Image interpolation as a boundary value problem

Hei Tao Fung
Kevin J. Parker

Electrical Engineering Department
University of Rochester
Rochester, New York 14627

ABSTRACT

Image interpolation is the determination of unknown pixels based on some known pixels. The conventional interpolation methods such as pixel replication, bilinear interpolation, and cubic spline interpolation, assume that the known pixels are located regularly on a Cartesian mesh. They cannot be easily extended to other cases where the configurations of the known pixels are different. We propose a novel formulation of the image interpolation problem to deal with the more general cases, such as the case where a region of image is missing and the case where the known pixels are irregularly placed. The interpolation problem is formulated into a boundary value problem involving the Laplacian equation and the known pixels as the boundary conditions. The matrix equation resulting from the formulation has a unique solution. It can solved efficiently by the successive over-relaxation (SOR) iteration. The advantage of the proposed interpolation method lies in its flexibility in handling the general cases of interpolation.

Keywords: image interpolation, Laplacian equation, successive over-relaxation (SOR), and boundary value problem.

1. INTRODUCTION

A major application of interpolation in image processing is the enlargement of a digital image. There are well known techniques of different computational complexity, and they achieve different image quality. Examples are pixel replication, bilinear interpolation and cubic spline interpolation [1]. These interpolation techniques assume that the known pixels are located regularly on a Cartesian mesh. Though they may produce desirable results for conventional
cases, their applications are not simple extensions to some unconventional cases, e.g., the known pixels are located irregularly, or an image region (composed of multiple pixels) is missing.

We propose a novel approach to image interpolation, which is particularly suitable to the unconventional cases. We frame the interpolation problem as a boundary value problem. The known pixels, irrespective of their locations, are treated as boundary conditions. The smoothness or the variation of the interpolated pixels is constrained by the Laplacian equation. Although mathematical theories of differential equations are applied to image filtering and image restoration [2][3], their application to image interpolation has not been found. We shall present the theory of the boundary value problem in section 2. In section 3, we focus on the application of the boundary value problem to image interpolation. Simulation results are shown in section 4. Section 5 concludes our findings.

2. BOUNDARY VALUE PROBLEM

The theory of boundary value problem, especially that involves the Laplacian equation, has been widely published. One area of its applications is electrostatics [4]. We focus our discussions on the boundary value problem that involves the Laplacian equation; however, other differential equations can be used to model an image differently. The Laplacian is particularly appealing because of its mechanical analog, the stretched membrane, which can be intuitively understood [5]. In the following subsections, we shall discuss the formulation of the boundary value problem, solving the boundary value problem, and the successive over-relaxation iteration.

2.1 Boundary value problem involving the Laplacian equation

A boundary value problem (in two dimensions) can be cast in terms of a function, $\Phi(x,y)$, that satisfies the Laplacian equation:

$$
\left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) \Phi(x,y) = 0
$$

at every point $(x,y)$ except at the boundary. At the boundary, $\Phi(x,y)$ must be prescribed by boundary values. The solution of (1) is unique and always exists [5]. The solution also satisfies the min-max principle, which states that there is no local minimum nor local maximum in the solution region [5]. This implies that the global maximum and the global minimum occur only at the boundary.

FUNG & PARKER
This boundary value problem is generally difficult to solve. For complicated cases, explicit solutions are intractable. Numerical approximations to the solutions are often sought. An approximation to the differential equation (1) is given by the so-called five-point formula

\[4I_{m,n} - I_{m-1,n} - I_{m+1,n} - I_{m,n-1} - I_{m,n+1} = 0\]  

where \( I_{m,n} = \Phi(m\Delta,n\Delta) \). \( \Delta \) is the sampling step size, and \( m \) and \( n \) are the indices for the vertical and horizontal axes, respectively. The approximation defines a Cartesian grid over the solution region of (1), and the grid points are the only defined quantities. (2) applies to all unknown grid points, whose values are to be solved. The known grid points being 4-neighbors to the unknown grid points specify the boundary conditions. Consequently, in (2), \( I_{m,n} \) must be an unknown grid point, and the rest of the terms can be an unknown or a known. The solution to (2) involves a matrix equation

\[Af = b\]

where \( A \) is a \( K \times K \) coefficient matrix, \( f \) is a \( K \times 1 \) vector, and \( b \) is also a \( K \times 1 \) vector. \( K \) is the total number of unknown grid points to be solved. The elements of vector \( f \) correspond to the unknown grid points specifically ordered. For simplicity, we use the lexicographic ordering (figure 1), where the grid points are numbered in a raster scan manner. A different ordering affects the distribution of non-zero coefficients in matrix \( A \), and hence, it may lead to different computational load in solving (3). Vector \( b \) is related to the boundary conditions. It can be proved that \( A \) is positive definite, regardless of the boundary shape. This property of \( A \) guarantees the existence of a unique solution [6]. \( A \) is also sparse when \( K \) is large because there are at most five non-zero coefficients in a row. The above two properties lead to simplification in computation of the solution.

![Diagram of lexicographic ordering of unknown grid points](image)

Figure 1. Lexicographic ordering of unknown grid points
There are various ways to solve (3). We recommend the use of successive over-relaxation iteration for its greater speed of convergence compared to other iterative methods and smaller memory requirement compared to the direct methods [7].

2.2 Successive over-relaxation iteration (SOR)

SOR updates the current element based on the last updated values of all elements. Applying SOR to solving (3) leads to the following:

\[
I_{m,n}^k = I_{m,n}^{k-1} - \omega \left( 4I_{m,n}^{k-1} - I_{m-1,n}^k - I_{m+1,n}^k - I_{m,n-1}^{k-1} - I_{m,n+1}^{k-1} \right) / 4 \quad \forall (m,n) \in U \tag{4}
\]

\(I_{m,n}^k\) denotes the \(I_{m,n}\) at the \(k\)th iteration, and \(U\) denotes the set that holds the coordinates of the unknown grid points. \(\omega\) is a real number, and \(1 < \omega < 2\). In (4), the unknown \(I_{m,n}^k\)'s are processed in the lexicographic order. (4) is applied iteratively until a termination condition is satisfied.

SOR converges for all cases where the coefficient matrix is symmetric positive definite provided that \(0 < \omega < 2\) [8, pp. 193]. As a result, convergence of (4) is guaranteed.

All unknown grid points must be bounded by some known grid points both inside and outside. To specify the values along the outermost boundary, alternatively, we may place a “mirror” along the boundary such that whenever an unknown grid point \(I_{m,n}\) along the outermost boundary is considered, the neighboring boundary grid point takes the value of \(I_{m,n}\) (figure 1). For example, if the region of support is rectangular, with \(m\) ranging from 0 to \(m_{\text{max}}\) and \(n\) ranging from 0 to \(n_{\text{max}}\), (4) is modified to

\[
I_{m,n}^k = I_{m,n}^{k-1} - \omega \left( 4I_{m,n}^{k-1} - I_{m,\text{MAX}(n-1,0)}^k - I_{m,\text{MAX}(n+1,0)}^k - I_{m,\text{MIN}(n+1,n_{\text{max}})}^{k-1} - I_{m,\text{MIN}(n-1,n_{\text{max}})}^{k-1} \right) / 4 \quad \forall (m,n) \in U \tag{5}
\]

where \(\text{MAX()}\) and \(\text{MIN()}\) return the maximum and the minimum of two numbers, respectively. Convergence is still guaranteed.

3. APPLICATION TO IMAGE INTERPOLATION

We have described the boundary value problem with Laplacian equation, its solution, and the method to solve it. We shall treat image interpolation as the boundary value problem. In the following discussions, we use the term "pixel" instead of "grid point." We propose the following image interpolation algorithm:
Step 1: Define the boundary conditions, i.e., specifying values to the known pixels, for the image region of support.

Step 2: Number the unknown pixels, i.e., all pixels within the region of support except the known pixels, in a raster scan manner.

Step 3: Initialize the unknown pixels.

Step 4: Calculate the unknown pixels using (4) (or (5)) in floating point arithmetic.

Step 5: Repeat step 4 until \( \max_{(m,n) \in U^{-}} \| I_{m,n}^{t} - I_{m,n}^{t-1} \| \leq T \). \( T \) is the preset tolerance.

Step 6: Quantize the unknown pixels to the nearest integers.

In step 1, the boundary conditions along the boundary of the region of support must be specified if (4) instead of (5) is used. In step 3, properly initializing the unknown pixels can reduce the number of iterations to achieve the required precision, but this step is not necessary. The initialization can be implemented in the light of an application. In step 5, the termination condition is related to step 6, which produces integer pixel values.

4. SIMULATION RESULTS

We shall demonstrate the usefulness and flexibility of the proposed interpolation algorithm in three cases. In the first case, where known pixels are located regularly on a Cartesian mesh, the results from the proposed algorithm are compared to those from bilinear interpolation. The second case deals with irregularly located known pixels. The third case considers missing image regions.

4.1 Regularly located known pixels

The case where known points are located regularly on a Cartesian mesh is commonly encountered in image interpolation. The test image, Lena, 512 pixels by 512 pixels, is first subsampled at a factor 2. The image interpolated by a factor 2 using the bilinear interpolation is shown in figure 2. For the proposed algorithm, since there are unknown pixels located along the boundary of the region of support, a mirror wall is put along the rim of the region of support. The result is shown in figure 3. Figures 2 and 3 resemble each other. The proposed algorithm requires 74 iterations with \( T = 0.1 \), \( \omega = 1.85 \), and unknown pixels initialized to zeroes.
As with our method, bilinear interpolation also produces unknown pixels obeying the min-max principle, i.e., the extrema occur only at the known pixels. In bilinear interpolation, an unknown pixel is affected by its four nearest known pixels, while in the proposed method, an unknown pixel is virtually affected by all known pixels. This effect may be desirable in some cases and undesirable in other cases. Therefore, with a large interpolation factor, the result from the proposed method and the result from bilinear interpolation appear more different. They are both not smooth. The cubic spline method (involving two separable one-dimensional interpolations) may be preferable for smooth interpolated image at a large interpolation factor. However, the cubic spline method will not obey the min-max principle.

4.2 Irregularly located known pixels

The advantage of the proposed image interpolation method is its flexibility in handling different sets of boundary conditions. In the case of irregularly located known pixels, the application of bilinear interpolation or cubic spline is not as simple as in the first case, though still amenable. On the contrary, the application of the proposed method is not affected. Figure 4 shows a picture showing the locations of known pixels and unknown pixels for Lena. Its black dot indicates a known pixel. A white dot represents an unknown pixel. If the pixels in the original image approximately satisfy (2), they are selected as the unknown pixels as the proposed interpolation method using (5) should be able to reconstruct them. The interpolated image is shown in figure 5, which closely resembles the original image. Sixty-one iterations are required with $T = 0.1$, $\omega = 1.85$, and unknown pixels initialized to zeroes.

4.3 Missing region

In the case of missing regions, especially regions with irregular boundaries, the application of bilinear interpolation or cubic spline is difficult. The application of the proposed image interpolation method is unaffected. Figure 6 shows a 40 x 40 image whose inner pixels are interpolated. The image is magnified for the sake of presentation. The boundary pixels vary from 214 at the top right to 150 at the bottom left. 83 iterations are needed, using $T = 0.1$ and $\omega = 1.85$ and initializing the unknown pixels to 0. The result shows a smooth transition between the boundary pixels.
5. CONCLUSIONS

A novel image interpolation algorithm is proposed. It is flexible in handling different configurations of unknown pixels and produces smooth transitions between their bounding known pixels. The method intrinsically models an image by the Laplacian equation, setting the interpolation problem as a boundary value problem. The boundary value problem is solved by successive over-relaxation iteration. Convergence to a unique solution is guaranteed by mathematical theorems. The proposed method is satisfactorily applied to three different cases of image interpolation: one that has regularly located known pixels, one that has irregularly located known pixels, and one that has missing regions.

6. REFERENCES

Figure 2(a). Image scaled up (x2) using bilinear interpolation

Figure 2(b). A zoom on the eye region of figure 2(a)
Figure 3(a). Image scaled up (x2) using the proposed interpolation method

Figure 3(b). A zoom on the eye region of figure 3(a)
Figure 4. A picture showing unknown pixels (white) and known pixels (black).

Figure 5. Interpolated image using the proposed method, using the irregularly located known pixels of figure 4.
Figure 6(a). A missing region (in black) of 40 x 40 pixels at the center

Figure 6(b). Interpolated image for figure 6(a)