_k \cap \Omega_c = 0,$$

$$(2) \quad p = [p_x, p_y], \quad p \in \Omega, \\ x = 0, \dots, M, \quad y = 0, \dots, N,$$

A pixel has value $h \in (0, 255)$ (intensity).

[‡] This project was supported by the Ministry of Education of the Czech Republic - Grant No.: MSM 235200005

[§] Department of Computer Science and Engineering, Faculty of Applied Science, University of West Bohemia in Pilsen, Czech Republic (kuhlir@kiv.zcu.cz)

Some pixels of the image Ω have incorrect values (missing or overwritten), see Fig. 1c. Thus the image Ω has two parts, one with correct pixel values Ω_k and the second with incorrect pixels values Ω_c , Eq. 1. We would like to restore the original image. Let us assume that we can detect “missing pixels” and pixels with corrupted values, too. We used “Lena” picture [13] to explain our approach (Fig. 1a) and the noise mask (Fig. 1b) was used to corrupt them.

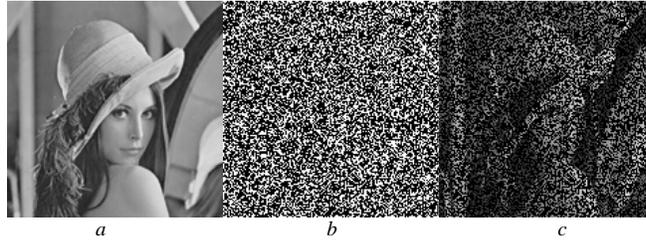


FIG. 1. Original image, mask with corrupted pixels and the final image prepared for reconstruction.

Note that restoration of the original image is related to scattered data interpolation problem, where many points are not defined and we want to find a value for them.

3. Radial Basis Function. Let us describe the RBF method now. The RBF method may be used to interpolate a smooth function given by n points. The resulting interpolating function thus becomes [5]:

$$(3) \quad f(\mathbf{x}) = \sum_{i=1}^n I_i f(\|\mathbf{x} - \mathbf{c}_i\|) + P(\mathbf{x}), \quad P(\mathbf{x}) = a_x x + a_y y + a_z$$

$$(4) \quad \sum_{i=1}^n I_i c_i^x = \sum_{i=1}^n I_i c_i^y = \sum_{i=1}^n I_i = 0,$$

where $\mathbf{c}_i = [c_i^x, c_i^y]^T$ are given locations of a set of n input points (pixels), I_j are unknown weights, \mathbf{x} is a particular point, $f(\|\mathbf{x} - \mathbf{c}_i\|)$ is a radial basis function, $\|\mathbf{x} - \mathbf{c}_i\| = r_i$ is the Euclidean distance (of pixels in our case), $f(\mathbf{c}_i) = h_i$, for $i=1, \dots, n$, and $P(\mathbf{x})$ is a polynomial of degree m depending on the choice of f . There are some popular choices for the basis function, e.g. the thin-plate spline $f(r) = r^2 \log(r)$, the Gaussian $f(r) = \exp(-\mathbf{x}r^2)$, the multiquadric $f(r) = \sqrt{r^2 + \xi^2}$, biharmonic $f(r) = |r|$ and triharmonic $f(r) = |r|^3$ splines, where ξ is a parameter.

Now we have a linear system of equations Eq. 3. with unknowns I_j, a_x, a_y, a_z . Natural additional constraints for the coefficients I_j must be included, Eq. 4., to ensure orthogonality of a solution. These equations and constraints determine the linear system:

$$(5) \quad \mathbf{B} \begin{bmatrix} ? \\ \mathbf{a} \end{bmatrix} = \begin{bmatrix} \mathbf{h} \\ \mathbf{0} \end{bmatrix}, \quad \text{where } \mathbf{B} = \begin{bmatrix} \mathbf{A} & | & \mathbf{P} \\ \mathbf{P}^T & | & \mathbf{0} \end{bmatrix} \quad \text{and} \quad \mathbf{P} = \begin{bmatrix} c_1^x & c_1^y & 1 \\ \vdots & \vdots & \vdots \\ c_n^x & c_n^y & 1 \end{bmatrix}$$

$$A_{i,j} = f(\|c_i - c_j\|), \quad i, j = 1, \dots, n$$

$$a = [a_x, a_y, a_z]^T, l = [l_1, l_2, \dots, l_n]^T, h = [h_1, h_2, \dots, h_n]^T$$

The polynomial $P(x)$ in Eq. 3. ensures positive-definiteness of the solution, of matrix B , see [8]. Afterwards, the linear equation system Eq. 5. is solved and the solution vector with l and a is known, the function $f(x)$ can be evaluated for an arbitrary point x (a pixel position in our case), see [8, 10, 4, 11].

4. Image reconstruction. We have the image Ω described in Eq. 1 and we would like to restore part Ω_c of the image Ω . The algorithm for reconstruction of corrupted pixels is based on the construction and the linear equation system solving for part of the image. The part of the image “window” is selected in “scan-line” algorithm. It is working with k -neighborhood of current pixel p . Note that in our case k is equal to 24, see Fig. 2.

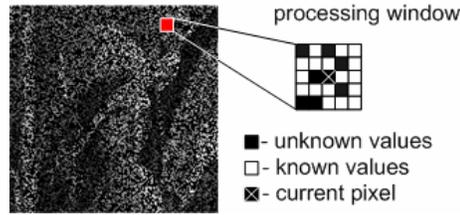


FIG. 2. Definition of the processing window for current pixel.

The correct pixels were selected from the window. These pixels were used as an input for the RBF method. The RBF function was computed and used iteratively to compute the “missing” pixel values. The RBF function differs from case to case as in the window several pixels might be missing. Now we can specify the proposed approach, see Algorithm 1.

```

LoadImage( $\Omega$ );
DefineNeighborhood(5,5);
Repeat
For (i,j=1;i<=M,j<=N;i++,j++)
{
    if(Hole(i,j)){ /* pixel [i,j] is not defined */
        K = SelectNeighborhoodOfPixel(i,j);
        DeleteHoles(K); /* remove all undefined pixels */
        CreateSystemAndSolve(K);
        Pixel[i,j] = ComputeValueFromSystem(i,j);
    }
}
Until (all pixels reconstructed)
    
```

ALGORITHM 1.

If there are too many undefined pixel values in the specified window of the size 5 x 5 pixels, then the undefined pixel is not restored and algorithm continues and the missing pixels are restored in the next iteration. It was the first modification of the algorithm mentioned above and the second was in detection of corrupted pixels in a way of the scan-line algorithm and the direction of movement changing, see Fig. 4 and Fig. 6.

5. Results. The LU factorization method was used for solving the linear system and $f(r) = r^2 \cdot \log(r)$ was used as the RBF function in our experiments. The LU factorization is very useful but different methods can be used, too.

5.1. Evaluation methods. For evaluation of the proposed method we used two methods. The first one was a following criterion:

$$(6) \quad S^k = \left(\sum_{i=1}^M \sum_{j=1}^N |(\Omega_1(i, j) - \Omega_2(i, j))|^k \right) \cdot \frac{1}{T}$$

where: Ω_1 is the original image (without corruption), Ω_2 is the reconstructed image, $k = 1$ (linear) or $k = 2$ (quadratic differences) and T could be the number of all image pixels or the number of corrupted pixels. Then S has two meanings. If the number of all image pixels is used as T value then S is error refers to the one image pixel. If T is used as the number of all corrupted pixels then S is error refers to the one corrupted image pixel. Full image was not used for the S evaluation because the image could be expanded error from outside of image where part of processing window is outside the image. The second method for comparison was visual. It is about the quadratic difference visualization (differential image), see Fig. 3b and Fig. 3c. The maximal simple error is $S = 255$ and quadratic error is $S^2 = 65\,025$ per pixel.

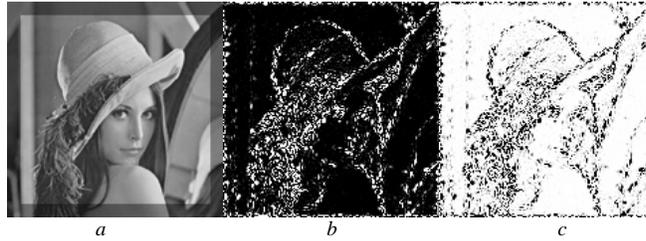


FIG. 3. The example of the selected area for S evaluation (a) and the example of black and white differential image (b,c).

5.2. Scan-line algorithm. We used several images that were corrupted by noise and the “Lena image”[13] and “Mars”[14] examples are presented here. For results of the one side scan-line algorithm reconstruction of the corrupted image, see Fig. 4 and Fig. 6.

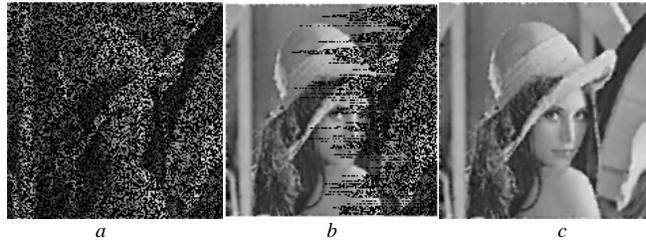


FIG. 4. The Lena image with 60% of corrupted pixels (a), intermediate step of the algorithm (b) and the result (c).

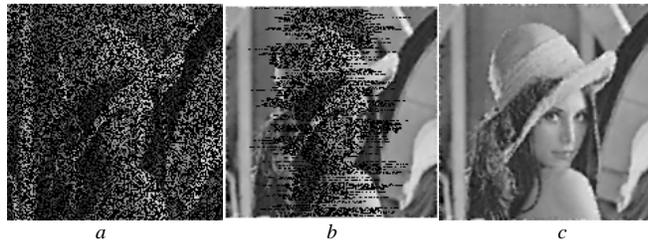


FIG. 5. The Lena image with 60% of corrupted pixels (a), intermediate step of the algorithm (b) and the result of two side scan-line algorithm(c).

For the evaluation of corrupted image with two side scan-line method is not necessary to have so many iterations. Results are better not only for images corrupted with the noise but also for images corrupted with any thickness scratches, see Fig. 5., Fig. 7. and Table 1.

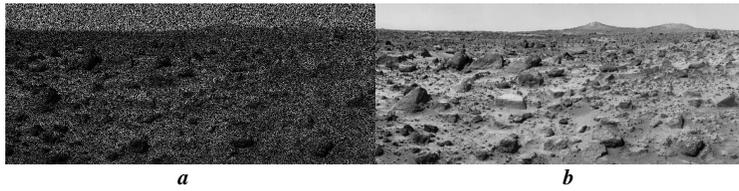


FIG. 6. The Mars image with 60% of corrupted pixels (a) and the result after reconstruction (b).

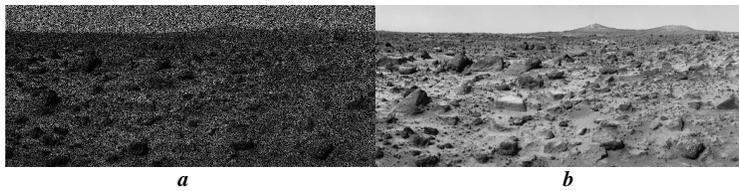


FIG. 7. The Mars image with 60% of corrupted pixels (a) and the result of two side scan-line algorithm (b).

The Table 1. presents results obtained for the case when only 60 % of pixels are left from the original image. In this table you can see the difference between one and the two side scan-line algorithm.

| Method | Image | all pixels | | corrupted pixels | |
|----------|-------|------------|----------------|------------------|----------------|
| | | S | S ² | S | S ² |
| One-side | Mars | 6.72 | 169.28 | 11.14 | 280.30 |
| | Lena | 4.28 | 92.63 | 6.99 | 151.43 |
| Two-side | Mars | 6.72 | 169.14 | 11.13 | 280.07 |
| | Lena | 4.26 | 91.15 | 6.96 | 149.00 |

TABLE 1. Results for 60% of corrupted pixels.

With increasing number of corrupted pixels the value of S and S^2 increase too. The problem is with the case of the highest percentage of corrupted pixels in the window. The window is small and it could happen that there are just few original pixels in the window. Therefore, the value in corrupted pixels is computed from ill-conditioned linear system. The graphs in Fig. 8. and Fig. 9. shows how S and S^2 errors increase for different percentage of corrupted pixels.

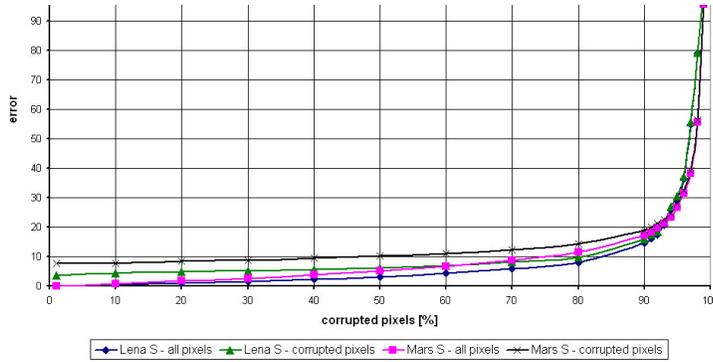


FIG. 8. The Lena and the Mars image S error comparison.

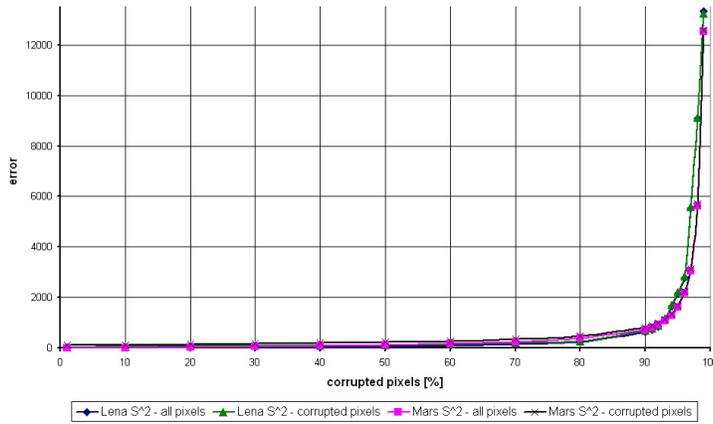


FIG. 9. Comparison of S^2 error on the Lena and the Mars image.

You can see that for more than 90% of corrupted pixels S and S^2 have an acceptable value and the result after the reconstruction is still good. The object at the image is still identifiable, see Fig. 10.

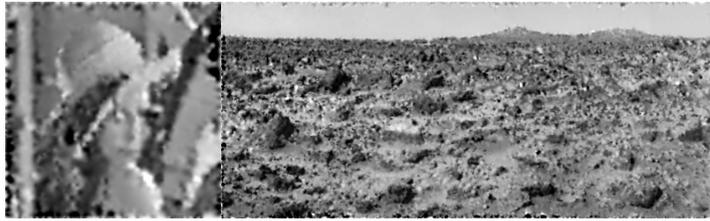


FIG. 10. The result of 90% corrupted images reconstruction.

5.3. Different basis functions and the window size. Finally we would like to compare different basis functions in order to check the quality of the reconstruction. Results for different basis functions are summarized in Table 2. and Fig. 11.

| | Functions | $r^2 \log r$ | r^3 | $(1-r)^4(4r+1)$ |
|------------------|----------------|--------------|--------|-----------------|
| all pixels | S | 3.19 | 3.21 | 4.11 |
| | S ² | 62.23 | 63.69 | 94.13 |
| corrupted pixels | S | 6.24 | 6.27 | 8.05 |
| | S ² | 121.72 | 124.56 | 184.10 |
| Figure | | 11a | 11b | 11c |

TABLE 2. Results obtained for different basis functions.



FIG. 11. Results from reconstruction with different basis function.

6. Conclusion and future work. It can be seen, that the results obtained by RBF application are very good. We also compared our results with the result of Bertalmio et. al [1], see Appendix A. His method was very good for the reconstruction of image inpainting. Our method could be used for this kind of reconstruction, too [12].

It can be seen that the proposed method has a good property, see the Fig. 12. where the problem is of our scan-line reconstruction method. The problem is in edges. On the sharp edges arise higher errors then in the image with slow changes. We would like to solve this problem in a future work. Next we would like to improve our method for obtaining better results with smaller S and S^2 error.

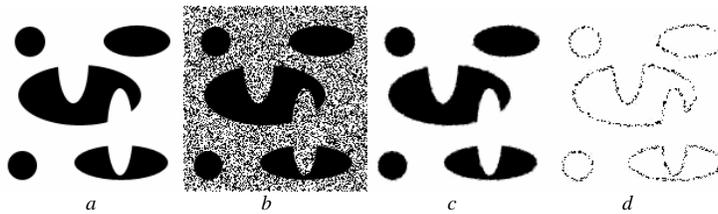


FIG. 12. Sharp edges reconstruction: a) the original image, b) the corrupted image, c) the reconstructed image and d) is the differential image of a) and c).

REFERENCES

- [1] M. BERTALMIO, G. SAPIRO, V. CASELLES AND C. BALLESTER, *Image Inpainting*, in Proceedings of the ACM SIGGRAPH Conference on Computer Graphics, SIGGRAPH 2000, pp.417-424, 2000.
- [2] J. BLOOMENTHAL, *Polygonization of Implicit Surface*, Computer-Aided Geometric Design, vol. 5, no. 4, pp. 341-355, 1988.
- [3] J. BLOOMENTHAL, C. BAJAJ, J. BLINN, M. P. CANI-GASCUEL, A. ROCKWOOD, B. WYVILL, G. WYVILL, *Introduction to Implicit Surfaces*, Morgan Kaufmann, 1997.
- [4] J. C. CARR, R. K. BEATSON, J. B. CHERRIE, T. J. MITCHELL, W. R. FRIGHT, B. C. MCCALLUM, T. R. EVANS, *Reconstruction and Representation of 3D Objects with Radial Basis Functions*, Computer Graphics (SIGGRAPH 2001 proceedings), pp. 67-76, August 2001.
- [5] J. DUCHON, *Splines minimizing rotation-invariant semi-norms in Sobolev space*, In W. Schempp and K. Zeller, editors, Constructive Theory of Functions of Several Variables, No. 571 in Lecture Notes in Mathematics, Springer-Verlag, pp. 85-100, Berlin, 1977.
- [6] N. KOZHEKIN, N., V. SAVCHENKO, M. SENIN, I. HAGIWARA, *An Approach to Surface Retouching and Mesh Smoothing*, International journal "The Visual Computer", Volume 19, Number 7-8, pp. 549-564, 2003.
- [7] N. KOJEKINE, V. SAVCHENKO, *Using CSRBFs for Surface Retouching*, Proceedings of The 2nd IASTED International Conference Visualization, Imaging and Image Processing VIIP2002, Malaga, Spain, 2002.
- [8] B. MORSE, T. S. YOO, P. RHEINGANS, D. T. CHEN, K. R. SUBRAMANIAN, *Interpolating Implicit Surfaces from Scattered Surface Data Using Compactly Supported Radial Basis Functions*, in Proceedings of the Shape Modeling conference, Genova, Italy, pp. 89-98, May 2001.
- [9] I. TOBOR, P. REUTER, C. SCHLICK, *Multiresolution Reconstruction of Implicit Surfaces with Attributes from Large Unorganized Point Sets*, SMI 2004, Genova, Italy, 06/2004.
- [10] G. TURK, J. F. O'BRIEN, *Modelling with Implicit Surfaces that Interpolate*, ACM Transactions on Graphics, Vol. 21, No. 4, pp. 855-873, October 2002.
- [11] K. UHLIR, *Modeling methods with implicitly defined objects*, State of the Art and Concept of Doctoral Thesis, University of West Bohemia, Czech Republic, Technical Report No. DCSE/TR-2003-04, 2003.
- [12] K. UHLIR, V. SKALA, *Radial Basis Function use for the Restoration of Damaged Images*, ICCVG'2004, Poland, 2004.
- [13] <http://sipi.usc.edu/services/database/Database.html>
- [14] <http://mars3.jpl.nasa.gov/MPF/>

APPENDIX A

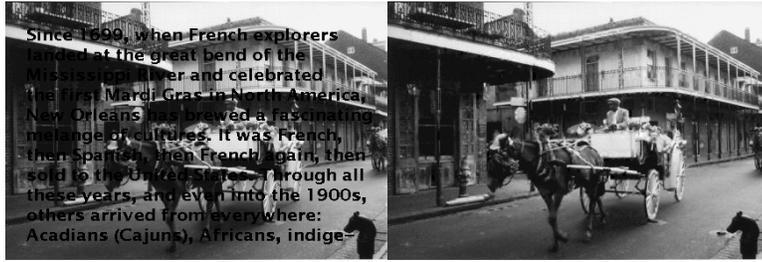


FIG. 13. Image inpainted and its reconstruction by Bertalmio method, taken from [1].



FIG. 14. The image reconstructed by our method.

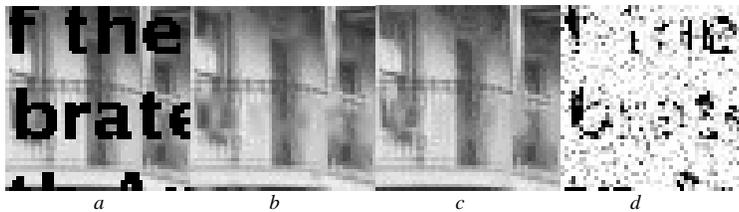


FIG. 15. Selected parts of images: a) the original image, b) the image reconstructed by Bertalmio, c) the image reconstructed by our method and d) the differential image of b) and c).