15th International Symposium on Computer Methods in Biomechanics and Biomedical Engineering
and
3rd Conference on Imaging and Visualization
CMBBE 2018
P. R. Fernandes and J. M. Tavares (Editors)

3D LANCZOS INTERPOLATION FOR MEDICAL VOLUMES

Thiago Moraes[1], Paulo Amorim[1], Jorge Silva[1] and Helio Pedrini[2]

[1]Division of 3D Technologies, Center for Information Technology Renato Archer
Campinas-SP, Brazil, 13069-901

[2]Institute of Computing, University of Campinas
Campinas, SP, Brazil, 13083-852

Keywords: Image Interpolation, Resampling, Medical Imaging.

Abstract: *Medical imaging techniques have been widely used in the diagnosis and treatment of many diseases, such as magnetic resonance imaging, computed tomography, mammography, ultrasonography, and photon emission tomography. Their main purpose is to enable the visualization of internal anatomic structures, such as organs and tissues, for clinical procedures. Resampling is required for certain medical tasks, for instance, registration and reconstruction, which occur when images or volumes are scaled, translated or rotated. The result of the resampling depends on the interpolation filter. In this paper, we develop and analyze a novel three-dimensional Lanczos resampling method in the context of medical imaging.*

1. INTRODUCTION

In the context of numerical analysis, interpolation techniques [12–15, 19, 21–23] aim to construct new data points from a set of known samples. In the field of image analysis, resampling techniques [9, 21, 23] transform a sampled image from one coordinate system to another.

Resampling is typically employed to create a new version of an image with a different dimension. The increase and reduction of an image are called upsampling and downsampling, respectively.

Image interpolation and resampling techniques play an important role in several biomedical applications, such as data reslicing, image registration, tomographic reconstruction, and volumetric rendering [23]. In these tasks, magnification or rotation of a region of interest in an image is commonly used to provide more appropriate biomedical data representations for analysis and visualization purposes [3–5, 20].

Several methods for image interpolation have been proposed in the literature [8, 11, 27], for instance, nearest neighbor interpolation, bicubic interpolation, bilinear interpolation and spline interpolation, which demand different computational costs and provide distinct levels of quality. In this work, we introduce and analyze a novel three-dimensional Lanczos resampling method in the biomedical context.

This text is organized as follows. Some relevant concepts and approaches related to the topic under investigation are briefly presented in Section 2. The proposed resampling method is described in Section 3. Experimental results are presented and discussed in Section 4. Some final remarks and directions for future work are outlined in Section 5.

2. BACKGROUND

Interpolation of sampled data is a common task required in several medical image processing operations, for instance, magnification, rotation, warping, registration, among others.

Various techniques for image interpolation have been developed over the last decades [8, 11, 21, 23, 27]. These techniques play an important role in the development of analysis and visualization algorithms for two- and three-dimensional medical data [7, 17, 24].

In order to generate a final image, the interpolation methods may employ a different number of sampled points. Level of quality and computational time are two important factors that affect the results when distinct interpolation methods are used.

Nearest neighbor interpolation is a multivariate method for approximating the value of a function for an unknown point as the value of its nearest point. Although it is a fast method, the nearest neighbor interpolation may result in jagged edges.

Trilinear interpolation is a multivariate interpolation technique that operates on a three-dimensional regular grid, which approximates the value of an intermediate data point (x, y, z). The method employs the eight nearest points on the grid along x, y and z directions to linearly approximate the value of the data point.

Tricubic interpolation [16] obtains values at arbitrary points on a three-dimensional regular grid. The method requires the approximation of a function expressed as

$$f(x,y,z) = \sum_{i=0}^{3}\sum_{j=0}^{3}\sum_{k=0}^{3} a_{ijk} x^i y^j z^k \tag{1}$$

where a_{ijk} are coefficients that must be determined based on the data. The tricubic interpolation preserves fine detail in the output image, however, it demands more processing time.

3. METHOD

Lanczos resampling [6, 10, 18] is a 1-D interpolation method based on the Lanczos kernel, which is dynamic and must be calculated for each value to be interpolated, expressed as

$$S(x) = \sum_{i=\lfloor x \rfloor - a + 1}^{\lfloor x \rfloor + a} s_i L(x - i) \tag{2}$$

where x is a real value to be interpolated, a is the filter size, and $L(x)$ is

$$L(x) = \begin{cases} \operatorname{sinc}(x)\ \operatorname{sinc}(x/a) & \text{if } -a < x < a, \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Lanczos resampling is a separable filter, which means that it is possible to first apply it in the horizontal direction, then in the vertical direction and, finally, in the depth direction to be used in the 3D interpolation. Since the kernel may have negative values, the range of values of the output image may be wider than the input image. Therefore, the output image must be

rescaled after the interpolation process in order to obtain the same value interval as the input values.

Algorithm 1 presents the main steps of the proposed interpolation method. The algorithm takes as input a volumetric image, the position to be interpolated, and the value of a, which indicates the size of the filter (equal to $2a-1$). Two arrays are created (`tempx` and `tempy`) that temporarily store the interpolation performed in the x-direction, then in the y-direction.

Between lines 10 and 20, the interpolation in the x-direction is performed. Lines between 21 and 29 perform interpolation in the y-direction using previously calculated interpolation values (`tempx`). Finally, it is performed in the z-direction using the values of `tempx` (interpolation values on the y-axis). Finally, the interpolation value is returned.

This process is illustrated in Figure 1. The objective is to calculate the interpolation for the point at position $(1.5, 1.5, 1.5)$. Initially, the plane where $X = 1.5$ is computed, that is, the interpolation in the X direction. Such points are shown in red color in the figure. It is possible to observe that only the X values are equal to 1.5, whereas coordinates Y and Z are integers $(0, 1, 2, 3, 4, 5)$. Then, the interpolation in Y direction is calculated. In this case, points at $X = 1.5$ and $Y = 1.5$ are computed, which corresponds to a line that intercepts the plane $X = 1.5$. Such points are shown in green color in the figure. Finally, the value at position $(1.5, 1.5, 1.5)$ is computed, which is the blue point that intercepts the line $X = 1.5$ and $Y = 1.5$ at position $Z = 1.5$.

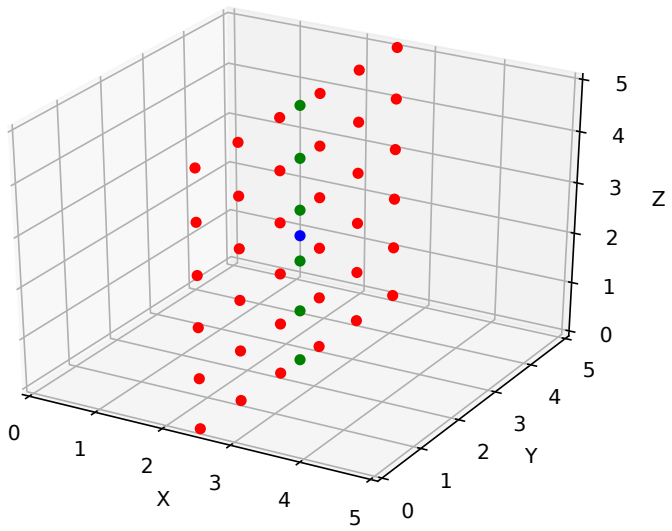


Figure 1. Illustration of the three main steps for calculating the 3D Lanczos interpolation.

In our experiments, we observed that $a = 2$ (the filter size) can generate images with ringing artifacts, which did not occur with $a = 3$ or $a = 4$. We used several data sets to demonstrate the effectiveness of the proposed approach. Results were compared with nearest neighbor, trilinear and tricubic interpolation methods.

Algorithm 1 Lanczos Interpolation

INPUT: Image (volumetric image)
INPUT: x, y, z (position to interpolate)
INPUT: a (Lanczos a)
OUTPUT: The interpolated value

```
xi ← ⌊x⌋ − a + 1
xf ← ⌊x⌋ + a
yi ← ⌊y⌋ − a + 1
yf ← ⌊y⌋ + a
zi ← ⌊z⌋ − a + 1
zf ← ⌊z⌋ + a

size ← a ∗ 2 − 1
tempx[size][size]
tempy[size]
m = 0
for k = zi to zf do
    n = 0
    for j = yi to yf do
        for i = 0 to xf do
            tempx[m][n] += image[i, j, k] * L(x - i, a)
        end for
        n++
    end for
    m++
end for
m = 0
for k = zi to zf do
    n = 0
    for j = yi to yf do
        tempy[m] =tempx[m][n] * L(y - j, a)
        n ++
    end for
    m++
end for
m = 0
for k = zi to zf do
    value += tempy[m] * L(z - k, a)
    m++
end for
return value
```

4. EXPERIMENTAL RESULTS

Experiments were performed comparing nearest neighbor, trilinear, tricubic and Lanczos (with $a = 3$ and $a = 4$) interpolation methods. In these experiments, a geometric scaling transformation was used to reduce the volumetric medical images by half (scale of 0.5) and then another scale to duplicate the image (scale of 2.0). The same interpolation method was used in the two scaling transformations.

The result of these two transformations was then compared to the original volumetric image using the Structural Similarity (SSIM) metric [25]. The SSIM metric varies in the range $[-1, 1]$, where the value 1 indicates that there is no difference between the two data sets. The experiments were conducted on the Goudurix and Manix images [2] and Chest, Clavicle, Cranium, Feet, Fox and MRI T1 [1]. The results of these experiments are available in Table 1. The best SSIM values are highlighted in bold in Table 1.

Table 1. Comparison of SSIM results for nearest neighbor, trilinear, tricubic and Lanczos ($a = 3$ and $a = 4$) interpolation methods by initially scaling a volumetric by half and then by two compared to the original one.

Model	**Nearest Neighbor**	**Trilinear**	**Tricubic**	**Lanczos**	
				$a = 3$	$a = 4$
Goudurix	0.9984416	0.9998742	**0.9999394**	0.9999169	0.9999357
Manix	0.9987755	0.9998514	0.9999105	0.9998855	**0.9999117**
Chest	0.9988911	0.9998576	0.9999103	0.9998914	**0.9999133**
Clavicle	0.9987620	0.9997331	0.9998004	0.9997795	**0.9998067**
Cranium	0.9966246	0.9994788	0.9996321	0.9996222	**0.9996422**
Feet	0.9985482	0.9997178	0.9998197	0.9998071	**0.9998499**
Fox	0.9986468	0.9998103	**0.9998872**	0.9998497	0.9998575
MRI T1	0.9961121	0.9986351	**0.9987272**	0.9986532	0.9985971

From Table 1, it is possible to observe that the Lanczos method with $a = 4$ obtained superior results for Manix, Chest, Clavicle, Cranium and Feet data sets. In the other cases, the tricubic method performed better. It is also possible to verify that, in none of the cases, the Lanczos method with $a = 3$ achieved better results than the tricubic method. However, the Lanczos $a = 3$ method obtained better results in all cases when compared to the trilinear and nearest neighbor methods.

Figure 2 illustrates the original image (corresponding to slice 301) and the results after applying the five tested interpolation methods. The result for the nearest neighbor interpolation presents certain loss of details compared to the original image, as well as aliasing problems. In the trilinear interpolation, the effect of image smoothing is noticeable. In region 1 (labeled in the original image), the image smoothing made the bone protrusion narrower and closed the hole. In region 2, the hole is larger than in the original image. In region 3, the higher degree of smoothing is noticeable. The effect of smoothing in the tricubic interpolation method is less

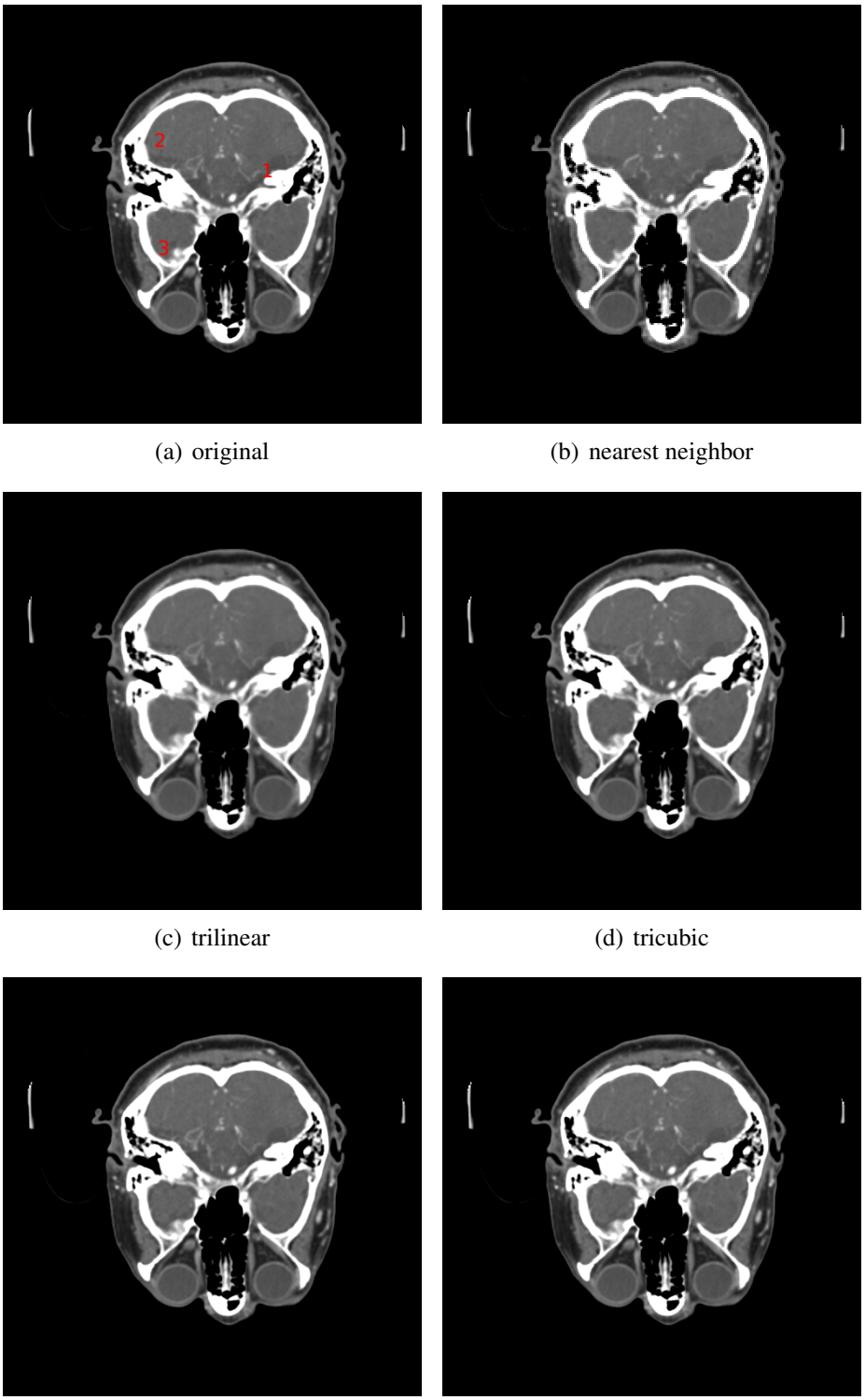
noticeable. In region 1, the tricubic method did not narrow the protrusion between the bones, however, was able to close the hole. In region 2, the hole increased a little bit. The image smoothing affected region 3. The Lanczos methods ($a = 3$ and $a = 4$) did not narrow the gap between the bones in region 1 and preserved part of the hole. In region 2, the hole increased a little bit, similarly to the other methods. In region 3, the Lanczos methods also smoothed the area, however, the edges are slightly more preserved than in the other methods.

Experiments were conducted using frames of cell division time lapse videos extracted from [26]. For the experiment, the frames were reduced in $\frac{1}{4}$ by taking one every four frames. Then, the intermediate frames were regenerated using a geometric scaling transformation in the temporal direction, that is, increasing the number of frames, not the resolution of the frames. In addition to the geometric transformation, several interpolation algorithms were applied: trilinear, tricubic and Lanczos (with $a = 3$ and $a = 4$). Then, the interpolated frames were compared with the original frames. Table 2 summarizes the results, where the comparison metric is the Mean Square Error (MSE). The lower the MSE value, the closer the images are. The best MSE values are highlighted in bold in Table 2.

Table 2. Comparison of MSE results for trilinear, tricubic and Lanczos ($a = 3$ and $a = 4$) interpolation methods.

Model	**Nearest Neighbor**	**Trilinear**	**Tricubic**	**Lanczos**	
				$a = 3$	$a = 4$
DKWH7_L10	2512.44	2132.93	1847.33	1736.53	**1726.70**
DKWH7_L1	4314.23	3609.43	3311.63	**3214.00**	3216.03
DKWH7_L15	2532.01	2215.70	1939.61	**1864.37**	1870.41
DKWH7_L19	1623.58	**1310.97**	1332.40	1613.68	1770.36
DKWH7_L6	4688.67	3976.92	3547.25	3279.69	**3198.15**
DKWH7_L4	6260.87	5144.85	4690.35	4406.36	**4310.74**
DKWH7_L3	2603.42	2197.08	1919.24	1803.28	**1789.05**

As observed in Table 2, the Lanczos interpolation method achieved superior overall results, that is, it generated intermediate frames closer to the original ones. The trilinear method produced the best results only for the DKWH7_L19 model. In this model, the Lanczos methods were surpassed by the tricubic method, whereas the Lanczos method with $a = 4$ was surpassed by the nearest neighbor interpolation. In all other cases, the Lanczos methods produced better results. Unlike the results shown in Table 1, Lanczos interpolation with $a = 3$ obtained better results in some models when compared to Lanczos with $a = 4$. Since the difference between the two Lanczos interpolation methods is small, it may be more advantageous to use Lanczos with $a = 3$ because it is faster than Lanczos with $a = 4$.



(a) original (b) nearest neighbor

(c) trilinear (d) tricubic

(e) Lanczos $a = 3$ (f) Lanczos $a = 4$

Figure 2. Results for Manix data set obtained with different interpolation methods.

5. CONCLUSIONS AND FUTURE WORK

We introduced and analyzed a three-dimensional Lanczos resampling method for medical images. Experiments conducted on volumetric data sets demonstrated the effectiveness of the proposed method compared to nearest neighbor, trilinear, tricubic interpolation techniques.

The assessment of the results was performed through Structural Similarity (SSIM) and Mean Square Error (MSE) metrics. The results demonstrated the potential use of our method for the interpolation of medical images and the generation of intermediate frames in cell division time lapse videos.

As directions for future work, we intend to test larger size values for the filter in the Lanczos method, extend other three-dimensional interpolation techniques to four dimensions, as well as apply the methods to several other medical volumetric data sets.

ACKNOWLEDGEMENTS

The authors are thankful to FAPESP (grants #2014/12236-1 and #2017/12646-3) and CNPq (grant #305169/2015-7) for their financial support.

References

[1] InVesalius | DICOM Dataset, Accessed: 2018-02-09. `https://www.cti.gov.br/en/invesalius#download`.

[2] OsiriX | DICOM Image Library, Accessed: 2018-02-09. `http://www.osirix-viewer.com/resources/dicom-image-library/`.

[3] P. Amorim, T. Moraes, J. Silva, and H. Pedrini. An Out-of-Core Volume Rendering Architecture. In *IV ECCOMAS Thematic Conference on Computational Vision and Medical Image Processing*, Funchal, Portugal, Oct. 2013.

[4] P. Amorim, T. Moraes, J. Silva, and H. Pedrini. InVesalius: An Interactive Rendering Framework for Health Care Support. In *International Symposium on Visual Computing*, pages 45–54. Springer, 2015.

[5] P. Amorim, T. M. R. Rezende, J. Silva, and H. Pedrini. Medical Imaging for Three-Dimensional Computer-Aided Models. *3D Printing and Biofabrication*, pages 1–27, 2017.

[6] A. Bentbib, M. El Guide, K. Jbilou, and L. Reichel. A Global Lanczos Method for Image Restoration. *Journal of Computational and Applied Mathematics*, 300:233–244, 2016.

[7] P. S. Calhoun, B. S. Kuszyk, D. G. Heath, J. C. Carley, and E. K. Fishman. Three-Dimensional Volume Rendering of Spiral CT Data: Theory and Method. *Radiographics*, 19(3):745–764, 1999.

[8] W. K. Carey, D. B. Chuang, and S. S. Hemami. Regularity-preserving Image Interpolation. *IEEE Transactions on Image Processing*, 8(9):1293–1297, 1999.

[9] N. A. Dodgson. Quadratic Interpolation for Image Resampling. *IEEE Transactions on Image Processing*, 6(9):1322–1326, 1997.

[10] S. Fadnavis. Image Interpolation Techniques in Digital Image Processing: An Overview. *International Journal of Engineering Research and Applications*, 4(10):70–73, 2014.

[11] J.-W. Han, J.-H. Kim, S.-H. Cheon, J.-O. Kim, and S.-J. Ko. A Novel Image Interpolation Method using the Bilateral Filter. *IEEE Transactions on Consumer Electronics*, 56(1), 2010.

[12] H. Hou and H. Andrews. Cubic Splines for Image Interpolation and Digital Filtering. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(6):508–517, 1978.

[13] J.-J. Huang, W.-C. Siu, and T.-R. Liu. Fast Image Interpolation via Random Forests. *IEEE Transactions on Image Processing*, 24(10):3232–3245, 2015.

[14] H. Jiang and C. Moloney. A New Direction Adaptive Scheme for Image Interpolation. In *International Conference on Image Processing*, volume 3, pages III–III. IEEE, 2002.

[15] T. M. Lehmann, C. Gonner, and K. Spitzer. Survey: Interpolation Methods in Medical Image Processing. *IEEE Transactions on Medical Imaging*, 18(11):1049–1075, 1999.

[16] F. Lekien and J. Marsden. Tricubic interpolation in three dimensions. *Journal of Numerical Methods and Engineering*, 63:455–471, 2005.

[17] M. Levoy. Display of Surfaces from Volume Data. *IEEE Computer Graphics and Applications*, 8(3):29–37, 1988.

[18] B. Madhukar and R. Narendra. Lanczos Resampling for the Digital Processing of Remotely Sensed Images. In *International Conference on VLSI, Communication, Advanced Devices, Signals & Systems and Networking*, pages 403–411. Springer, 2013.

[19] A. Malik, G. Sikka, and H. K. Verma. An Image Interpolation based Reversible Data Hiding Scheme using Pixel Value Adjusting Feature. *Multimedia Tools and Applications*, 76(11):13025–13046, 2017.

[20] T. F. Moraes, P. Amorim, J. da Silva, H. Pedrini, and M. I. Meurer. Medical Volume Rendering based on Gradient Information. In *V ECCOMAS Thematic Conference on Computational Vision and Medical Image Processing*, volume 5, pages 181–186, Tenerife, Canary Islands, Spain, Oct. 2015.

[21] J. A. Parker, R. V. Kenyon, and D. E. Troxel. Comparison of Interpolating Methods for Image Resampling. *IEEE Transactions on Medical Imaging*, 2(1):31–39, 1983.

[22] N. Plaziac. Image Interpolation using Neural Networks. *IEEE Transactions on Image Processing*, 8(11):1647–1651, 1999.

[23] P. Thévenaz, T. Blu, and M. Unser. Image Interpolation and Resampling. *Handbook of Medical Imaging, Processing and Analysis*, 1(1):393–420, 2000.

[24] U. Tiede, K. H. Höhne, M. Bomans, A. Pommert, M. Riemer, and G. Wiebecke. Investigation of Medical 3D-Rendering Algorithms. *IEEE Computer Graphics and Applications*, 10(2):41–53, 1990.

[25] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[26] A. Zaritsky, S. Natan, D. Kaplan, E. Ben-Jacob, and I. Tsarfaty. Supplemental Materials - Live Time-Lapse Dataset of in Vitro Wound Healing Experiments, 2015. `http://gigadb.org/dataset/100118`.

[27] D. Zhou, X. Shen, and W. Dong. Image Zooming using Directional Cubic Convolution Interpolation. *IET Image Processing*, 6(6):627–634, 2012.