

Bio-inspired, Invariant Image Reconstruction from Irregularly Placed Samples

Flore Faille and Maria Petrou
Imperial College London, London, SW7 2AZ, UK

Abstract

We present in this paper an algorithm inspired by the human visual system to reconstruct images from irregularly placed samples. This algorithm is based on linear spline approximations with control points on a hexagonal grid. Several spline approximations are computed for different transformations of the control point grid (e.g. translations and rotations). These approximations are then merged together after compensation of the transformations, yielding a high-quality invariant image reconstruction. Evaluations show that the use of hexagonal grids and of the “invariance by integration” principle improves reconstruction quality.

1 Introduction

Non uniform sampling is a problem occurring in some imaging fields like e.g. geophysics, medical imaging or motion estimation. An overview of existing methods to handle this problem is given in [7]. We present in this paper a new algorithm inspired by the human visual system which reconstructs images from few irregularly placed pixels, based on spline interpolation. It gives a continuous scene representation, which can be used to compute images of any resolution at any position or rotation. The obtained representation is invariant to chosen image transformations, like rotations, translations, scalings, etc.

In the human visual system, images are sampled by irregularly placed sensors, with a higher density of photoreceptors in the fovea than in the eye periphery. Saccades are executed when viewing a scene. Between saccades, the eyes perform fixational eye movements [4]. Because of these eye movements and of the varying photoreceptor density, our eyes measure light intensity at irregular positions in the scene. Yet, we perceive a continuous, aliasing-free image. Thus, the human visual system performs image reconstruction from irregular samples. People are also able to recognise patterns independently of their position and orientation, implying that this reconstruction is invariant to image transformations. Our work uses principles of the human visual system for image reconstruction.

In the primary visual cortex, complex cells achieve translation invariant feature detection by combining the responses of several simple cells which detect the same feature, each at different positions in the receptive field [2]. Invariance is hence obtained by integrating simple responses over several transformations (in this case translations). The role of fixational eye movements is largely unknown [4]. One hypothesis based on this “invariance by integration principle” was proposed in [3]: microsaccades implement invariant image reconstruction by integrating simple image reconstructions over a large number of

sensor positions. For each position, a representation of the scene is reconstructed. All representations are merged together after motion compensation, yielding a high quality, invariant image. This “invariance by integration” principle was emulated in [3]: several linear spline approximations were estimated from irregularly placed samples for different grids of control points. These were merged after compensating the transformations between grids, producing a higher quality image than each single spline approximation.

Here, we develop further this algorithm by introducing hexagonal grids, which mimic the cell organisation in the fovea. For this, we use linear splines with control points placed on a hexagonal grid. In addition to copying nature, hexagonal grids have several advantages over square grids. More points can be fitted in 2D areas. Each grid point has 6 equidistant neighbours instead of 4 for square grids. Hexagonal grids have more symmetries than square grids. Further advantages of hexagonal grids can be found in [5].

We present in this paper a bio-inspired algorithm for invariant image reconstruction from irregular samples. The novelty of this algorithm is the use of linear B-splines with control points on hexagonal grids. Invariant image reconstruction is obtained by integrating over transformations of the control point grids (translations, rotations, etc.) as in [3]. We compare the performance of this new algorithm with the algorithm in [3] which uses square grids and with normalised convolution [1] which is a state of the art algorithm for reconstruction from irregularly placed samples. These comparisons allow us to measure the benefits of hexagonal grids and of the bio-inspired invariance framework, respectively. After an introduction to hexagonal grids in section 2, we present splines on hexagonal grids in section 3. The reconstruction algorithm is then explained in section 4. Last, the reconstruction results and conclusions are given in sections 5 and 6, respectively.

2 Hexagonal grids

We use the hexagonal grid shown in fig. 1 (a). When the distance between two grid points is d , the distance between two rows is $\frac{\sqrt{3}}{2}d \simeq 0.866d$. Hence, roughly 15% more grid points can be fitted in a rectangular area than with a square grid. To describe a point on a hexagonal grid, the oblique coordinate system (\mathbf{u}, \mathbf{v}) will be used instead of the usual (\mathbf{x}, \mathbf{y}) orthogonal coordinate system, because it represents all three grid directions more easily (see fig. 1 (b)). The coordinates of point $\mathbf{P} = u\mathbf{u} + v\mathbf{v} = x\mathbf{x} + y\mathbf{y}$ in the two coordinate systems are related by:

$$\begin{cases} u = x + \frac{1}{\sqrt{3}}y \\ v = \frac{2}{\sqrt{3}}y \end{cases} \quad \text{and} \quad \begin{cases} x = u - \frac{1}{2}v \\ y = \frac{\sqrt{3}}{2}v \end{cases} \quad (1)$$

To store data sampled on hexagonal grids, the simple framework of [6] will be used: the hexagonal grid is approximated by a brickwall, so the data can be stored as usual in a 2D array (see fig. 1 (c)). It is an efficient storage method but it must be kept in mind that every second row is translated by $d/2$, which makes processing more difficult.

3 Splines on hexagonal grids

Splines are often used to reconstruct continuous image signals, e.g. for interpolation or resampling. The B-spline framework was extended to hexagonal grids in [9, 8]: the

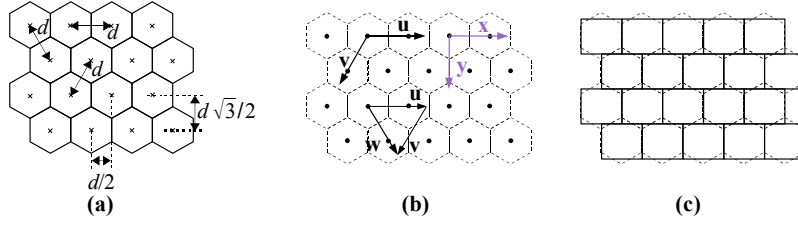


Figure 1: Hexagonal grid: (a) properties, (b) oblique coordinate system and (c) brickwall approximation for storage.

B-spline of degree 0 is the indicator function of the Voronoi cell of the hexagonal grid (i.e. the hexagonal tile). The B-spline of degree n is built by successive convolutions:

$$\beta^n(u, v) = \frac{(\beta^{n-1} * \beta^0)(u, v)}{\Omega} \quad (n \geq 1) \quad (2)$$

where Ω is the surface area of the Voronoi cell. $\Omega = \frac{\sqrt{3}}{2}d^2$, where d is the distance between two grid points. We shall base our algorithm on this spline family.

Here, we fit a linear spline to irregularly placed samples. This requires the analytic form of the linear B-spline, which was not given in [9, 8]. We derived it with eq. (2) and $n = 1$. This leads to the tiling shown in fig. 2. For the central 6 rhombi, the B-spline function is:

$$\beta^1(u, v) = 1 - \frac{2}{3} \frac{a(u, v) + b(u, v)}{d} + \frac{1}{3} \frac{a(u, v)b(u, v)}{d^2} \quad (3)$$

For the outer 6 rhombi, the B-spline function is:

$$\beta^1(u, v) = \frac{1}{3} \left(2 - \frac{a(u, v)}{d} \right) \left(2 - \frac{b(u, v)}{d} \right) \quad (4)$$

For both equations, variables $a(u, v)$ and $b(u, v)$ are given in fig. 2. Outside these 12 rhombi, the linear B-spline is equal to 0.

Linear spline approximations on hexagonal grids are obtained with:

$$s(u, v) = \sum_{k \in \mathbb{Z}} c(k) \beta^1(u - u_k, v - v_k) \quad (5)$$

where (u_k, v_k) are the oblique coordinates of control point k . $c(k)$ is the coefficient of control point k . When the linear B-spline support is taken into account, the image plane is tiled with 3 types of rhombus shown in white, light grey and dark grey in fig. 3 (a). This tiling is identical to the tiling for the analytic form of the B-splines in fig. 2. Each pixel is influenced by the 4 nearest control points as shown for the light grey rhombus in fig. 3 (a). For the other two rhombus types, the 4 control points are obtained by rotation. As a result, the sum defining the spline in eq. (5) has only 4 non zero terms for linear splines on hexagonal grids. For comparison, the tiling resulting from linear spline approximation on a square grid is given in fig. 3 (b): the plane is tiled with identical squares. Like for hexagonal grids, each point in the plane is influenced by the 4 nearest control points: the corners of the square tile. Figs. 3 (a) and (b) show that the tiling obtained with the hexagonal grid is finer and more direction selective than with the square grid.

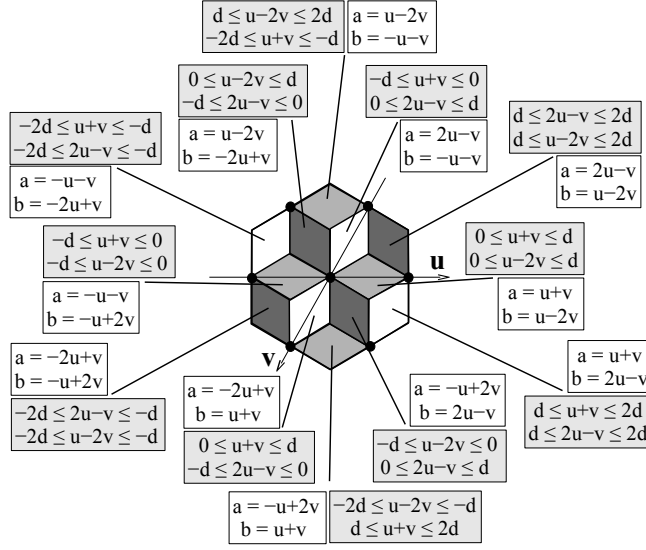


Figure 2: Tiling for the analytic form of the linear B-spline for hexagonal grids. The inequalities used to detect each rhombus are given in the grey rectangles. For the central 6 rhombi, a and b should be used with eq. 3. For the outer 6 rhombi, a and b should be used with eq. 4. The colour coding of the rhombi reflects a and b .

The analytic form of the linear spline approximation is obtained from eq. (5) and the B-spline formula in eqs. (3) and (4). For the configuration of control points shown in fig. 3 (a), this gives after simplifications:

$$\begin{aligned}
 s(u, v) = & c_1 + \frac{a(u, v)}{3d} (c_4 + c_2 - 2c_1) + \frac{b(u, v)}{3d} (c_3 + c_2 - 2c_1) \\
 & + \frac{a(u, v)b(u, v)}{3d^2} (c_1 + c_2 - c_3 - c_4)
 \end{aligned} \tag{6}$$

where c_i is the spline coefficient of control point C_i . d is the distance between control points. This formula can be used for the three rhombus types by rotating the control points, with $a(u, v)$ and $b(u, v)$ given in fig. 2. The spline approximation is a continuous signal. It can hence be used to estimate pixels placed on a square or on a hexagonal grid, regardless of the grid formed by the control points.

4 Image reconstruction from irregularly placed samples

The reconstruction algorithm is based on [3]: several linear spline approximations, each for a different transformation of the control point grid, are merged to obtain an invariant image. We improve this using linear splines with control points on a hexagonal grid.

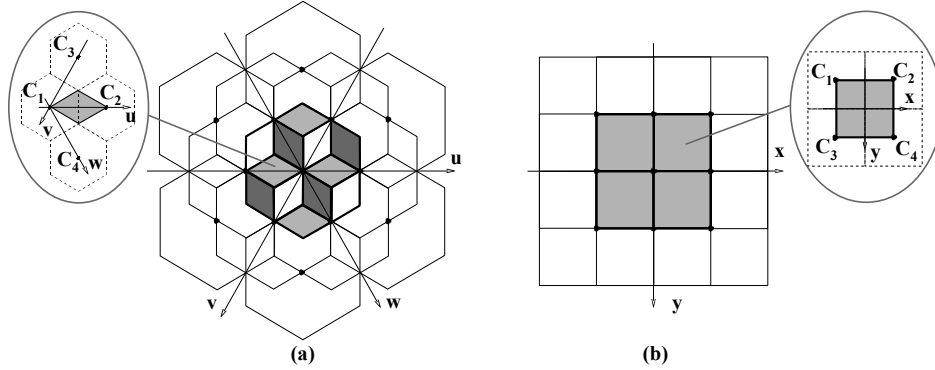


Figure 3: Image plane tiling for the linear spline approximation for (a) hexagonal grids and (b) square grids. The tiling is shown for the central hexagon and square (drawn in thick lines). The supports and control points of all used B-splines are shown with thin lines and dots respectively. Also shown is which four control points influence a tile.

4.1 Linear spline approximation from irregular samples

As in [3], we approximate the image represented by samples $f(u_i, v_i)$ with a linear spline. This is achieved by choosing spline coefficients $c(k)$ that minimise:

$$\sum_i (s(u_i, v_i) - f(u_i, v_i))^2 \quad (7)$$

where $s(u_i, v_i)$ is the linear spline approximation at sample point (u_i, v_i) . This yields the linear spline with the best fit to the samples. However, the samples are irregularly placed, so the distance between adjacent samples can be greater than twice the distance between control points. As a result, some spline coefficients might not have any constraint from the samples. To overcome this, a regularisation term is used, as in [3]. It adds constraints between coefficients of adjacent control points so that the function is linearly interpolated between samples. The function to minimise becomes:

$$\sum_i (s(u_i, v_i) - f(u_i, v_i))^2 + \varepsilon \iint_I (s_x^2 + s_y^2) dx dy \quad (8)$$

where s_x and s_y are the partial derivatives of the spline in x and y directions. The integration is performed over the whole image. ε is a small parameter which should be adapted to the desired amount of smoothing during spline fitting. In this paper, $\varepsilon = 10^{-2}$.

The spline approximation is a linear combination of the spline coefficients: $s(u_i, v_i) = \sum_k c(k) \beta^1(u_i - u_k, v_i - v_k)$ (eq. (5)). The regularisation term will be expressed as a Tikhonov regularisation matrix in section 4.2. Therefore, the optimal spline coefficients can be determined by solving a regularised linear system of equations:

$$\min_{\mathbf{c}} (\|\mathbf{M}\mathbf{c} - \mathbf{f}\|^2 + \varepsilon \|\mathbf{R}\mathbf{c}\|^2) \quad (9)$$

where matrix \mathbf{M} contains the $\beta^1(u_i - u_k, v_i - v_k)$ terms. Vectors \mathbf{c} and \mathbf{f} contain the spline coefficients $c(k)$ and the samples $f(u_i, v_i)$. \mathbf{R} is the regularisation matrix. \mathbf{M} is built from

the sample positions and the analytic form of the linear B-spline (fig. 2, eqs. (3) and (4)). Due to the limited support of the splines, \mathbf{M} and \mathbf{R} are band matrices (see sections 3 and 4.2). The system of equations can be solved with:

$$\mathbf{c} = (\mathbf{M}^T \mathbf{M} + \epsilon \mathbf{R}^T \mathbf{R})^{-1} \mathbf{M}^T \mathbf{f}. \quad (10)$$

4.2 Regularisation term for the linear spline approximation

The regularisation term R in eq. (8) is the integral of $s_x^2 + s_y^2$ over the whole image. It can be expressed as the sum for all image tiles of the integral over each image tile:

$$R = \sum_i \iint_{T_i} ((s_x(x,y))^2 + (s_y(x,y))^2) dx dy = \sum_i R_{T_i} \quad (11)$$

where T_i represents an image tile. These tiles are chosen to be the tiles shown in fig. 3, so the analytic form of the spline in a tile is given by eq. (6). Using the relation between (u, v) and (x, y) in eq. (1), the resulting integral over one tile is:

$$R_{T_i} = \frac{(c_2(i) - c_1(i))^2}{2\sqrt{3}} + \frac{(c_4(i) - c_3(i))^2}{6\sqrt{3}} + \frac{(c_1(i) + c_2(i) - c_3(i) - c_4(i))^2}{27\sqrt{3}} \quad (12)$$

for the control point configuration of fig. 3. $c_1(i)$, $c_2(i)$, $c_3(i)$ and $c_4(i)$ are the coefficients of the control points C_1 , C_2 , C_3 and C_4 in fig. 3 for tile T_i . The integrals for the two other rhombus types can be obtained from eq. (12) by rotating the control points.

R is hence a sum of regularisation terms, each modelled with a Tikhonov regularisation matrix \mathbf{R}_j . Each matrix \mathbf{R}_j models one term of eq. (12) for one rhombus type (e.g. $\sum_i (c_2(i) - c_1(i))^2$). It results in 9 regularisation matrices (3 terms in (12) for 3 rhombus types). For comparison, square grids result in 3 regularisation matrices because they use only one sort of tile. These matrices are all combined in one single term for spline fitting with eq. (10): $\mathbf{R}^T \mathbf{R} = \sum_j \mathbf{R}_j^T \mathbf{R}_j$. The regularisation matrix $\mathbf{R}^T \mathbf{R}$ is a band matrix. It depends only on the grid, so it is computed once and re-used for each spline fitting.

4.3 Complete reconstruction framework

Equation (10) provides a linear spline approximation from the irregularly placed samples. To improve reconstruction, we merge several spline approximations with each other as in [3] and according to the invariance by integration principle. For several transformations of the control point grid, a linear spline approximation is computed with eq. (10). In a second step, the obtained spline coefficients are used to estimate images of the scene. During this step, the transformations of the control point grids are compensated, so that several intensity estimations are obtained for the same scene point. Merging these estimations improves the reconstruction without inducing any blurring. In our experiments, we used the median value to merge intensities as it gave better results than averaging.

In [3], translations, rotations, scalings and affine transformations were applied. We tested: small translations (amplitude smaller than the distance between control points), rotations of the grid about the image centre or about the central control point, and small scalings. We chose to apply in the experiments in section 5 only rotations about the image centre as it was the most efficient. This approach results in both rotation and a small translation of the grid because the image centre is usually between control points. Integration over scales did not improve the results much.

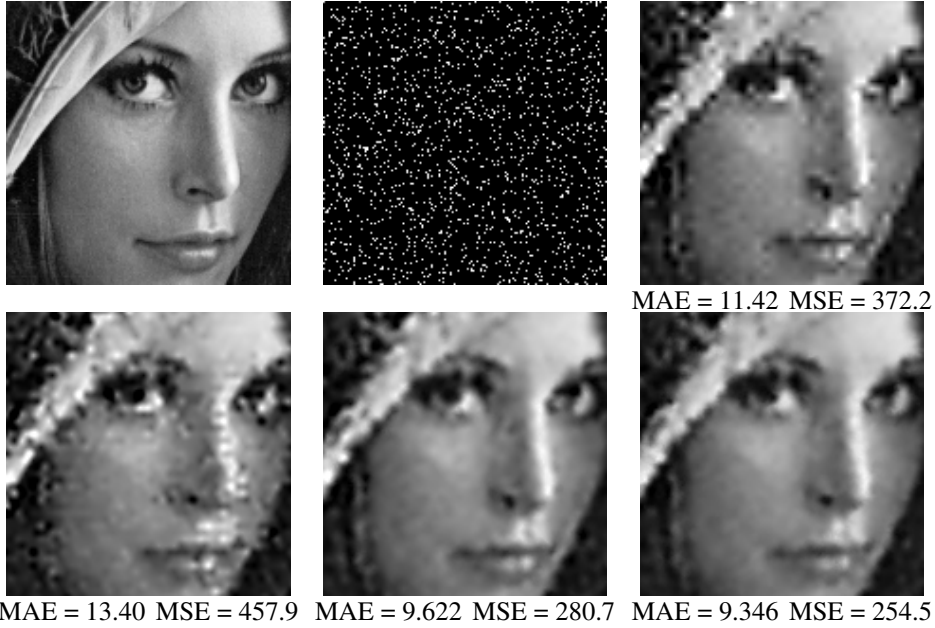


Figure 4: Invariance by integration. Top, from left to right: original image, random sampling (7% of the pixels), linear spline on square grid. Bottom, from left to right: linear spline on hexagonal grid, reconstruction with [3], reconstruction with our algorithm.

5 Reconstruction results

The algorithm was tested by subsampling 5, 7 and 10% of the pixels at random positions for many images. The distance between control points was set to $d = 4$ pixels, so the numbers of control points and of samples are similar. All results were obtained by rotating the control point grid about the image centre with an angle varying from 0 to π in steps of $\frac{\pi}{20}$. Thus, 20 linear splines were merged to get the reconstructed images. This gave a good compromise between computation time and performance. Reconstruction could be further improved by adding grid transformations or reducing the distance between control points. The intensity was reconstructed at the original pixel positions, so reconstructed and original pixels can be compared to assess reconstruction quality. Splines are continuous functions, so the fact that the original images were sampled on square grids did not influence much the reconstruction performances for hexagonal control point grids. To assess reconstruction quality, we use the mean absolute error (MAE, average of the absolute differences between original and reconstructed intensities) and the mean square error (MSE, average of the squared differences between original and reconstructed intensities).

In fig. 4, we show the influence of the “invariance by integration” framework for our algorithm and the algorithm in [3]. The difference between both algorithms is that square grids are used in [3], while hexagonal grids are used in our work. The same parameters were applied for both methods, so performance differences are caused by the grid. Reconstruction with a single linear spline is much worse than the final reconstruction for both algorithms. The linear spline reconstruction is worse for the hexagonal grid than

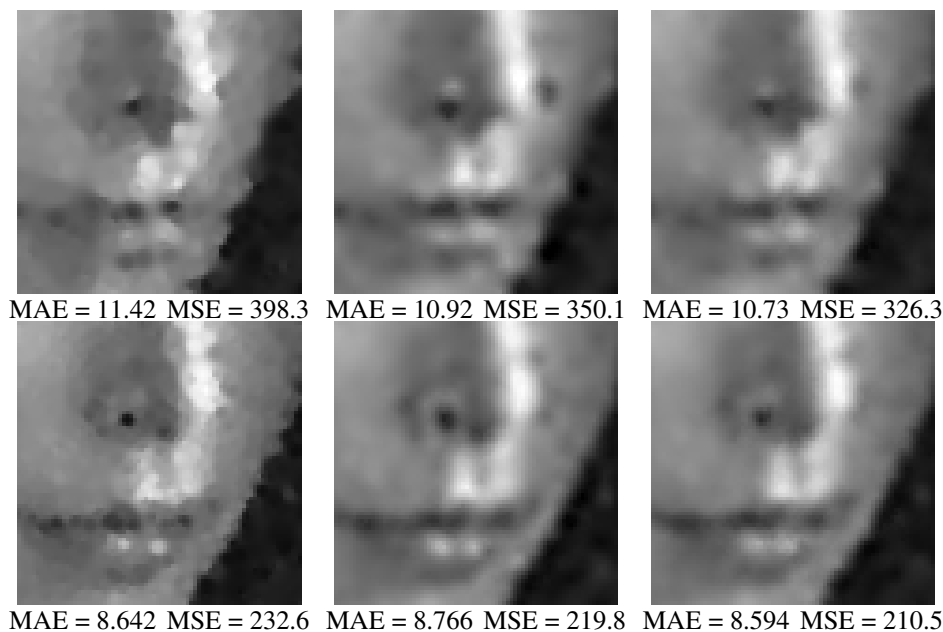


Figure 5: Reconstruction for 5% (top line) and 10% (bottom line) random sampling for normalised convolution [1] (left column), invariant reconstruction on square grids with [3] (middle column), and our algorithm (right column). A detail of the *Lenna* image is shown for better visualisation, the performances were computed on the whole image.

for the square grid: due to its direction selectivity, some directions are well modelled (e.g. left eyebrow) and some directions are badly modelled (e.g. vertical details). However after “invariance by integration”, the reconstruction with the hexagonal grid is better than the reconstruction with the square grid. It appears smoother, less noisy, and there are fewer artifacts near edges and small details (see e.g. the nostril and the face contour). These visual impressions are confirmed by the reconstruction performances given in fig. 4.

In fig. 5, the two previous algorithms are compared with the continuous normalised convolution [1], the latest version of a good state of the art algorithm to reconstruct images from irregularly placed samples. The kernel we used for [1] is $(r + 0.25)^{-4}$ where $r = \sqrt{x^2 + y^2}$, as it gave the best results. The results are given for two sampling rates to show its influence on reconstruction. For all algorithms, reconstruction quality improves with higher sampling rate. This effect is the strongest for normalised convolution. The results of normalised convolution have a very different appearance from the other two algorithms as it is not based on splines. With higher sampling rate, finer details appear with normalised convolution. This is not the case for the other two algorithms as the distance between control points is fixed. Like in fig. 4, hexagonal grids produce better results than square grids. This difference decreases with higher sampling rate.

The mean performances for the three algorithms are given in table 1 for several images. They were obtained by averaging MAE and MSE over 50 trials to obtain results independent of the sample positions. The images are representative of three image types in which all tested images could be divided. The first image (*Lenna*) contains a mix







| added noise | image | sampling rate | norm. convol. | | square grids | | hexagonal grids | |
|--------------|---|---------------|---------------|--------------|--------------|--------------|-----------------|--------------|
| | | | MAE | MSE | MAE | MSE | MAE | MSE |
| $\sigma = 0$ |  | 5% | 11.13 | 361.1 | 10.79 | 328.7 | 10.65 | 315.5 |
| | | 7% | 9.895 | 291.2 | 9.745 | 272.3 | 9.579 | 260.3 |
| | | 10% | 8.701 | 231.2 | 8.845 | 225.1 | 8.688 | 214.7 |
| |  | 5% | 5.788 | 114.4 | 5.318 | 92.38 | 5.406 | 95.18 |
| | | 7% | 5.016 | 89.81 | 4.687 | 72.60 | 4.739 | 74.98 |
| | | 10% | 4.338 | 69.74 | 4.160 | 56.98 | 4.222 | 59.24 |
| |  | 5% | 8.477 | 156.8 | 8.956 | 170.2 | 8.516 | 152.2 |
| | | 7% | 7.844 | 137.9 | 8.587 | 159.1 | 8.115 | 140.5 |
| | | 10% | 7.056 | 115.8 | 7.986 | 138.7 | 7.628 | 124.9 |
| $\sigma = 5$ |  | 5% | 11.78 | 368.9 | 11.68 | 347.9 | 11.34 | 330.1 |
| | | 7% | 10.67 | 303.0 | 10.72 | 296.7 | 10.31 | 275.6 |
| | | 10% | 9.546 | 241.6 | 9.673 | 241.3 | 9.359 | 226.5 |
| |  | 5% | 6.878 | 125.3 | 6.865 | 111.3 | 6.560 | 107.5 |
| | | 7% | 6.217 | 101.3 | 6.270 | 91.50 | 5.955 | 87.23 |
| | | 10% | 5.629 | 81.53 | 5.615 | 73.44 | 5.415 | 71.77 |
| |  | 5% | 9.241 | 170.9 | 9.853 | 191.2 | 9.208 | 166.8 |
| | | 7% | 8.608 | 148.9 | 9.408 | 175.2 | 8.767 | 151.9 |
| | | 10% | 7.954 | 127.8 | 8.773 | 154.8 | 8.292 | 137.7 |

Table 1: Mean performances over 50 trials (i.e. different random sampling and noise). The best performance for each image and each sampling rate is highlighted in boldface. The noise added to the data is zero mean Gaussian noise with standard deviation σ .

of small details and bigger objects. Our algorithm gives the best reconstruction quality for this type of image. Normalised convolution gives the worst results because it cannot model well slowly varying shading. The second image (peppers) contains mainly big objects with slowly varying shading. The edges are mostly horizontal and vertical. The algorithm in [3] performs best, as it can model best horizontal and vertical edges. However, when noise is added (bottom half of table 1), our algorithm performs best. Hence, hexagonal grids yield a more robust, less noise sensitive reconstruction. The last image (pears) contains objects with fine texture (little dots on the pear skin) which cannot be modelled well with splines with $d = 4$. As a result, normalised convolution performs best especially for higher sampling rate. For low sampling rates and noisy data, our algorithm performs similarly or better. Yet our algorithm performs better than normalised convolution if the distance between control point is reduced. For example, for $d = 3$ and 7% sampling rate, the average MAE is 7.761 and the average MSE is 130.6 for our algorithm (50 trials, no noise), which is better than the result of normalised convolution.

6 Conclusions

In this paper, we presented a method inspired by the human visual system to reconstruct images from irregularly placed samples. Several linear spline approximations are computed from the samples, each for a different transformation of the control point grid

(e.g. translations and rotations). These reconstructions are merged together after transformation compensation, yielding a higher quality image which is invariant to transformations of the control point grid. This approach was introduced in [3] for linear splines on square grids. In this work, we improved the algorithm using linear splines with control points placed on a hexagonal grid. This is inspired by nature because photosensors are roughly placed on a hexagonal grid in the fovea. Linear splines are more direction selective on hexagonal than on square grids, thanks to the resulting image tiling. In addition, 15% more control points can be fitted in an image with hexagonal than with square grids.

The reconstruction results showed that the hexagonal grid increases reconstruction quality. The images look smoother and contain fewer artifacts especially near sharp edges and small details. The reconstruction on a hexagonal grid is also more robust to noise and to small details which cannot be modelled with the spline. Last, the “invariance by integration” principle improves reconstruction more for hexagonal grids than for square grids due to the direction selectivity of the spline. The algorithm was also compared with normalised convolution. Our algorithm is usually better, especially for low sampling rates and for noise. However, normalised convolution achieves better reconstruction for fine texture which cannot be modelled well with the spline. This modelling problem may be solved by reducing the distance between the control points.

Acknowledgements This work was funded by a UK Research Councils translation grant.

References

- [1] K. Andersson, C.-F. Westin, and H. Knutsson. Prediction from off-grid samples using continuous normalized convolution. *Signal Processing*, 87:353–365, 2007.
- [2] D. H. Hubel. *Eye, Brain, and Vision*. W H Freeman, 1988.
- [3] A. Kadyrov and M. Petrou. Reverse engineering the human vision system: a possible explanation for the role of microsaccades. In *Proc. of the ICPR'04*, 2004.
- [4] S. Martinez-Conde, S. L. Macknik, and D. H. Hubel. The role of fixational eye movements in visual perception. *Nature Reviews Neuroscience*, 5:229–240, 2004.
- [5] L. Middleton and J. Sivaswamy. *Hexagonal Image Processing: A Practical Approach*. Advances in Pattern Recognition. Springer-Verlag, 2005.
- [6] I. Overington. *Computer Vision: A Unified Biologically-Inspired Approach*. Elsevier Science Publishers B.V., 1992.
- [7] R. Piroddi and M. Petrou. Analysis of irregularly sampled data: A review. In *Advances in Imaging and Electron Physics*, volume 132, pages 109–165. Elsevier, 2004.
- [8] D. Van De Ville, T. Blu, M. Unser, W. Philips, I. Lemahieu, and R. Van de Walle. Hex-splines: A novel spline family for hexagonal lattices. *IEEE Transactions on Image Processing*, 13(6):758–772, 2004.
- [9] D. Van De Ville, W. Philips, and I. Lemahieu. Least-squares spline resampling to a hexagonal lattice. *Signal Processing: Image Communication*, 17:393–408, 2002.