A Super-Resolution Image Reconstruction using Natural Neighbor Interpolation

Christian J. Enríquez-Cervantes, Ramón M. Rodríguez-Dagnino

Tecnologico de Monterrey, Electrical and Computing Engineering Department, Monterrey, Mexico

christian.enriquez@itesm.mx, rmrodrig@itesm.mx

Abstract. A super-resolution image reconstruction algorithm using natural neighbor interpolation is proposed and its performance is evaluated. The algorithm is divided into two stages: image registration and the reconstruction of a high-resolution color image. In the first stage, as shifts between images are usually unknown, the algorithm computes an approximation of these displacements by solving the system of linear equations proposed by Keren, Peleg, and Brada, then the pixels of all low-resolution images are mapped into a high-resolution grid by computing the new coordinates using the motion vectors. In the second stage, the pixel values that match the high-resolution grid are interpolated using natural neighbor interpolation which is a weighted average interpolation method for scattered data, based in the areas of the Voronoi polygons of the neighboring pixels. Finally, the proposed natural neighbor super-resolution algorithm is compared with some popular super-resolution algorithms presented in literature.

Keywords. Super-resolution, natural neighbor interpolation, motion estimation, high-resolution image reconstruction.

1 Introduction

High quality digital images are often needed and desired in many applications. Digital images are used in a variety of areas such as medical imaging, surveillance, astronomy, microscopy, consumer electronics, and computer vision. In recent years the popularity and use of digital images have increased significantly, therefore, the demand for high quality images has increased accordingly.

As an example, a practical application where high detail images can be used is recognition of the license plate number of a car in an image taken by a surveillance camera. Another example is facial recognition systems where the algorithm processing time is reduced and the quality of obtained results is increased if the images employed as inputs are detailed.

In order to increase the "quality" or "detail level" of an image, its spatial resolution must be increased. Spatial resolution is defined as the number of independent pixels per unite of area of an image; in other words, it is a measurement of pixel density. Spatial resolution depends largely on the optical system that acquires the image, since the number of independent pixels per unite area of a digital image is limited by the density of imaging sensors in the array of the system. Therefore, in order to increase the quality of images it is necessary to increase the density of imaging sensors in the array. These sensors are usually distributed into a two-dimensional (2D) array in an image acquisition system, the higher the sensor density in the array, the higher the spatial resolution of images captured. The size of this array of sensors is limited, so spatial resolution has a physical limit. Beside physical limitations in the imaging sensors, there is also the disadvantage of a high cost in the manufacturing of these devices.

An alternative to increase spatial resolution of an image without decreasing the size of the sensors is image processing techniques. Image processing techniques are commonly cheaper to implement because they do not require specialized hardware used in high-resolution imaging systems, which means lower manufacturing costs and reducing the final price. For these reasons image processing techniques arise as alternatives to improve image resolution at lower cost. One of the most innovative image processing techniques deeply studied

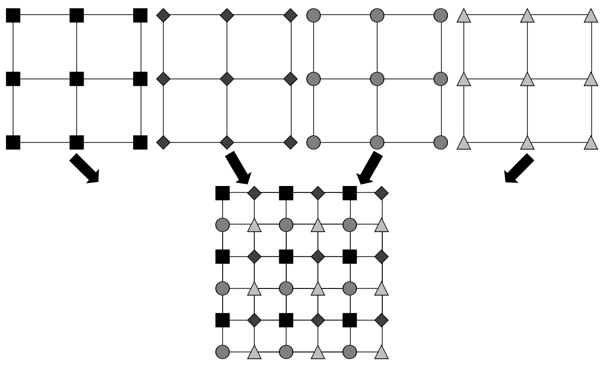


Fig. 1. Information from multiple LR images is used in order to reconstruct an HR image

in recent years is called super-resolution image reconstruction.

Super-resolution is an image reconstruction technique that consists in obtaining a high-resolution (HR) image from multiple low-resolution (LR) images of the same scene with small shifts in each image relative to a reference image. The basic premise is to reconstruct an HR image from pixel information that LR images provide.

In super-resolution there is a condition to be fulfilled: the pixel information must be different in each LR image. That is, if images have integer unit shifts between them, then there is no new information available and super-resolution cannot be achieved. In other words, by fusing new pixel value information from each input image, a new HR image is created. In Fig.1 the basic principle of super-resolution is illustrated.

As explained above, some conditions must be met in order to achieve super-resolution:

1. Availability of several LR images capturing the same scene.

2. Low-resolution images must have sub-pixel displacements between them.

3. Low-resolution images are aliased or sub-sampled by an image acquisition system.

The number of LR images to be used must be selected according to the value of the up-scaling factor of an HR image; this means, if only a few LR images are used as input and the algorithm is set to obtain an output image with a high up-scale factor, then bad quality results will be obtained. Otherwise, if too many images are used as input and an output image with a low up-scale factor is desired, then the computation time will be too big compared to the time consumed in the processing of the same output image using fewer input images.

In order to fulfill the second condition, displacements can be global or local, as well as translational or rotational. If the movement between images is very large, the reconstructed image will be of poor quality. Small displacements are measured with respect to a reference image.

Spatial aliasing could be explained as distortion present in the output images caused by the sub-sampling of a continuous scene digitized by an image acquisition system. In super-resolution algorithms information with which we reconstruct the HR image is embedded in the form of spatial aliasing in the LR images. Therefore, it is required that the sensors used to capture the LR images be weak enough so that aliasing be present.

Super-resolution image reconstruction algorithms usually differentiate three stages (see Fig.6 in [17]). These stages can be implemented separately or simultaneously and they are *1*) Image registration, *2*) Interpolation, and *3*) Restoration.

The first stage is estimation of motion information of a set of input LR images and arrangement of LR pixels into an HR grid. This stage is referred to in the literature as image registration. There are several algorithms proposed to calculate relative displacements between images [4, 7, 13, 14, 22]. As explained earlier, a condition that must be met to achieve super-resolution is the existence of sub-pixel shifts between the input images and the original scene; in most super-resolution methods these displacements are assumed to be known, however, in practice this is not always true. Motion estimation is a critical step in image reconstruction, the quality of the estimated image and the computation time depend largely on motion estimation results. If estimation is not precise, the result will be a lower quality image with a high processing time. Although motion estimation is an important stage to achieve super-resolution, most proposed algorithms do not focus on it. A motion estimation algorithm calculates the global translations in horizontal and vertical axis respectively as well as the rotation angle for each image. Shifts between LR images are non-uniform and may be global or local. For purposes of simplicity, translational and rotational displacements will be assumed to be global for each LR image.

After motion estimation is performed, new coordinates for each LR image are calculated using the computed displacements and then arranged into an HR grid. Once the pixels are arranged within the grid they remain non-uniformly distributed along it. Therefore, it is necessary to perform a 2D interpolation for scattered data in order to calculate pixel values aligned with the HR grid using neighboring pixel values. In particular, we have selected the natural neighbor interpolation for scattered data proposed by Sibson [24].

Finally, as a third stage in some algorithms, an image restoration method is applied to the interpolated image in order to remove blurring and noise and the output HR image is obtained. There are several super-resolution algorithms which implement, simultaneously or separately, two or three of the steps previously seen.

The main contribution of this work is the development of a new super-resolution image reconstruction algorithm using the natural neighbor interpolation that has a better performance than other popular algorithms proposed in literature in terms of improving spatial resolution of resulting images. The natural neighbor is a local multivariate interpolation technique for scattered data, which is widely used in data interpolation in geophysical and terrain modeling. This interpolation based on areas has been previously used for 2D image reconstruction to generate intermediate images in animations [3], but it has never been used for reconstruction of an HR image using a set of LR images. It is noteworthy that the combination of algorithms for motion estimation and the natural neighbor interpolation studied in this work have not been proposed before in super-resolution image reconstruction literature.

In this paper the first two stages of super-resolution seen previously are implemented separately and the paper is organized as follows. In Section 2, motion is estimated by solving the system of linear equations proposed by Keren, Peleg, and Brada [14]. The coefficients of this system of linear equations are calculated by convolving the LR images with a bivariate Gaussian function and its partial derivatives. The convolutions are computed numerically in the frequency domain using the Fast Fourier Transform (FFT) algorithm. The LR pixels are then mapped into an HR grid by computing their coordinates using motion vectors.

Once the displacements are calculated, in Section 3 the natural neighbor interpolation algorithm for scattered data is implemented. The natural neighbor interpolants are constructed by calculating the areas of the Voronoi polygons using the compound signed decomposition algorithm. The third stage of super-resolution is omitted in this work. There are a lot of algorithms for image restoration and it is left to the reader's choice which one to use.

The proposed natural neighbor super-resolution algorithm is compared with three popular super-resolution image reconstruction algorithms in order to evaluate its performance. Spatial resolution improvement is measured statistically by comparing the output images of each algorithm against an HR image created from input LR images. Images are compared using the normalized cross correlation factor (NCC) as a measure of similarity. Also, the HR reconstructed image of a license plate is included to show a practical application of the algorithm. Finally, experiments and results are shown in Section 4 while conclusions are discussed in Section 5.

2 Image Registration

2.1 Motion Estimation (ME)

The quality of the reconstructed HR image depends largely on the results obtained in the ME stage. For this reason, it is required to define a method that calculates accurately the sub-pixel shifts between the LR frames.

Several methods for ME have been proposed in literature; however, most methods are not accurate enough for the sub-pixel accuracy required as a condition to achieve super-resolution. The authors of [14] proposed a method for image registration which is accurate enough at the sub-pixel level required. They proposed a system of linear equations (SLE) defining the spatial relationship between a reference image and warped images starting from basic geometric operations. The operations of rotation about any point and displacements are defined in the following equations (see Appendix for details):

$$x' = (x - x_o) \cdot \cos(\theta) - (y - y_o) \cdot \sin(\theta) + x_o + a, \tag{1}$$

$$y' = (y - y_o) \cdot \cos(\theta) + (x - x_o) \cdot \sin(\theta) + y_o + b. \tag{2}$$

From the basic geometric operations (1) and (2) they derived the following SLE with displacements a, b, and rotation angle θ as unknowns, the reference and warped images defined as the functions f and g respectively with spatial variables x and y and partial derivatives f_x and f_y:

$$\begin{gathered}
\left[\sum_{x,y} f_x^2\right] a + \left[\sum_{x,y} f_x f_y\right] b + \left[\sum_{x,y} R f_x\right] \theta = \\
\sum_{x,y} [f_x \cdot (g - f)] \\
\left[\sum_{x,y} f_x f_y\right] a + \left[\sum_{x,y} f_y^2\right] b + \left[\sum_{x,y} R f_y\right] \theta = \\
\sum_{x,y} [f_y \cdot (g - f)] \\
\left[\sum_{x,y} R f_x\right] a + \left[\sum_{x,y} R f_y\right] b + \left[\sum_{x,y} R^2\right] \theta = \\
\sum_{x,y} [R \cdot (g - f)]
\end{gathered} \tag{3}$$

where

$$R = f_y \cdot (x - x_0) - f_x \cdot (y - y_o). \tag{4}$$

Detailed derivations of the equations in (3) can be found in Appendix. The SLE is only valid for small values of the angle θ. It is noteworthy that the subtractions in the right-hand side of (3) are slightly different from the independent terms shown in [14]. This was corrected later in [12] and [13], however, there are still differences in the summations of independent terms in the right-hand side of the equation. It is also noteworthy that the term R is different from the works mentioned above since the SLE computes the rotations about any point (x_o, y_o) (see Appendix).

In order to solve the SLE in (3) for the unknowns a, b, and θ, it is necessary to calculate the coefficients of the system in the left-hand side and the independent terms in the right-hand side. To compute the above terms, the functions f and g must be defined and the partial derivatives f_x and f_y calculated.

Usually the images f and g are not defined as analytical functions. They are defined as a matrix of size $M \times N$ containing the discrete pixel value intensities arranged in rows and columns according to their coordinates. To calculate the partial derivatives of the function f, we take advantage of implicit Gaussian blur h in images as well as the derivative property of the 2D convolution.

The Gaussian blur is defined as the result of blurring an image by a Gaussian function. In two dimensions, the circular Gaussian function is the distribution function for uncorrelated variables x and y having a bivariate normal distribution and equal standard deviation $\sigma = \sigma_x = \sigma_y$, and it is expressed as

$$h = \frac{1}{2\pi\sigma^2} \cdot \exp\left[-\left(\frac{(x - x_o)^2 + (y - y_o)^2}{2\sigma^2}\right)\right]. \tag{5}$$

Mathematically, applying a Gaussian blur to an image is the same as convolving an image f' with a Gaussian function h where f' is the original reference image:

$$f = f' * h. \tag{6}$$

We take advantage of the convolution derivative property for functions of several variables which states that "the derivative of a convolution is the convolution of either of the functions with the derivative of the other" [6]. Applying this property to (6) we obtain

$$f_x = [f' * h]_x = h_x * f',$$

$$f_y = [f' * h]_y = h_y * f'.$$

Therefore, it is only necessary to obtain the partial derivatives of the bivariate Gaussian function:

$$h_x = -\frac{(x-x_o)}{2\pi\sigma^4}\cdot\exp\left[-\left(\frac{(x-x_o)^2+(y-y_o)^2}{2\sigma^2}\right)\right], \tag{7}$$

$$h_y = -\frac{(y-y_o)}{2\pi\sigma^4}\cdot\exp\left[-\left(\frac{(x-x_o)^2+(y-y_o)^2}{2\sigma^2}\right)\right]. \tag{8}$$

Since images f and g are stored as a matrix of discrete pixels, it is necessary to produce discrete versions of partial derivatives using formulas (7) and (8). Then, these discrete finite versions of the partial derivatives are convolved with the reference image f' in order to calculate the coefficients of (3). To compute the convolution, we will use the 2D discrete Fourier Transform (DFT) and the theorem of circular convolution. The spatial convolution between a digital image and a Gaussian function or its partial derivatives can be calculated using the backward DFT of the product of two forward DFTs. The Fast Fourier Transform (FFT) is an efficient algorithm to compute forward and backward DFTs. From now on the notation $\mathrm{FFT}[f]$ will be used to indicate that the forward DFT of an image f is computed using the FFT algorithm, and the notation $\mathrm{IFFT}[f]$ means that backward FFT algorithm is implemented. Finally, in order to compute the nine coefficients on the left-hand side of (3), the following three terms are calculated first:

$$\mathbf{T}_1 = \quad f \quad = \mathrm{IFFT}\left\{\mathrm{FFT}\left[f'\right]\cdot\mathrm{FFT}\left[h\right]\right\}, \tag{9}$$
$$\mathbf{T}_2 = \quad f_x \quad = \mathrm{IFFT}\left\{\mathrm{FFT}\left[f'\right]\cdot\mathrm{FFT}\left[h_x\right]\right\}, \tag{10}$$
$$\mathbf{T}_3 = \quad f_y \quad = \mathrm{IFFT}\left\{\mathrm{FFT}\left[f'\right]\cdot\mathrm{FFT}\left[h_y\right]\right\}, \tag{11}$$

where $0 \le x < M$ and $0 \le y < N$. Then we proceed to calculate the term R in (4) using the terms (10) and (11):

$$\mathbf{R} = \mathbf{T}_3\cdot(x-xo) - \mathbf{T}_2\cdot(y-yo). \tag{12}$$

The following matrix of coefficients is calculated using the terms (10), (11), and (12) according to the left-hand side in (3):

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \tag{13}$$

where

$$\begin{aligned} a_{11} &= \sum_{x,y}\mathbf{T}_2\cdot\mathbf{T}_2, \quad a_{21} = \sum_{x,y}\mathbf{T}_2\cdot\mathbf{T}_3, \\ a_{31} &= \sum_{x,y}\mathbf{R}\cdot\mathbf{T}_2, \\ a_{21} &= \sum_{x,y}\mathbf{T}_2\cdot\mathbf{T}_3, \quad a_{22} = \sum_{x,y}\mathbf{T}_3\cdot\mathbf{T}_3, \\ a_{23} &= \sum_{x,y}\mathbf{R}\cdot\mathbf{T}_3, \\ a_{31} &= \sum_{x,y}\mathbf{R}\cdot\mathbf{T}_2, \quad a_{32} = \sum_{x,y}\mathbf{R}\cdot\mathbf{T}_3, \\ a_{33} &= \sum_{x,y}\mathbf{R}\cdot\mathbf{R}. \end{aligned} \tag{14}$$

Then, each shifted input LR image g'_k is convolved with the Gaussian function. This yields a fourth term for the k-th warped image:

$$\mathbf{T}_4{}^k = g_k = \mathrm{IFFT}\left\{\mathrm{FFT}\left[g'_k\right]\cdot\mathrm{FFT}\left[h\right]\right\}. \tag{15}$$

Finally, the constant term vector is computed using (9), (10), (11), and (15) according to the right-hand side in (3):

$$\mathbf{b}_k = \begin{bmatrix} b_{1k} & b_{2k} & b_{3k} \end{bmatrix}^T, \tag{16}$$

where

$$\begin{aligned} b_{1k} &= \sum_{x,y}\mathbf{T}_2\cdot\left[\mathbf{T}_4{}^k - \mathbf{T}_1\right], \\ b_{2k} &= \sum_{x,y}\mathbf{T}_3\cdot\left[\mathbf{T}_4{}^k - \mathbf{T}_1\right], \\ b_{3k} &= \sum_{x,y}\mathbf{R}\cdot\left[\mathbf{T}_4{}^k - \mathbf{T}_1\right]. \end{aligned} \tag{17}$$

Therefore the SLE is as follows:

$$\mathbf{A}\mathbf{m}_k = \mathbf{b}_k,$$

where $\mathbf{m}_k$ is the k-th ME vector. To find $\mathbf{m}_k$, the matrix $\mathbf{A}$ is inverted such that

$$\mathbf{m}_k = \mathbf{A}^{-1}\mathbf{b}_k = \begin{bmatrix} a_k & b_k & \theta_k \end{bmatrix}^T. \tag{18}$$

It is noteworthy that the nine coefficients of the matrix $\mathbf{A}$ in (13) are computed once, so only the function g and consequently the vector $\mathbf{b}_k$ in (16) have to be calculated for each image. The matrix $\mathbf{A}$ is easily invertible using the adjoint method.

The ME algorithm is summarized below.

Algorithm 1 Calculate a_k,b_k,θ_k

```
Calculate $h$ [eq. (5)].
Calculate $h_x$ and $h_y$ [eq. (7) and eq. (8)].
Calculate FFTs of $f'$, $h$ , $h_x$ and $h_y$.
Calculate terms $\mathbf{T}_1$, $\mathbf{T}_2$ and $\mathbf{T}_3$ [eq. (9), eq. (10) and eq. (11)].
Calculate term $\mathbf{R}$ [eq. (12)].
Calculate matrix $\mathbf{A}$ [eq. (13) and eq. (14)].
Inverse matrix $\mathbf{A}$.
for $k = 1$ to $Number\_of\_Images$ do
    Compute ${\mathbf{T}_4}^k$ [eq. (15)].
    Compute ${\mathbf{T}_4}^k - \mathbf{T}_1$.
    $b_{1k} = \sum_{x,y} \mathbf{T}_2 \cdot \left[{\mathbf{T}_4}^k - \mathbf{T}_1\right]$ ,
    $b_{2k} = \sum_{x,y} \mathbf{T}_3 \cdot \left[{\mathbf{T}_4}^k - \mathbf{T}_1\right]$ ,
    $b_{3k} = \sum_{x,y} \mathbf{R} \cdot \left[{\mathbf{T}_4}^k - \mathbf{T}_1\right]$ [eq. (17)],
    $\mathbf{b_k} = [b_{1k}, b_{2k}, b_{3k}]$ [eq. (16)].
    $\mathbf{m}_k = \mathbf{A}^{-1}\mathbf{b}_k$ [eq. (18)].
end for
```

2.2 High-Resolution Grid Mapping

Once the motion parameters of each image are estimated, we proceed to map the LR images into the desired HR grid. Using the ME vector $\mathbf{m_k}$, the new LR image coordinates are calculated combining (1) and (2), and taking into account the horizontal and vertical up-scale factors F_x and F_y of the desired HR image. This yields

$$x' = F_x \cdot (x - x_o) \cdot \cos(\theta_k) - F_y \cdot (y - y_o) \cdot \sin(\theta_k) + F_x x_o + a_k F_x, \tag{19}$$

$$y' = F_y \cdot (y - y_o) \cdot \cos(\theta_k) + F_x \cdot (x - x_o) \cdot \sin(\theta_k) + F_y y_o + b_k F_y. \tag{20}$$

Using the coordinates obtained in the above equations, each LR image g_k is mapped on the desired HR grid. As an example, in Fig. 2 three LR images are translated and rotated with respect to a reference image (square pixels image). Then the motions of each image are estimated. The new coordinates into the HR grid with the up-scale factors $F_x = F_y$ are computed using (19) and (20). Finally, all the pixels of the LR images are mapped in the HR grid as shown in Fig.3. Only the pixels within the HR grid are taken into account for the scattered interpolation.

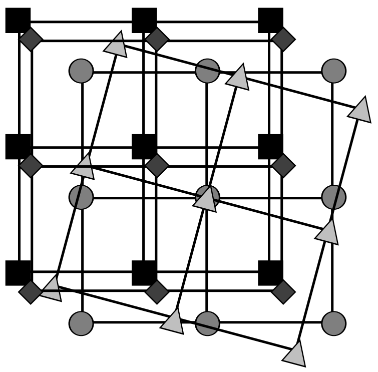


Fig. 2. Warped LR images

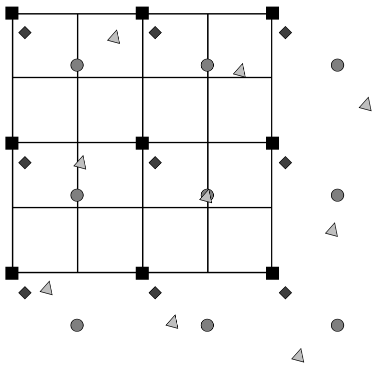


Fig. 3. Low resolution pixels mapped into the HR grid

3 High-Resolution Color Image Reconstruction

Since the shifts between the LR images are arbitrary and with sub-pixel precision, the mapped LR pixels do not match the HR grid as shown in Fig.3. Therefore, it is necessary to perform a multivariate non-uniform data interpolation in order to estimate the points that match with the grid.

The ME from Section 2 is performed in the luminance component Y of a multi-component *YCrCb* color system. The same principle can be used in color image spatial interpolation. Since non-uniform data interpolation is more complex than a uniform data interpolation, it is preferable to perform a robust scattered data interpolation on Y component, using the mapped pixels on the HR grid of Section 2, and a uniform interpolation on chrominance channels using the original Cr and

Cb components of the reference LR image. A simple and fast spatial uniform interpolation to perform is bilinear interpolation.

Within the classification of scattered data spatial interpolation methods are local techniques [1, 2]. Local interpolation methods have an advantage: interpolation is continuous at data points and smooth around them. In these local techniques the interpolation function is influenced by the neighboring data points of the interpolated point. A robust local coordinate technique is natural neighbor interpolation.

3.1 Natural Neighbor Interpolation

The natural neighbor is a local multivariate interpolation technique for scattered data, which is widely used in data interpolation and modeling of geophysical phenomena. It was first introduced by Sibson in 1981 [24]. This method is based on Voronoi diagrams and its dual graph Delaunay triangulation of a given set of discrete scattered spatial points. Natural neighbor interpolation is a weighted average method, where the weights are defined by the area stolen by the interpolated point when simulating its insertion in the Voronoi diagram of scattered points. This is related to the concept of local coordinates proposed by Sibson. To better understand the mechanics of natural neighbor interpolation and local coordinates we first review some basic concepts and properties of the Voronoi diagram and the Delaunay triangulation.

The Voronoi diagram, also known as Voronoi tessellation or Dirichlet tessellation, is a geometric partitioning of a given space into regions based on a set of scattered data points called sites or centroids. Assuming that a given space is a 2D Euclidean space, the space is divided into regions called Voronoi cells associated to a site. The Voronoi cells are determined by the distance from a given site to the other sites in the plane. Namely, a Voronoi cell is the set of all points in the given space whose distance to the associated site is no greater than their distance to the other sites. Consider a set of distinct sites in the Euclidean 2D space $\mathbf{X} :\in \mathbb{R}^2$

$$\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N\}\,.$$

The Voronoi cell $\mathbf{V}_n$ associated to site $\mathbf{p}_n$ is defined as a set of all points in $\mathbf{X}$ whose distance to the site $\mathbf{p}_n$ is no greater than their distance to other sites $mathbfp_m$, where n is an index different from m. This is formalized as follows:

$$\mathbf{V}_n = \{\mathbf{x} \in \mathbf{X}\ :\ d(\mathbf{x}, \mathbf{p}_n) < d(\mathbf{x}, \mathbf{p}_m)\ \forall\ n \neq m\}\,, \tag{21}$$

where $d(\mathbf{x}, \mathbf{p})$ is the Euclidean distance from the point $\mathbf{x}$ to the site $\mathbf{p}$. Then, the Voronoi diagram is the set of all Voronoi cells $\mathbf{V}_n$ in the space $\mathbf{X}$.

In the field of image processing a given space is a 2D finite Euclidean space. If we apply the concepts seen above to the HR grid studied in Section 2, then the given space $\mathbf{X}$ is the HR grid divided into a set of regions. If each mapped pixel is considered a site in the space $\mathbf{X}$, then each divided region is associated with a pixel mapped according to the new coordinates that were calculated using the ME vectors, such that any point inside a specific region is closer to that pixel associated with the region than to any other pixel in the grid. In the simplest case, if only two pixels are mapped into the grid, the HR grid is divided into two half planes bounded by the perpendicular bisector of both pixels. If more pixels are uniformly inserted into the grid, then each region will intersect $n-1$ Voronoi cells and thus convex polygons are formed in the grid as shown in Fig.4.

One important property of Voronoi diagrams is the duality with the Delaunay triangulation for the same set of sites in the Euclidean space.

The Delaunay triangulation, also known as Delaunay tessellation, was first introduced by Delaunay in 1934 as cited in [15]. Delaunay triangulation is based on the concept of a Voronoi diagram that divides the plane into a set of regions bounded by convex polygons and each region is associated with a single point or site, then the Delaunay triangulation is constructed by connecting the points with which the Voronoi cells have common boundaries [30]. It is said that Voronoi diagrams and Delaunay triangulation are dual structures because both contain the same information but represented in a different form. There is a Delaunay triangle edge between two points or sites only if their Voronoi cells are adjacent as can be seen in Fig.5 for the same Voronoi diagram shown in Fig.4.

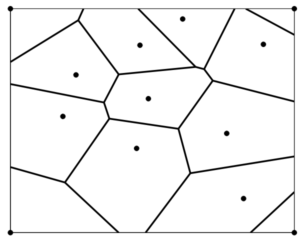


Fig. 4. Voronoi diagram of thirteen scattered points

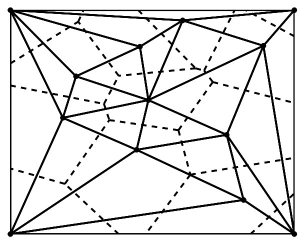


Fig. 5. Delaunay triangulation and its dual graph the Voronoi tessellation

A property of the Delaunay triangulation is that it maximizes the minimum angle in each single triangle avoiding the construction of skinny triangles, this is a desired property for finite element triangulation [30]. A Delaunay triangle is validated using the empty circumcircle criterion, this criterion states that if a circle circumscribing any Delaunay triangle does not contain any other points within its circumference, then the line connecting two vertices of the triangle that intersects the circumference is an edge of a Delaunay triangle. Formally, the empty circumcircle criterion is the name of the theorem that states that for every finite set of points in a plane, there is a minimum number of circles such that every point in the convex hull of the set of points lies exactly in the circumference of one or more of the circles, but none of the points lies within the circle. This means that each point lies on one or more circles but not in the interior of any. Fig.6 illustrates the empty circumcircle criterion that validates some of Delaunay triangles in Fig.5.

As explained before, natural neighbor interpolation is a local non-uniform data interpolation technique based on local coordinates. In local interpolation techniques, points influencing the interpolator are the neighbors surrounding a given point to interpolate. The level of "influence" of each neighbor over the interpolate point is defined by the local coordinates of the given point. More properties of local coordinates are studied in [9,21,23].

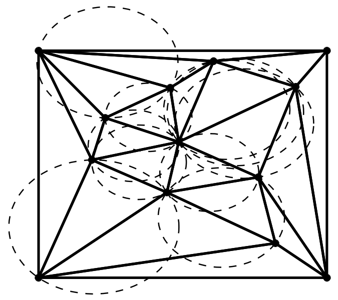


Fig. 6. Empty circumcircle criterion

As seen in Fig.4, the Voronoi diagram partitions the plane into segments or tiles called Voronoi cells defined formally in (21). By observing the equation again, each Voronoi cell encloses the point $\mathbf{p}_n$, then the Voronoi polygon $\mathbf{V}_n$ is composed of the points that are closer to the given point $\mathbf{p}_n$ than any other point $\mathbf{p}_m$. Hence, it is said that the point $\mathbf{p}_n$ is a natural neighbor of the point $\mathbf{p}_m$ if and only if their Voronoi polygons $\mathbf{V}_n$ and $\mathbf{V}_m$, respectively, share a common edge.

Translating the above definition to the concept of circumcircles, two data points are natural neighbors if the circle passing through them and a third point does not enclose a fourth data point. Therefore, if these three points are contiguous, they are natural neighbors meeting the Delaunay empty circumcircle criterion, therefore, these points are the vertices of a Delaunay triangle. Then, it is obvious that the minimum number of natural neighbors of a given point in a Voronoi diagram is two (when $n \geq 3$). The given point and its natural neighbors form the elemental natural neighbor triplets in Voronoi diagrams.

In order to build the natural neighbor interpolant, it is necessary to construct a Voronoi diagram from the original scattered data point set $\mathbf{S}$, then we proceed to simulate the insertion of a query point $\mathbf{q}$, which is the point to be interpolated. This new point $\mathbf{q}$ creates a new Voronoi polygon that overlaps the

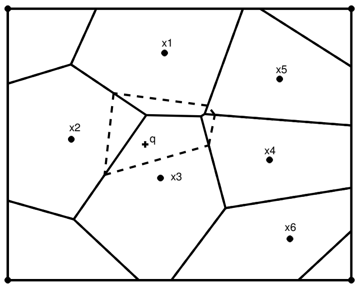


Fig. 7. Voronoi tile of the query point $\mathbf{q}$ overlaps the tiles of points

original Voronoi tessellation as it can be seen in Fig.7. Given (21), the new Voronoi cell created by the query point is defined by

$$\mathbf{V(q)} = \{\mathbf{z} \in \mathbf{S} \ : \ d(\mathbf{z}, \mathbf{q}) < d(\mathbf{z}, \mathbf{x}_i) \ \forall \ i = 1 \dots n\} \,, \tag{22}$$

where $\mathbf{z}$ is the subset of points in the original set $\mathbf{S} \in \mathbb{R}^2$ inside the Voronoi cell, $\mathbf{q}$ is the query point, $\mathbf{x_i}$ is the i-th natural neighbor of the query point, and n is the number of natural neighbors of $\mathbf{q}$. The intersection of the new Voronoi polygon with the original Voronoi diagram is defined by

$$\mathbf{V_i(q)} = \mathbf{V(q)} \cap \mathbf{V_i}, \tag{23}$$

where $\mathbf{V_i}$ is the Voronoi polygon of the i-th natural neighbor $\mathbf{x_i}$. It is said that the new Voronoi polygon $\mathbf{V(q)}$ "steals" some area from the neighboring Voronoi tiles $\mathbf{V_i}$ of the query point. Then, the ratio of the area of each intersection $\mathbf{V_i(q)}$ by the total area of the polygon $\mathbf{V(q)}$ constructs the natural neighbor interpolant of the query point $\mathbf{q}$ and is defined by

$$\widehat{f}(\mathbf{q}) = \sum_i w_i(\mathbf{q}) f(\mathbf{x_i}), \tag{24}$$

where

$$w_i(\mathbf{q}) = \frac{Area[\mathbf{V_i(q)}]}{Area[\mathbf{V(q)}]} \tag{25}$$

and $f(\mathbf{x_i})$ is the function value in the i-th natural neighbor point. In our case, it is the luminance value Y of the neighboring pixel.

The ratio of the areas $w_i(\mathbf{q})$ is the weight that measures the "influence" of each natural neighbor in order to interpolate the point $\mathbf{q}$. Therefore, it is said that the natural neighbor interpolation is a weighted average of the set of points in $\mathbf{S}$ that are the natural neighbors of the query point. Eq. (25) shows that $w_i(\mathbf{q})$ is normalized, therefore

$$0 \leq w_i(\mathbf{q}) \leq 1, \ \sum_i w_i = 1.$$

Note that the area ratio varies between 0 and 1, where 0 means that the area of the Voronoi cell of the given point does not intersect the area of the query point, and 1 means that the query point coincides exactly with the original point.

The function $\widehat{f}(\mathbf{q})$ defines a surface that is continuous and differentiable, except in data points $\mathbf{x_i}$ [28]. Then, the interpolator in (24) is of C^0 type, which means that its derivatives are discontinuous at points $\mathbf{x_i}$. There is a method to obtain the gradient of the function $\widehat{f}(\mathbf{q})$ at points $\mathbf{x_i}$ and construct a natural neighbor of C^1 type interpolant [24]. However, for the purposes of this work, the C^0 type interpolant will be used since an image is discrete and only the points that match with the HR grid will be interpolated.

By observing Fig. 7 it would be obvious that in order to compute the natural neighbor interpolant of each query point, it is necessary to construct a Voronoi diagram of the original point set $\mathbf{S}$ and then build a new Voronoi diagram for each point to be interpolated. It is noteworthy here that in the fastest algorithms used to compute a Voronoi diagram usually a Delaunay triangulation is computed first. Also, to find each weight w_i in (25), it is required to calculate the areas of the convex polygons that are stolen from the original Voronoi diagram, and since it is easier to find the area of a triangle than the area of an irregular polygon, a typical method is to divide the polygon into triangles and perform the summation of the areas of all triangles, which means to compute more triangulations. Therefore, it is computationally costly to use the above method to interpolate the HR image where it is needed to interpolate each pixel that matches with the HR grid.

The compound signed decomposition technique for natural neighbor interpolation proposed by Watson [27] is an algorithm that calculates the area ratio w_i computing only the original Delaunay triangulation from the original set of points. This algorithm is based on the Bowyer-Watson algorithm and performs local triangulations for each query point instead of computing a whole new triangulation, which saves computation time. Each pixel mapped in the HR grid represents a vertex of a Delaunay triangle; using this characteristic, it is possible to perform local triangulations. The fundamental principle of this algorithm is to construct a super-triangle that contains the polygon whose area will be calculated. Each edge of the super-triangle contains some edge of the polygon. The portions of the super-triangle that are outside of the polygon are also triangles. Therefore, the final area of the polygon is the sum of the "positive" area of the super-triangle and the "negative" areas of the triangles outside the polygon. This is the concept of signed triangulation, because the areas of some triangles are considered positive and some other areas are negative.

The method for obtaining the local coordinates w_i using the compound single decomposition is explained in detail in [29] and is summarized below:

Algorithm 2 Calculate natural neighbor coordinates $\mathbf{w_i}$

```
Construct a Delaunay triangulation $Dt$ from original point set $\mathbf{S}$ using Bowler-Watson algorithm [25].
**for** each Delaunay triangle $t$ in $Dt$ whose circumcircle contains $\mathbf{q}$ **do**
    Let $\mathbf{cc_t}$ be the circumcenter of $t$
    **for** each edge $i$ of $t$ with vertices $\mathbf{j}$ and $\mathbf{k}$ **do**
        $\mathbf{cc_i} := circumcenter(\mathbf{q}, \mathbf{j}, \mathbf{k})$
    **end for**
    **for** each vertex $i$ of $t$ **do**
        $w_i := w_i + 0.5 \cdot DET(\mathbf{cc_j}, \mathbf{cc_k}, \mathbf{cc_t})$
    **end for**
**end for**
Normalize all $w_i$ in order to obtain the local coordinates of $\mathbf{q}$.
```

The sign of the areas of triangles is given by the sequence of its vertices in the determinant. The order of the sequence is given by the table shown on page 9 in [29].

3.2 Bilinear Interpolation in Chrominance Color Components

Since scattered data interpolation is complex, it is preferable to use a simple interpolation in the chrominance color components Cr and Cb of the image. This is because only chrominance components contain color information. This method saves computation time and the resulting HR image quality is not affected significantly. Bilinear interpolation is a common and simple interpolation technique in the image processing field and is mostly used to resize digital images. In order to increase the spatial resolution of chrominance components, the pixel values between the HR grid must be computed. An efficient algorithm of bilinear interpolation applied to images is explained in [20].

4 Experiments and Results

The Natural Neighbor Super-Resolution (NNSR) algorithm was coded in the C programming language. The algorithm was divided into two stages as follows.

In the image registration stage the input LR images are converted from the *RGB* color space to the textitYCrCb color space. The *Y* luminance component of each image is used in the computation of the motion vectors. The ME was performed by implementing the algorithm proposed by Keren, Peleg, and Brada [14] following the steps in Algorithm 1 shown in Section 2. The FFTs were computed using the *FFTW* open-source library [10]. Then, pixels of the luminance component of each input LR image were mapped (using the computed motion vectors) into an HR grid whose size is defined by the up-scale factor F.

In the HR color image interpolation stage, the luminance pixel values of the HR image are interpolated using the natural neighbor interpolation and the *nn-c* open-source library that implements a hardware assisted algorithm based in compound signed decomposition that takes advantage of the capacity of modern video cards [8]. The chrominance (*Cr* and *Cb*) pixel values of the HR image are interpolated using the bilinear interpolation. Finally, the interpolated *Y*, *Cr*, and *Cb* components are converted to the *RGB* color system.

Standard LR test images were created from the HR images using the Matlab GUI developed in the Reproducible Research project [26]. This program is a graphical user interface that implements several motion-estimation and super-resolution algorithms. The NNSR algorithm was compared with three combinations of popular image registration and super-resolution algorithms implemented in the Super-Resolution Matlab GUI mentioned above.

In each experiment a different test image was used. Each HR image obtained from the NNSR algorithm and the combination of algorithms mentioned above was compared with the original HR image from which LR images were obtained. The comparisons were performed using the normalized cross-correlation (NCC) coefficient as a quality measure. The NCC coefficient was computed over each *RGB* color channel of the compared images and the coefficients were averaged.

The Normalized Cross Correlation (NCC) is a commonly used metric to evaluate the degree of similarity between two compared images. The main advantage of NCC, which makes it different from the other image comparison measures, is that it is less sensitive to linear changes in the amplitude of illumination in two compared images. The maximum absolute value of NCC coefficient is one which indicates perfect matching. Therefore, the closer the NCC coefficient is to one, the closer the reconstructed HR image is to the original HR image.

Let $\mathbf{X}$ and $\mathbf{Y}$ be the images to be compared [5], then

$$NCC = \frac{1}{N}\frac{\sum_{i=1}^{N}(X_i-\mu_X)(Y_i-\mu_Y)}{\sqrt{\sum_{i=1}^{N}(X_i-\mu_X)^2}\sqrt{\sum_{j=1}^{N}(Y_j-\mu_Y)^2}},$$

where μ_X and μ_Y are the means of images $\mathbf{X}$ and $\mathbf{Y}$, respectively, and N is the number of pixels assuming that the images are of the same size; therefore, summations are performed over all pixels of both images. The NCC coefficient has been used recently as an image quality measure for super-resolution algorithms [5, 11].

In the first experiment, the output HR images obtained from the proposed NNSR algorithm are compared with the HR images reconstructed from the combination of the ME algorithm of Marcel [16] and the Robust Super-Resolution (RS) algorithm [31]. Lenna 512×512 standard test image was used as the original HR image (Fig.8a). The HR image was sub-sampled by a factor of 4 in each dimension and was used as the reference image (Fig.8b). The reference LR image was shifted in order to produce eight Lenna 128×128 LR images with the shifts and rotations shown in table 1.

Table 1. Image motions of Lenna LR images

	Rotation	x shift	y shift
Image 1	0.00	0.00	0.00
Image 2	0.00	0.25	0.25
Image 3	0.00	0.5	0.50
Image 4	0.00	-0.25	-0.25
Image 5	0.00	0.15	0.15
Image 6	0.00	0.35	0.35
Image 7	0.00	-0.15	-0.15
Image 8	0.00	-0.35	-0.35

Table 2. NCC coefficients from the comparison with the Lenna HR image

	4 images		8 images	
	NNSR	RS	NNSR	RS
Red NCC	0.998047	0.977294	0.998142	0.975516
Green NCC	0.997776	0.970706	0.997891	0.968936
Blue NCC	0.993972	0.954812	0.994048	0.953366
Avg. NCC	0.9966	0.96765	0.996695	0.965984

Four HR images were obtained in this experiment by implementing the algorithms considered previously with an up-scale factor of 4 and using the LR images. The first two images were reconstructed by implementing the proposed NNSR algorithm. The image in Fig.9a was reconstructed using the LR images from 1 to 4. The image in Fig.10a is the result of using the eight images. The high-resolution images of Fig.9b and Fig. 10b were reconstructed with the RS algorithm using four and eight images, respectively. Each obtained HR image was compared with the original HR image of Fig. 8a by computing the NCC coefficient on each *RGB* color component. The results are shown in Table 2.

In the second experiment, the HR images produced by the NNSR algorithm and the combination

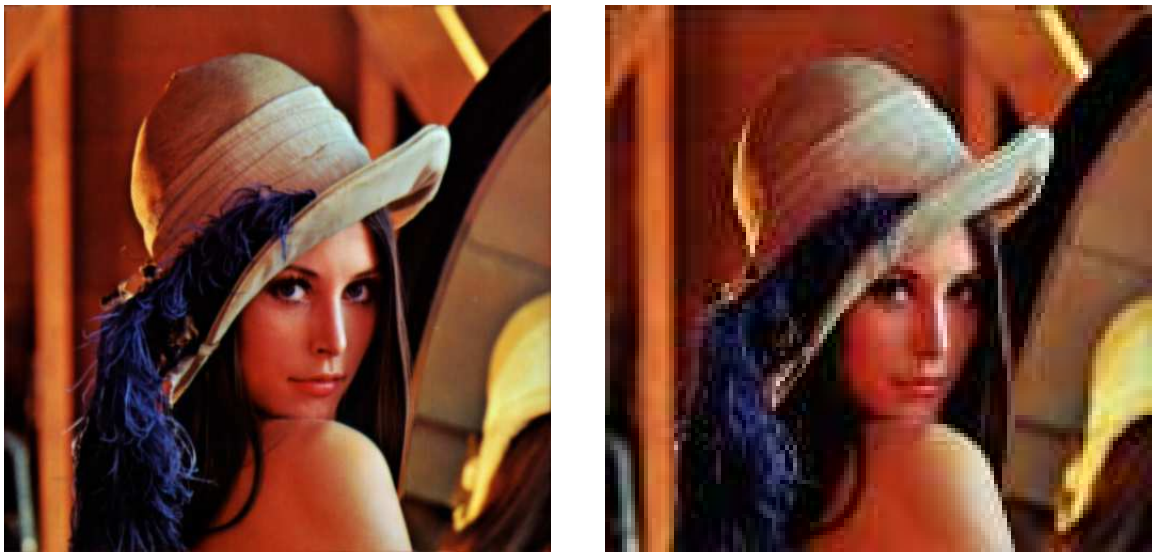


Fig. 8. (a) Lenna original HR image, (b) Lenna reference LR image

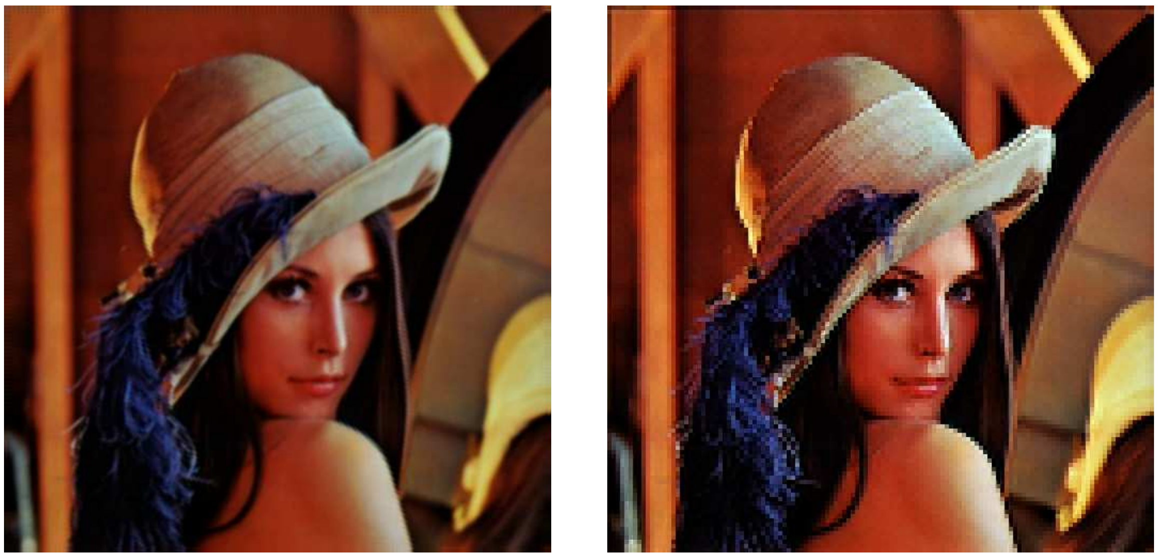
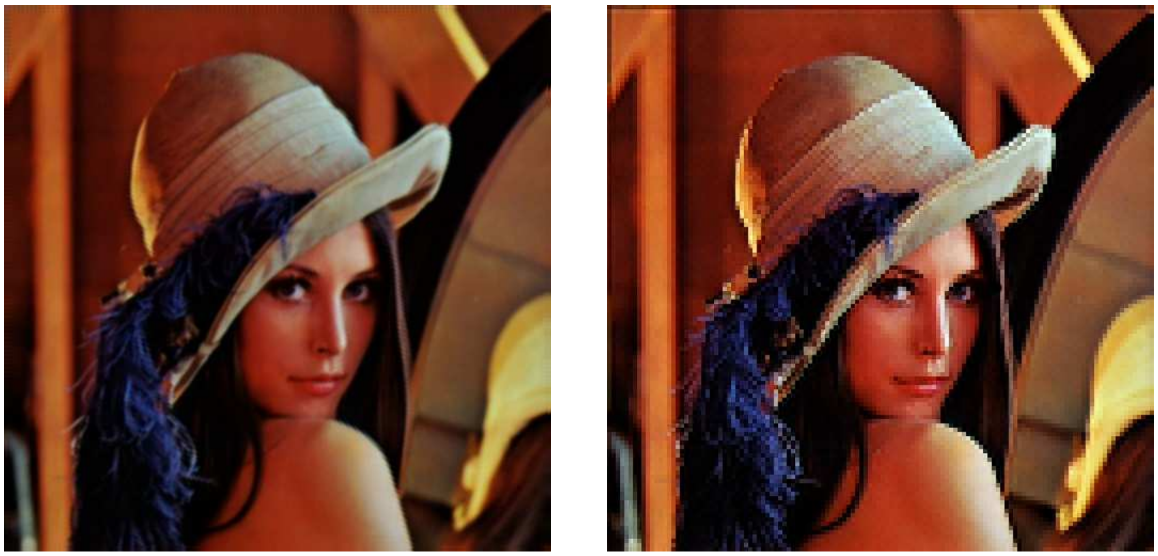


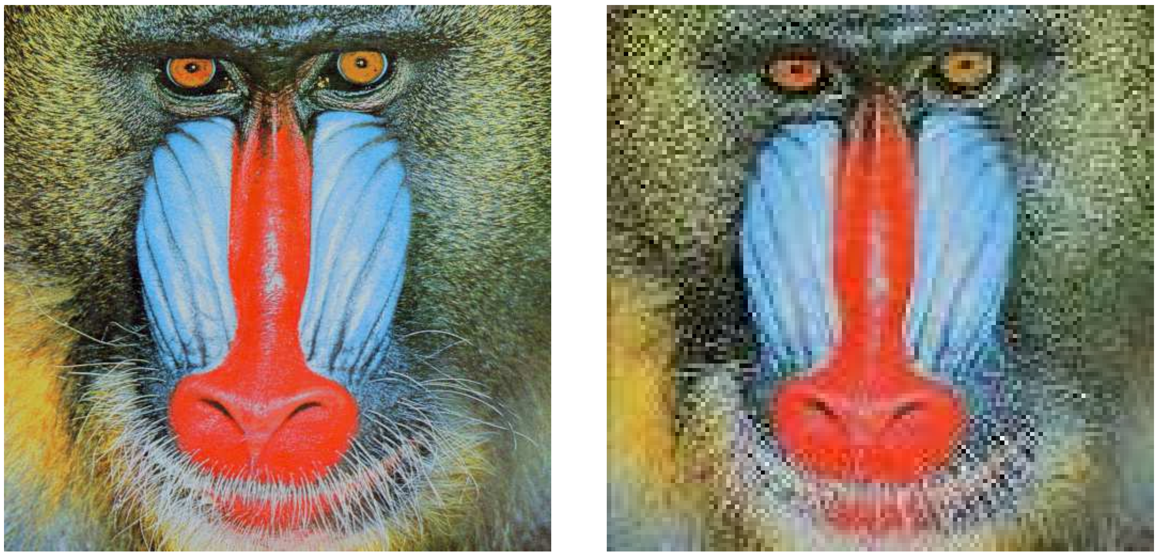
Fig. 9. (a) Lenna HR image reconstructed by the NNSR algorithm (4 images), (b) Lenna HR image reconstructed by the RS algorithm (4 images)

of the ME algorithm [16] and the Iterated Back-Projection (IBP) algorithm for image reconstruction [13] were compared. Mandrill 512×512 standard test image was used as the original HR image (Fig. 11a). Ten Mandrill 128×128 LR images were created from the HR image by sub-sampling the



(a) (b)

Fig. 10. (a) Lenna HR image reconstructed by the NNSR algorithm (8 images), (b) Lenna HR image reconstructed by the RS algorithm (8 images)



(a) (b)

Fig. 11. (a) Mandrill original HR image, (b) Mandrill reference LR image

original image by a factor of 4 and adding the shifts and rotations of Table 3 where the first image is the reference image.

An up-scale factor of 4 was used in the reconstruction of the HR images. The result of implementing the NNSR algorithm to the first four

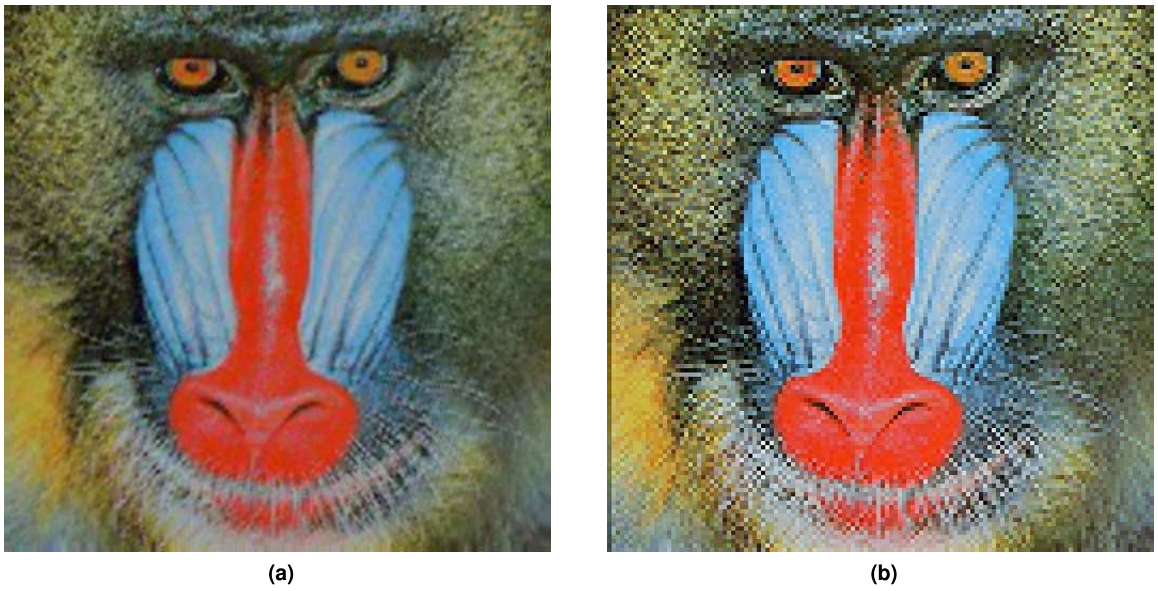


Fig. 12. (a) Mandrill HR image reconstructed by the NNSR algorithm (4 images), (b) Mandrill HR image reconstructed by the IBP algorithm (4 images)

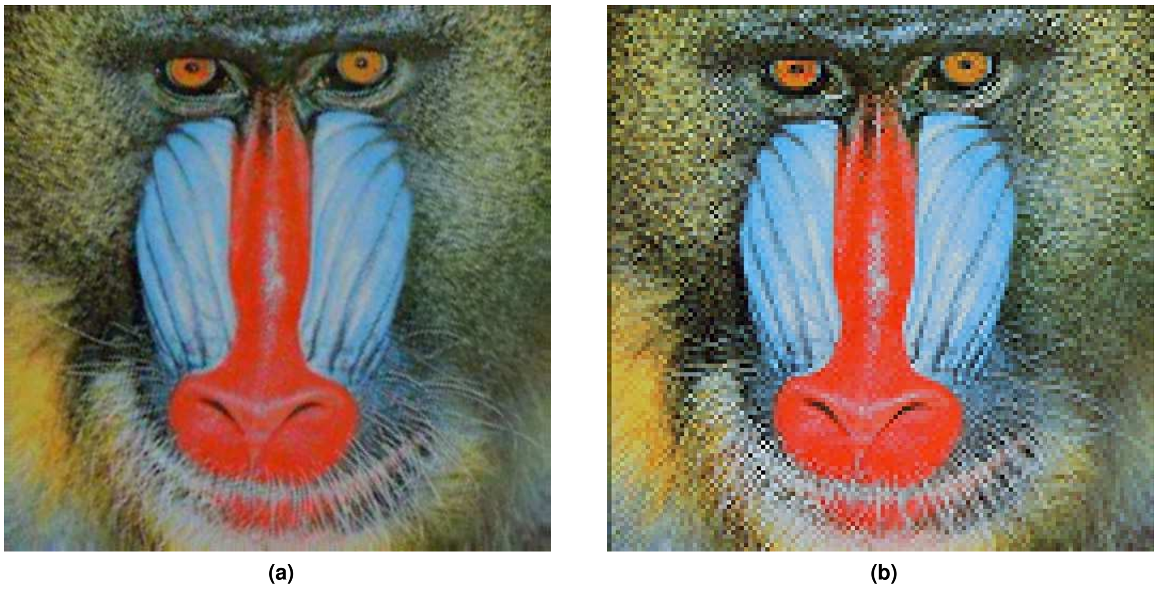


Fig. 13. (a) Mandrill HR image reconstructed by the NNSR algorithm (10 images), (b) Mandrill HR image reconstructed by the IBP algorithm (10 images)

images is shown in Fig. 12a. The resulting HR image using ten images is shown in Fig. 13a. Also, two HR images were reconstructed using the IBP algorithm. Fig. 12b shows the image obtained from

Table 3. Image motions of Mandrill LR images

	Rotation	x shift	y shift
Image 1	0.00	0.10	-0.10
Image 2	0.00	0.15	-0.15
Image 3	0.00	0.20	-0.20
Image 4	0.00	0.25	-0.25
Image 5	0.01	0.30	-0.30
Image 6	-0.01	-0.10	0.10
Image 7	0.02	-0.15	0.15
Image 8	-0.02	-0.20	0.20
Image 9	0.03	-0.25	0.25
Image 10	-0.03	-0.30	0.30

Table 4. NCC coefficients from the comparison with the Mandrill HR image

	4 images		10 images	
	NNSR	IBP	NNSR	IBP
Red. NCC	0.912466	0.802046	0.923084	0.83115
Green NCC	0.861507	0.683716	0.884554	0.721808
Blue NCC	0.900955	0.790601	0.912872	0.820018
Avg. NCC	0.891909	0.760657	0.906983	0.792516

four images, and Fig. 13b shows the reconstructed image from ten images. Each resulting HR image from the NNSR and IBP algorithms was compared with the original HR image of Fig. 11a by computing the NCC coefficient on each *RGB* color component. The results are shown in Table 4.

In the third experiment the NNSR algorithm is compared against the Normalized Convolution (NC) algorithm for image reconstruction [18], using the ME technique [14] discussed in Section 2. A 256×256 Jelly Beans image is used as the original HR image. Eight 64×64 Jelly Beans LR images were created from the original HR image by subsampling with a factor of 4 and adding the sub-pixel shifts and rotations of Table 5, where the first image is the reference image. Two HR images were constructed using the NNSR and NC algorithms and all the eight LR images were made with an up-scale factor of 4. Both HR images resulting from the NNSR and NC algorithms were compared with the original HR image computing the NCC coefficient over each *RGB* color component and the results are shown in Table 6. For space purposes, input and output images in this experiment are not shown in this work.

Table 5. Image motions of Jelly Beans LR images

	Rotation	x shift	y shift
Image 1	0.00	0.00	0.00
Image 2	0.00	0.40	-0.20
Image 3	0.017453289	0.50	0.60
Image 4	-0.017453289	0.35	-0.25
Image 5	0.026179933	-0.05	0.05
Image 6	-0.026179933	-0.10	0.30
Image 7	0.052359867	-0.15	-0.15
Image 8	-0.052359867	-0.1	0.20

Table 6. NCC coefficients from the comparison with the Jelly Beans HR image

	NNSR (8 images)	NC (8 images)
Red NCC	0.980665	0.915898
Green NCC	0.988527	0.961583
Blue NCC	0.98999	0.956812
Avg. NCC	0.986403	0.944987

Finally, a practical example is included in order to show the utility of the NNSR algorithm. The four LR images from Fig. 14a to Fig. 14d were taken from [19], the images were cropped to show the license plate number. The HR image in Fig. 15 was generated with the NNSR algorithm using the four LR images with an up-scale factor of 4.

5 Conclusions

In this paper we proposed a new super-resolution image reconstruction using natural neighbor interpolation. The main contribution of this work is the implementation of the area based method of natural neighbor interpolation in super-resolution image reconstruction where LR images are used in order to reconstruct an HR image. Also an optimized image registration method is introduced solving the SLE by computing the Fast Fourier Transform of images. The HR images constructed by the NNSR algorithm were compared against the images obtained from three popular super-resolution image reconstruction algorithms proposed in literature. The results of experiments show that the proposed NNSR algorithm has a better performance in terms

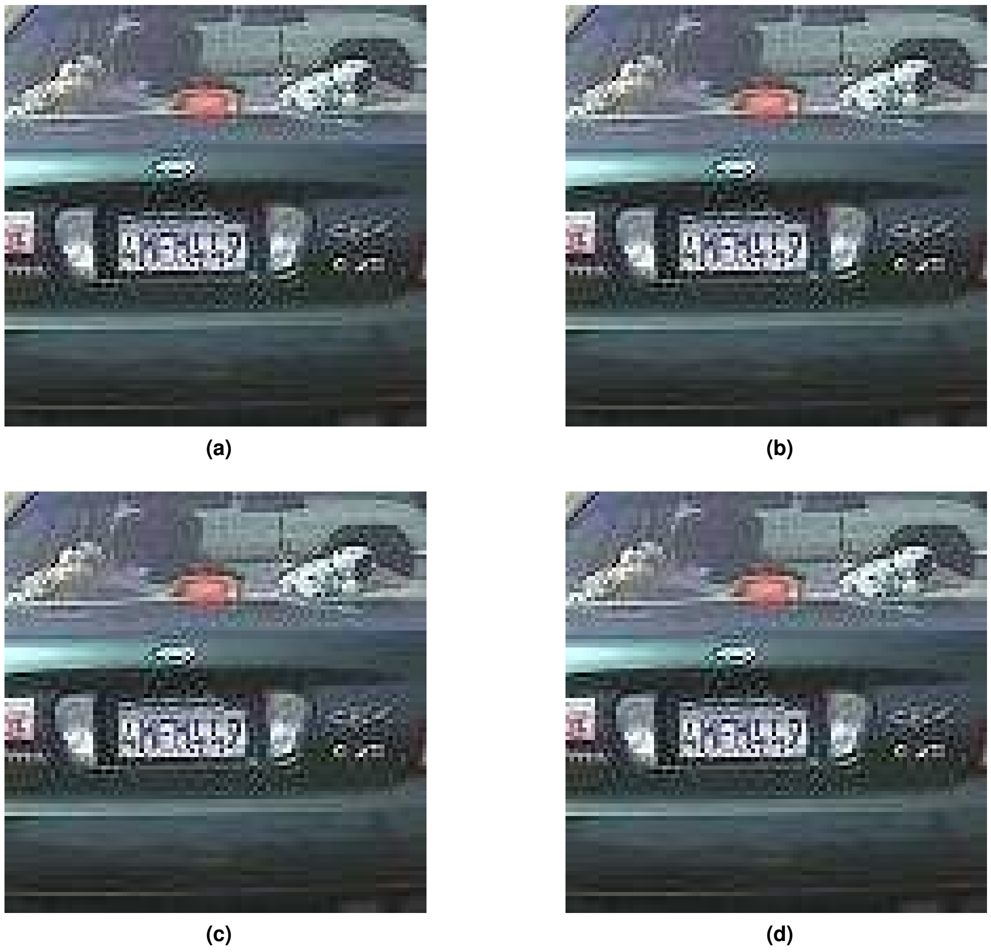


Fig. 14. Four LR images of a license plate

of quality of the resulting images than the algorithms it was compared against.

The results of the comparison against the RS algorithm demonstrate visually (Fig. 9a and Fig. 10a against Fig. 9b and Fig. 10b) and statistically using the NCC coefficient (Table 2), that the HR image obtained from the NNSR algorithm is superior in quality to the image produced by the RS algorithm when the input LR images have very small translational displacements and null rotations.

The results obtained from the comparison against the IBP algorithm show that the output NNSR image has a significantly better quality than that of the mentioned algorithm (from Fig. 12a to Fig. 13a and Table 4).

In the last experiment, the NNSR algorithm is compared against the normalized convolution method. It is shown (Table 6) that the NNSR algorithm output image has a slightly better statistical quality than that of the NC algorithm.

It is noteworthy that in the first experiment, the Lenna image quality is not improved significantly when eight images are used as input instead of four. However, in the second experiment the Mandrill image quality increases considerably when ten images are used instead of four images. This is because the Mandrill image has several well defined

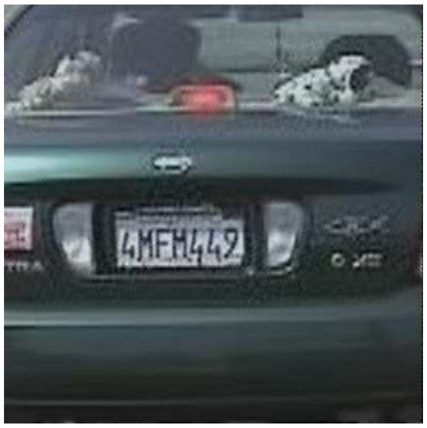


Fig. 15. License plate HR image reconstructed by the NNSR algorithm

borders and edges, so the area based interpolation has a better performance for this kind of images.

Finally, by visual inspection, the reconstructed HR image of the license plate in Fig. 15 has a significant resolution improvement compared with the LR images from Fig. 14a to Fig. 14d. The NNSR algorithm could be used in an image pre-processing stage for an automatic number plate recognition algorithm for better results in recognition.

Appendix

It is necessary to define the spatial relationship between the reference image and the warped image in order to obtain the SLE. Displacements and rotations are the basic geometric operations to change the size and shape of objects in an image. Usually, a left-handed Cartesian coordinate system is used in image processing, with the origin located in the upper left corner of the image. Fig. 16 shows the basic idea of the rotation operation in an image.

Looking at the picture and adding horizontal and vertical displacements, we derive the following equations:

$$x' = x \cdot \cos(\theta) - y \cdot \sin(\theta) + a, \tag{26}$$

$$y' = y \cdot \cos(\theta) + x \cdot \sin(\theta) + b. \tag{27}$$

The first two terms in each equation perform the rotation in any angle θ, whereas the terms a and b perform shifts in the horizontal and vertical directions. Equations (26) and (27) perform the rotation operation, but only about the origin located in the upper left corner in the image. Another kind of rotation operation allows any point (x_o, y_o) in the image to be the center of rotation. The operations of rotation about any point and displacements can be combined to generate operations given in (1) and (2). Once the basic geometric operations are defined, the relationship between the reference image and a single shifted rotated image can be defined: the reference image is set as a function $f(x, y)$, and the warped image is set as a function $g(x, y)$. The following relation is established using (1) and (2) for the horizontal shift a, the vertical shift b, and the rotation angle θ about the origin (x_o, y_o):

$$\begin{aligned} g(x,y) = f(x',y') = f\big((x-x_o)\cos(\theta) \\ - (y-y_o)\sin(\theta) \\ + a, (y-y_o)\cos(\theta) + (x-x_o)\sin(\theta) + b\big). \end{aligned} \tag{28}$$

If $\cos(\theta)$ is expanded to its first two terms and $\sin(\theta)$ is expanded to its first term in their Taylor series

$$\cos(\theta) \approx 1 - \frac{1}{2}\theta^2, \ \sin(\theta) \approx \theta, \tag{29}$$

then, substituting (29) in (28) and defining new coordinates $x'' = (x - x_o)$ and $y'' = (y - y_o)$, we get

$$g(x,y) \approx f\Bigg(x'' + a - y''\theta - x''\frac{\theta^2}{2} \ ,$$

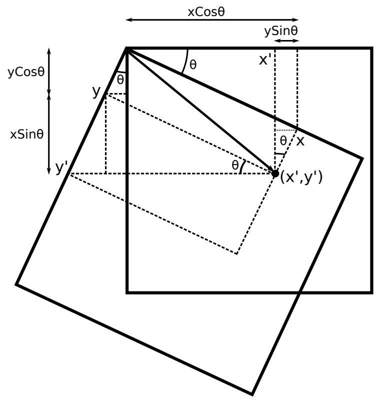


Fig. 16. Image rotation about origin

$$y'' + b + x''\theta - y''\frac{\theta^2}{2}\Bigg) . \qquad (30)$$

Expanding the function $g(x, y)$ to the first term of its own Taylor series around the point (x'', y''), we obtain

$$\begin{aligned} f(x', y') \approx f(x'', y'') + (x' - x'')f(x'', y'')_x \\ + (y' - y'')f(x'', y'')_y. \end{aligned} \qquad (31)$$

According to (30) we have

$$x' \approx x'' + a - y''\theta - x''\frac{\theta^2}{2},\ y' \approx y'' + b + x''\theta - y''\frac{\theta^2}{2}.$$

Therefore, from (31) and (28), and since $x'' = (x - x_o)$ and $y'' = (y - y_o)$, finally, we get

$$\begin{aligned} g(x, y) &\approx f(x, y) \\ &+ \left(a - (y - y_o)\theta - (x - x_o)\frac{\theta^2}{2}\right) f(x, y)_x \\ &+ \left(b + (x - x_o)\theta - (y - y_o)\frac{\theta^2}{2}\right) f(x, y)_y. \end{aligned} \qquad (32)$$

The mean square error (MSE) function (with variables a, b, and θ) of the approximation of function $g(x, y)$ in (32) is

$$\begin{aligned} MSE \approx \sum_{x,y} \Bigg[f + \left(a - (y - y_0)\theta - (x - x_o)\frac{\theta^2}{2}\right) f_x \\ + \left(b + (x - x_o)\theta - (y - y_o)\frac{\theta^2}{2}\right) f_y \\ - g\Bigg]^2 \end{aligned}$$

where the functions $f(x, y) = f$ and $g(x, y) = g$ were redefined to simplify notation. The summation is over the overlapping part of f and g. For small angles (less than 1 radian or $180/\pi$ degrees), the nonlinear terms θ^2 are very small so they can be neglected in the error function. Then

$$\begin{aligned} MSE \approx \sum_{x,y} \Big[f + (a - (y - y_0)\theta)\, f_x \\ + (b + (x - x_o)\theta)\, f_y - g\Big]^2. \end{aligned} \qquad (33)$$

The next step is to look for the minimum of (33) by computing its partial derivatives with respect to a, b, and θ and equating to zero. To simplify notation, a function U with variables a, b, and θ is defined as

$$U = [f + (a - (y - y_0)\theta)\, f_x + (b + (x - x_o)\theta)\, f_y - g] . \qquad (34)$$

The partial derivatives of the error function with respect to the horizontal shift a, the vertical shift b, and the rotation θ are

$$\begin{aligned} MSE_a &= 2\sum_{x,y} U \cdot U_a, \\ MSE_b &= 2\sum_{x,y} U \cdot U_b, \\ MSE_\theta &= 2\sum_{x,y} U \cdot U_\theta \end{aligned} \qquad (35)$$

where

$$\begin{aligned} U_a &= f_a + f_x \cdot (a - (y - y_0)\,\theta)_a \\ &\quad + [f_y \cdot (b + (x - x_o)\theta)]_a - g_a = f_x, \\ U_b &= f_b + [f_x \cdot (a - (y - y_0)\theta)]_b \\ &\quad + f_y \cdot (b + (x - x_o)\theta)_b - g_b = f_y, \\ U_\theta &= f_\theta + f_x \cdot (a - (y - y_0)\theta)_\theta + f_y \cdot (b + (x - x_o)\theta)_\theta \\ &\quad - g_\theta = f_y \cdot (x - x_0) - f_x \cdot (y - y_o). \end{aligned} \qquad (36)$$

Substituting (36) in (35), we obtain

$$\begin{aligned} MSE_a &= 2\sum_{x,y} U \cdot f_x, \\ MSE_b &= 2\sum_{x,y} U \cdot f_y, \\ MSE_\theta &= 2\sum_{x,y} U \cdot [f_y \cdot (x - x_0) - f_x \cdot (y - y_o)] . \end{aligned} \qquad (37)$$

Substituting (34) in (37) and equating partial derivatives to zero, we get

$$\begin{aligned}
0 = \sum_{x,y} & [f + (a - (y - y_0)\theta)\, f_x \\
& + (b + (x - x_o)\theta)\, f_y - g]\, f_x, \\
0 = \sum_{x,y} & [f + (a - (y - y_0)\theta)\, f_x \\
& + (b + (x - x_o)\theta)\, f_y - g]\, f_y, \\
0 = \sum_{x,y} & [f + (a - (y - y_0)\theta)\, f_x \\
& + (b + (x - x_o)\theta)\, f_y - g] \\
& \cdot [f_y \cdot (x - x_0) - f_x \cdot (y - y_o)]\,.
\end{aligned}$$

Then, substituting R as given in (4) and doing some algebra we obtain the SLE given in (3). The equations in (3) are the final SLE, with displacements a, b, and rotation angle θ as unknowns.

It is noteworthy that the SLE is only valid for small values of the angle θ due to the Taylor series approximations. The SLE derived in this work is slightly different from the SLE shown in [12–14].

References

1. **Alfeld, P.** (**1989**). Scattered data interpolation in three or more variables. In **Lyche, T. & Schumaker, L. L.**, editors, *Mathematical methods in computer aided geometric design*. Academic Press Professional, Inc., San Diego, CA, USA, pp. 1–33.
2. **Amidror, I.** (**2002**). Scattered data interpolation methods for electronic imaging systems: a survey. *Journal of Electronic Imaging*, Vol. 11, No. 2, pp. 157–176.
3. **Anton, F., Mioc, D., & Fournier, A.** (**2001**). Reconstructing 2d images with natural neighbour interpolation. *The Visual Computer*, Vol. 17, No. 3, pp. 134–146.
4. **Barnea, D. I. & Silverman, H. F.** (**1972**). A class of algorithms for fast digital image registration. *IEEE Transactions on Computers*, Vol. C-21, No. 2, pp. 179–186.
5. **Begin, I. & Ferrie, F. P.** (**2006**). Comparison of super-resolution algorithms using image quality measures. *Proceedings of the The 3rd Canadian Conference on Computer and Robot Vision*, IEEE Computer Society, Los Alamitos, CA, USA.
6. **Bracewell, R.** (**2000**). The basic theorems. In *The Fourier Transform and its Applications, 3th edition*, chapter 6. McGraw-Hill, New York, NY, USA, pp. 105–135.
7. **Brown, L. G.** (**1992**). A survey of image registration techniques. *ACM Computing Surveys*, Vol. 24, No. 4, pp. 325–376.
8. **Fan, Q., Efrat, A., Koltun, V., Krishnan, S., & Venkatasubramanian, S.** (**2005**). Hardware-assisted natural neighbor interpolation. *Proc. 7th Workshop on Algorithm Eng. and Experiments (ALENEX)*, Vancouver, BC, Canada, pp. 111–120.
9. **Farin, G.** (**1990**). Surfaces over dirichlet tessellations. *Comput. Aided Geom. Des.*, Vol. 7, No. 1-4, pp. 281–292.
10. **Frigo, M. & Johnson, S. G.** (**2005**). The design and implementation of FFTW3. *Proceedings of the IEEE*, Vol. 93, No. 2, pp. 216–231. Special issue on “Program Generation, Optimization, and Platform Adaptation”.
11. **Hsieh, C.-C., Huang, Y.-P., Chen, Y.-Y., Fuh, C.-S., & Ho, W.-J.** (**2008**). Video super-resolution by integrating sad and ncc matching criterion for multiple moving objects. *Proceedings of the Tenth IASTED International Conference on Computer Graphics and Imaging*, CGIM ’08, ACTA Press, Anaheim, CA, USA, pp. 172–177.
12. **Irani, M. & Peleg, S.** (**1990**). Super resolution from image sequences. *Proceedings of 10th International Conference on Pattern Recognition (ICPR)*, volume 2, Atlantic City, NJ, USA, pp. 115–120.
13. **Irani, M. & Peleg, S.** (**1991**). Improving resolution by image registration. *CVGIP: Graphical Models and Image Processing*, Vol. 53, No. 3, pp. 231–239.
14. **Keren, D., Peleg, S., & Brada, R.** (**1988**). Image sequence enhancement using sub-pixel displacements. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, Ann Arbor, MI, USA, pp. 742–746.
15. **Lertrattanapanich, S. & Bose, N.** (**2002**). High resolution image formation from low resolution frames using delaunay triangulation. *IEEE Transactions on Image Processing*, Vol. 11, No. 12, pp. 1427–1441.
16. **Marcel, B., Briot, M., & Murrieta, R.** (**1997**). Calcul de translation et rotation par la transformation de fourier. *TS. Traitement du signal*, Vol. 14, No. 2, pp. 135–149.
17. **Park, S. C., Park, M. K., & Kang, M. G.** (**2003**). Super-resolution image reconstruction: a techni-

cal overview. *IEEE Signal Processing Magazine*, Vol. 20, No. 3, pp. 21–36.

18. **Pham, T. Q., van Vliet, L. J., & Schutte, K.** (**2006**). Robust fusion of irregularly sampled data using adaptive normalized convolution. *EURASIP J. Appl. Signal Process.*, Vol. 2006, pp. 236–236.

19. **Philip, B. & Updike, P.** (**2005**). GCar dataset images taken for Caltech SURF project.

20. **Phillips, D.** (**2000**). Geometric operations. In *Image Processing in C, Electronic, 2nd edition*, chapter 13. Lawrence, Kansas: R & D Publications, pp. 197–208.

21. **Piper, B.** (**1993**). Geometric modelling. In **Farin, G., Hagen, H., Noltemeier, H., & Knödel, W.**, editors, *Properties of local coordinates based on Dirichlet tessellations*. Springer-Verlag, London, UK, UK, pp. 227–239.

22. **Reddy, B. & Chatterji, B.** (**1996**). An fft-based technique for translation, rotation, and scale-invariant image registration. *IEEE Transactions on Image Processing*, Vol. 5, No. 8, pp. 1266–1271.

23. **Sibson, R.** (**1980**). A vector identity for the Dirichlet tessellation. *Mathematical Proceedings of the Cambridge Philosophical Society*, Vol. 87, pp. 151–155.

24. **Sibson, R.** (**1981**). A brief description of natural neighbour interpolation. In **Barnett, V.**, editor, *Interpreting multivariate data*, chapter 2. John Wiley & Sons, Chichester, UK, pp. 21–36.

25. **Sloan, S. & Houlsby, G.** (**1984**). An implementation of Watson's algorithm for computing 2-dimensional delaunay triangulations. *Advances in Engineering Software*, Vol. 6, No. 4, pp. 192–197.

26. **Vandewalle, P., Kovacevic, J., & Vetterli, M.** (**2009**). Reproducible research in signal processing. *IEEE Signal Processing Magazine*, Vol. 26, No. 3, pp. 37–47.

27. **Watson, D. F.** (**1981**). Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes. *The Computer Journal*, Vol. 24, No. 2, pp. 167–172.

28. **Watson, D. F.** (**1992**). *Contouring: A Guide to the Analysis and Display of Spatial Data*, volume 10. Pergamon Press, New York, USA.

29. **Watson, D. F.** (**2002**). Compound signed decomposition, the core of natural neighbour interpolation in n-dimensional space. *(unpublished manuscript)*, pp. 1–15.

30. **Yongchang, C. & Hehua, Z.** (**2004**). A meshless local natural neighbour interpolation method for stress analysis of solids. *Engineering Analysis with Boundary Elements*, Vol. 28, No. 6, pp. 607–613.

31. **Zomet, A., Rav-Acha, A., & Peleg, S.** (**2001**). Robust super-resolution. *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pp. I645–I650.

Christian J. Enríquez-Cervantes received the B.Sc. in Electronics Engineering from the Instituto Tecnológico de Chihuahua, Chihuahua, México, in 2007, and the M.Sc. degree in Electronics Engineering (Electronics Systems) from the Tecnológico de Monterrey (ITESM), Monterrey, México, in 2012. He currently works as a Support Professional at the Electrical and Computing Engineering Department of ITESM. From 2007 to 2009, he worked at Safran Engineering Services as a Design Engineer in the Boeing 787 launch project. From January 2010 to 2012 he was a Research Assistant in the Three-Dimensional Video Transmission Research Chair at ITESM. His research interests include signal and image processing and embedded systems.

Ramón M. Rodríguez-Dagnino is a full Professor at Tecnológico de Monterrey (ITESM), Monterrey, México. He was the Director of the Telecommunications Management Master Program (2000–2009). He received his Ph. D. from the University of Toronto, 1993, and his M. Sc. from the Research and Advanced Studies Center (CInvEstAv) in México City, 1984. He worked at the R&D Center of TelMex (Mexican Telephone Co.) from 1984 to 1989. He was the Chair of the Electronics and Telecommunications Center at ITESM (2000–2001), and a member of the Academic University Council during the 2000–2001 academic years. He won the ITESM Best Teaching and Research Award twice, in 1998 and 2001. He is the Chairman of the IEEE-MTTS-17 Chapter in México. His research interests include video and image processing, performance analysis of telecommunication systems, and electromagnetics. He has served as a technical reviewer of IEEE journals and conferences, SPIE and Elsevier journals, and in the

Program Committee of SPIE conferences. He is a member of IEEE, AMS, the Mexican Academy of Sciences (AMC), and the Mexican National System of Researchers (SNI).

Article received on 27/10/2014; accepted on 21/04/2015.
Corresponding author is Ramón M. Rodríguez-Dagnino.