INCREASING RESOLUTION OF DIGITAL ELEVATION MODELS USING BICUBIC PARAMETRIC PATCHES

Rolando QUINTERO
Geoprocessing Laboratory, Centre for Computing Research, National Polytechnic Institute
Av. Juan de Dios Bátiz s/n Unidad Profesional "Adolfo López Mateos"
C.P. 07738, México, D.F., MEXICO
quintero@cic.ipn.mx

ABSTRACT

In this work, we propose an algorithm based on bicubic parametric patches to change the scale of DEM for real-time simulation processes preserving the semantics of the elevation data. We merge this algorithm with the discrimination algorithm presented in [1], which manages optimally the huge quantity of data contained in DEM. These algorithms allow navigating along the huge quantity of information within the elevation data and, at the same time, increase the level of detail for areas of interest. An application of this method is image processing. It may be used to *zoom-in* images with a non-linear re-sampling.

KEYWORDS

Digital Elevation Model, Parametric Patches, Image Resolution.

1. INTRODUCTION

Nowadays, Digital Elevation Models (DEM) have gained the terrain in applications for simulating natural disasters. The decision making tools for disaster prevention become important worldwide because the human lives could have been saved, if we have preventive information. Nevertheless, applications for the simulation of natural disasters require a huge amount of data. In many cases, the available data do not have enough quality for simulation processes. The National Institute of Statistics, Geography and Informatics of Mexico (INEGI) produces DEM with 50 meters of resolution [2][3], but some simulation processes require a better level of detail.

In this work, we propose an algorithm[1] to increase resolution of DEM for real-time simulation processes without change the semantics of the elevation data. Also, the application of this algorithm in image processing to enlarge (zoom-in) images with a non-linear re-sampling is presented.

In our case, it is very important that the algorithm does not change the semantics of the DEM, thus it contains all relevant information about the model. We are working about the formal definition of DEM semantics. In Section 5 we describe the early results of our work about this formal definition. For the aims of this work, we will consider the semantics like the set of all the well-known values of elevation. The algorithm described in this work allows obtaining new data on the basis of these values.

In [1], we presented an application to manage the huge quantity of data contained in DEM for real-time rendering. In that application, we discriminated the less significant elevation data, without alter the semantics of these data. However, we cannot improve the quality of more relevant data to obtain additional information.

Using the algorithm presented in [1], we can solve the problem of 3D data representation and build virtual scenes, which are ready to navigate, either by simulations or by defined trajectories [6].

In the next section, we present some frameworks of the underlying theory of bicubic parametric patch representation. In Section 3, we give the pertinent considerations for the application of parametric patches and outline the proposed approach. In Section 4, some results and tests are presented and analyzed. Finally, the conclusions are outlined in Section 5.

2. BICUBIC PARAMETRIC PATCHES

In [1] we mentioned that elevation data can be represented as a polygon mesh. The step from polygon meshes to patch meshes is straightforward. If we consider a mesh of four-sided polygons approximating a curved surface, then a parametric patch mesh can be defined as a set of curvilinear polygons, which actually lie in the surface.

[1] A preliminary result of the development of the PhD thesis "Adaptive Methods for Generation of Digital Elevation Models" supervised by Dr. Serguei Levachkine

The definition given in [5] for a parametric surface (either B-spline or Bezier surfaces) $Q(u,v)$ is in terms of two parameters, u and v, where $0 \leq u \leq 1$ and $0 \leq v \leq 1$, and the function Q is a cubic polynomial. The accurate values of the coefficients in the cubic determine the curve. A special and convenient way of defining these is to use 16 three-dimensional points known as control points. The shape of the patch is fully determined by the position of these points.

A bicubic surface is defined by

$$Q(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} P_{ij} B_{i,j}(u,v) \, , \tag{1}$$

where P_{ij} is an array of control points an $B_{i,j}(u,v)$ is a bivariate basis function. We can generate $B_{i,j}(u,v)$ in the following form

$$B_{i,j}(u,v) = B_i(u) B_j(v) \, , \tag{2}$$

where $B_i(u)$ and $B_j(v)$ are the univariate cubic basis function. The definition of these basis functions describes the type of surface to be generated. Next we give the definition of basis functions for the surfaces used in this work: Bezier and B-spline surfaces.

2.1. BEZIER SURFACE PATCHES

Bezier surfaces properties are extended from the Bezier curves formulation in [5][7], we can find out these properties. A bicubic surface is defined by its basis function. In the case of Bezier surfaces the basis function is defined by the Bernstein polynomials (for a set of $n+1$ control points):

$$B_k^n(u) = C(n,k) u^k (1-u)^{n-k} \, , \tag{3}$$

where $C(n,k)$ are the binomial coefficients:

$$C(n,k) = \frac{n!}{k!(n-k)!} \, , \tag{4}$$

then, the description of the Bezier surface is given by

$$Q(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} P_{ij} B_i^n(u) B_j^m(v) \, . \tag{5}$$

2.2. B-SPLINE SURFACE PATCHES

Similar to Bezier surfaces, the B-spline surfaces are characterized by its basis function. In B-spline surface the value of the patch depends exclusively on the values of a 4×4 array taken from the control point set. For a set of $n \times m$ control points we will have $(n-3) \times (m-3)$ patches.

Considering that, the B-spline basis function is defined as

$$\begin{aligned} B_0(u) &= \frac{(1+u)^3}{6} \, , \\ B_1(u) &= \frac{3u^3 - 6u^2 + 4}{6} \, , \\ B_2(u) &= \frac{-3u^3 + 3u^2 + 3u + 1}{6} \, , \\ B_3(u) &= \frac{u^3}{6} \, , \end{aligned} \tag{6}$$

then, the patch Q_{ij} is defined for $i=0,1, \ldots, n-3$ and $j=0,1, \ldots, m-3$ by

$$Q_{ij}(u,v) = \sum_{k=0}^{3}\sum_{l=0}^{3} P_{i+k,j+l} B_k(u) B_l(v) \, . \tag{7}$$

3. APPLYING BICUBIC PARAMETRIC PATCHES TO DEM

In [1] we presented the steps required to obtain and set up all data and parameters for real-time DEM rendering. In that work we defined a grid $G(i,j)$ that contains the elevation data. To describe the rendering algorithm, we defined some parameters (their meanings are illustrated in Figure 1) first.

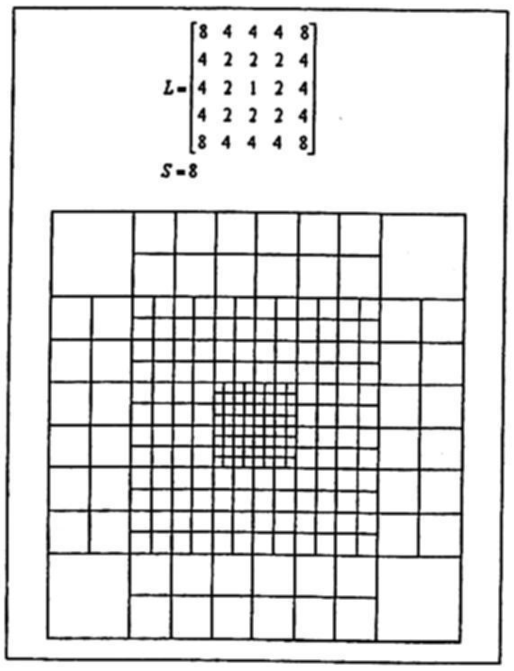

Figure 1. Visual representation of the parameters L and S of the algorithm. The grid at the bottom of the Figure illustrates how these parameters manage the number of polygons to be rendered.

- Matrix L of $H \times H$ defines the discrete level of detail to use. H is an odd number greater than 1, and $L[i,j] \neq 0$ for $1 \leq i \leq H$ and $1 \leq j \leq H$.
- Number S defines the optimization unit size; it means that it is necessary to optimize regions of $S \times S$ polygons.
- Vector o represents the observer position.

Using the parameters defined, we outline the algorithm.

```
RENDER (o)
  (o_x, o_y) ← RELATIVE-POSITION(o,G)
  for i = −½H to ½H
    x ← o_x+S(i − ½)
    for j = −½H to ½H
      y ← o_y+ S(j − ½)
      RENDER-BLOCK(x,y,L[i+½H, j+½H])

RENDER-BLOCK(x, y, lod)
  if lod > 0
    for i = x to x+S step lod
      for j = y to y+S step lod
        RENDER-QUAD(i,j,lod)

RENDER-QUAD(i,j,lod)
  RENDER-VERTEX(G[i,j])
  RENDER-VERTEX(G[i+lod,j])
  RENDER-VERTEX(G[i+lod,j+lod])
  RENDER-VERTEX(G[i,j+lod])
```

The algorithm is based on taking bigger blocks of data as far as they are, and represent them by sampling $G(i,j)$ in non regular steps along the grid. We can handle the number of polygons to be rendered by manipulating L-matrix and S- optimization size. In fact, the number of polygons to be rendered is given by,

$$N_P = \sum_{i=1}^{H}\sum_{j=1}^{H}\left(\frac{S}{L[i,j]}\right)^2. \tag{8}$$

3.1. PARAMETRIC CONSIDERATIONS

Thus, we can discriminate data from DEM. However, in the case when we need more detailed data than those that are in $G(i,j)$ we can apply parametric patches to obtain these intermediate data.

We will use the bicubic surfaces defined in Section 2. As a first approximation, we apply Bezier patches, which are commonly used in computer graphics, because they enable an efficient patch-splitting algorithm for rendering. But the main problem with Bezier patches is that the generated surface does not fit with the given points (control points). This is not desirable for the applications, thus we wish to increase the data resolution. We cannot fit all control points (the ones we have information) with the resultant Bezier patches. Nevertheless, by applying Bezier surfaces, we can compute how the whole data set behavior affects to a single point. See Equation 5.

On the other hand, we can apply B-spline patches. Thus, we can compute the inner points between the known ones (control points), using only local information. As we have mentioned before, a B-spline patch is always defined by a 4×4 control point array. So, with this type of surface we can find the new data without affecting the behavior of the whole data set.

3.2. THE ALGORITHM INTEGRATING PARAMETRIC SURFACES

To integrate the increasing resolution using parametric patches, we should modify the original algorithm in the following way. First, we should allow values less than one in L-matrix. Such values mean that we wish to obtain higher resolution for the block that is being rendered. On the other hand, we need to compute the parametric curve. This curve is stored in an alternate grid called Q. The values of Q will be defined by the control points and by the transformation matrix B (see [5] for matrix representation of Bezier and B-spline basis functions).

$$B = \frac{1}{6}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}. \tag{9}$$

Finally, the changes in the algorithm are the following:

```
RENDER-BLOCK(x, y, lod)
  if lod ≥ 1
    for i = x to x+S step lod
      for j = y to y+S step lod
        RENDER-QUAD(i,j,lod)
  else
    Q ← TO-SPLINE(G,x,y,lod)
    for i = x to x+S step lod
      for j = y to y+S step lod
        RENDER-QUAD-SPLINE(i,j)

TO-SPLINE(G,x,y,lod)
  P←CONTROL-POINTS(x,y)
  for u=0 to 1 step lod
    U←[u^3 u^2 u 1]
    for v=0 to 1 step lod
      V←[v^3 v^2 v 1]
      Q[u,v] ←U×B×P×B^T×V

RENDER-QUAD-SPLINE(i,j)
  RENDER-VERTEX(Q[i,j])
  RENDER-VERTEX(Q[i+1,j])
  RENDER-VERTEX(Q[i+1,j+1])
  RENDER-VERTEX(Q[i,j+1])
```

We only present the changes for integrating B-spline surfaces. Similar considerations must be taken for applying Bezier surfaces.

4. TESTS AND RESULTS

There have been done some performance tests with different data sets. As we have mentioned, the performance of the algorithm is constant, no matter the volume of elevation data involved. The results of the application are shown in Figure 2, using the proposed algorithm. The Figure 2a shows the result of rendering all elevation data. Figure 2b presents the result using our algorithm with the following parameters:

$$L=\begin{bmatrix}64 & 32 & 16 & 8 & 16 & 32 & 64\\ 32 & 16 & 8 & 4 & 8 & 16 & 32\\ 16 & 8 & 4 & 2 & 4 & 8 & 16\\ 8 & 4 & 2 & \frac{1}{2} & 2 & 4 & 8\\ 16 & 8 & 4 & 2 & 4 & 8 & 16\\ 32 & 16 & 8 & 4 & 8 & 16 & 32\\ 64 & 32 & 16 & 8 & 16 & 32 & 64\end{bmatrix}, \quad (9)$$

$$S=64$$

$$\Rightarrow N_p = 23524$$

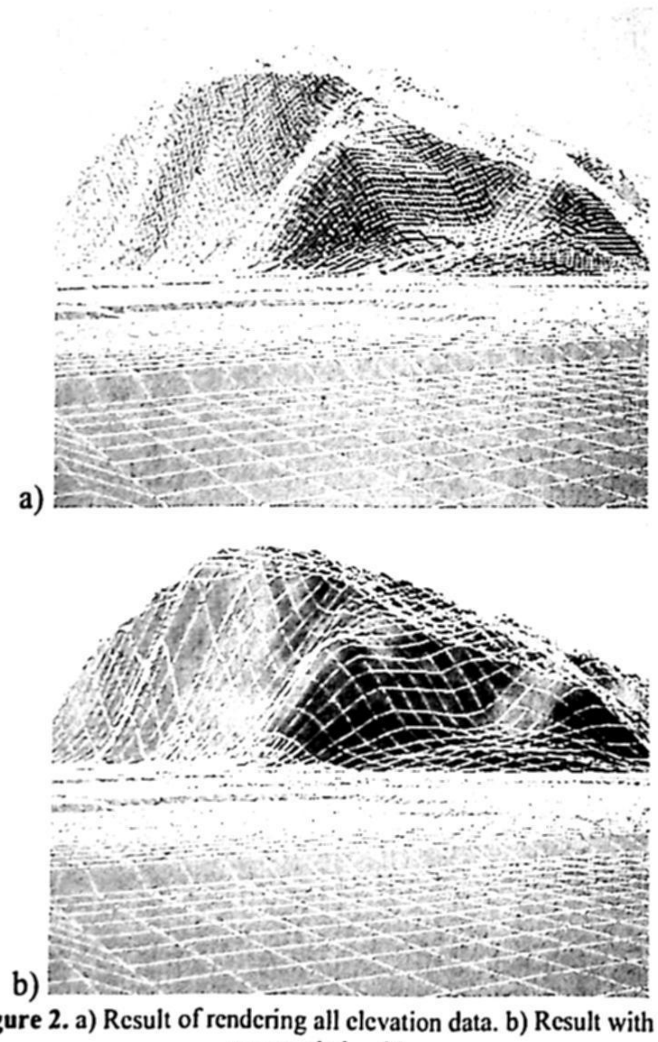

Figure 2. a) Result of rendering all elevation data. b) Result with the proposed algorithm.

Other data sets have been taken to test the *zoom-in* features of the algorithm. The results of this are shown in Figure 3. Figure 3a shows the data source used. Figures 3b and 3c present the results using our algorithm with B-spline and Bezier patches respectively.

As we can see, the B-spline patches allow us to interpolate the inner values of the image depending on their neighbors (because the B-spline basis function depends on 16 control points). The result contains the information about the local changes. The Bezier curve (Figure 3c) retrieves the information about all data, which allows applying smoothing to the image that not only depends on local values but also on the behavior of the whole data (memorizing of the data behavior).

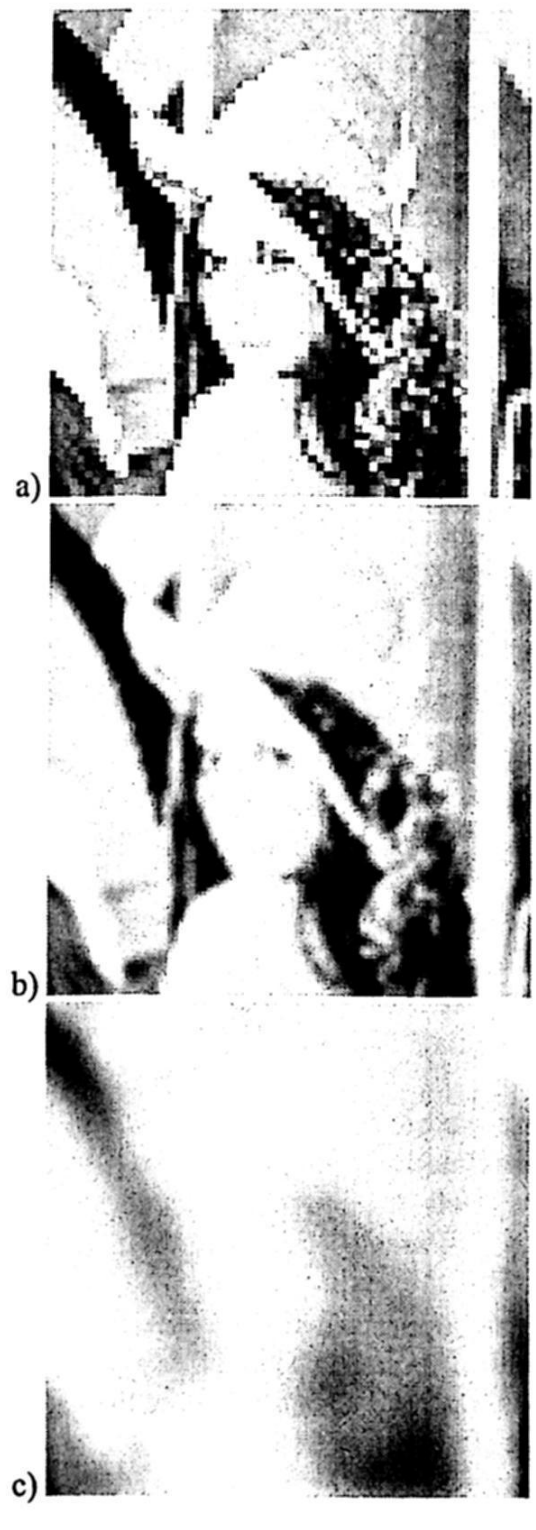

Figure 3. Example of the algorithm application (image of Lena on 64×64 resolution). a) Data source used. b) Image with B-spline 2x zoom. c) With Bezier 2x zoom.

In figure 4, we present the results of applying to a 32×32 image the most used algorithms to zoom images[1]:

[1] The zoom factor applied was 16x.

© Quintero GEOPRO 2003 – ISBN: 970-36-0103-0

Nearest neighbor gray-level interpolation (Figure 4a), Bilinear Interpolation [8] (Figure 4b) and Bicubic Interpolation [8] (algorithm proposed in this paper) (Figure 4c). Also, Figure 4 shows the image gradient to make notice the smoothness of the borders obtained with Bicubic Interpolation1.

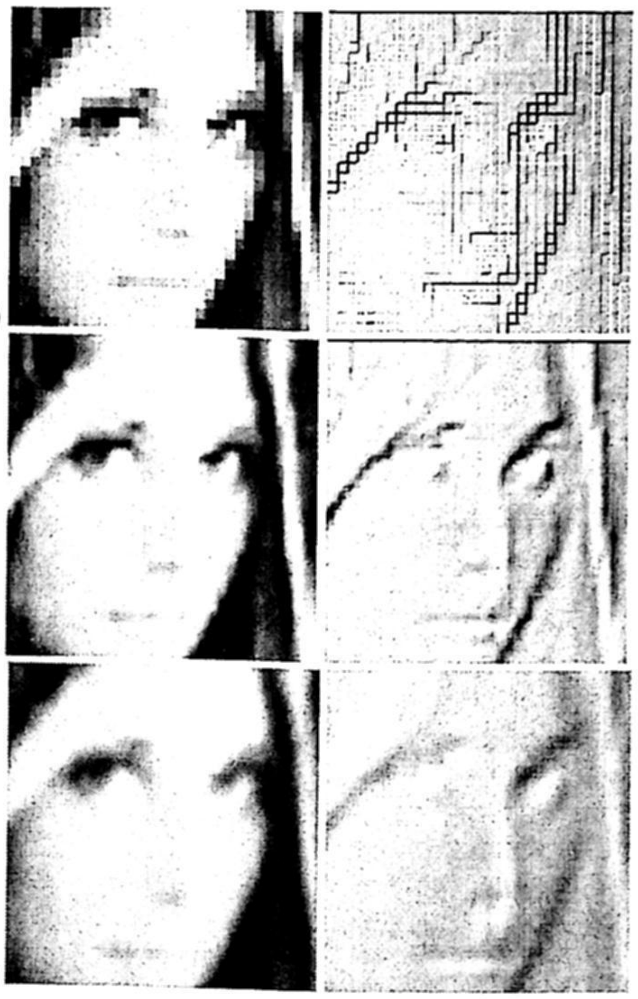

Figure 4. Results of applying different zooming algorithms to a 32×32 image. a) Nearest neighbor gray-level interpolation. b) Bilinear Interpolation. c) Bicubic Interpolation. The gradient obtained is shown as well.

5. CONCLUSIONS AND FUTURE WORK

In this work, the application has been designed to process spatial data in raster format. The implemented algorithm requires a low time to process the data, which are stored in DEM. The only restriction is that the images must correspond to the same scene. Two new digital images that are generated can be easily accessed in a faster way by the proposed rendering method.

The developed algorithm does not overload the processor, because it is very simple. Also, the rendering algorithm warranties a maximum number of elements to be rendered. In this way, we can manipulate this number (N_P) by varying the parameters of the matrix L. Additionally the algorithm does not modify the semantics of spatial data, it only discriminates or refines existing data without alter any characteristic of the elevation data.

This characteristic is essential to make the spatial semantic analysis.

An application of this algorithm in image processing is as follows: it may be used to *zoom-in* images with a non-linear re-sampling. Using parametric patches, it is possible to obtain the new values for the pixels between the known ones, containing local information (by means of B-spline patches) and global behavior (by means of Bezier patches) that improve the appearance of the enlarged image.

We are studying the application of recursive functions for the description of DEM, in order to obtain the semantics within the models. This can provide some significant advantages over the conventional ways of DEM description. First, it is possible to reach a very small granularity in the description and at the same time (another advantage) to obtain a very small (in amount of data) description. Thus, it can be possible to compress the huge amount of data in DEM. The application of the recursive functions can be seen as a model of adaptive approaches, in which an operation signature and a classification state (groups or classes) are registered in each recursion level. Under this scheme, the approaches can be refined every time adapting to the context of the land pattern. From this point of view, it is possible to say that the curves generated by the recursive functions adapt to the land shape.

DEM signatures as well as the classification states are obtained by applying recursively the classification function and calculating a measurement of the results2. The recursion finalizes when the measurement of the current result and the previous one are equivalent (the semantic has been preserved). With this information (the classification state and the operation signatures) it is possible to rebuild all the original DEM data.

As a future research; we must define the classification and measurement functions. As an early approach, we propose the classification function to be made of a set of basis functions. In the same manner as points, lines and polygons build a cartographic model, the basis functions will be used to describe DEM.

The selection of the basis function to be applied in each recursion level is an important problem to solve. We will have many basis functions then there will be many solutions in each level. To select the basis function and get the optimal result, we will apply different heuristics for the search of this optimal solution within the universe of possible solutions. In addition to the properties described in [9], the application of heuristic search is another reason that justifies the use of recursive functions.

1 Note the smoothness of the border in the right check and eyebrow

2 This measure is not defined yet, but it could be described as a classification quality rate.

ACKNOWLEDGMENTS

The authors of this paper wish to thank the Centre for Computing Research (CIC), General Coordination of Postgraduate Study and Research (CGEPI), National Polytechnic Institute (IPN) and CONACyT for their support. Additionally the authors wish to thank the reviewers for their pertinent comments.

REFERENCES

- R. Quintero, G. Guzman, M. Torres and M. Moreno, Spatial Analysis Based on 3D Rendering Techniques, *Proc. of 6th Association of Geographic Information Laboratories Europe conference on Geographic Information Science (AGILE 2003)*, Lyon, France, April 24 – 26, 2003, 401-408.
- INEGI, Modelos Digitales de Elevación, Generalidades y Especificaciones, Instituto Nacional de Estadística, Geografía e Informática de México, Aguascalientes, México, 1999 (ISBN: 970-13-2511-7).
- INEGI, Normas Técnicas para la Elaboración de Ortofotos Digitales, Instituto Nacional de Estadística, Geografía e Informática de México, Aguascalientes, México, 1999 (ISBN: 970-13-2510-9).
- A. Watt and M. Watt, Advanced Animation and Rendering Techniques, theory and practice. Addison-Wesley, New York, 1995.
- A. Watt, 3D Computer Graphics, 2nd edn. Addiscon-Wesley, New York, 1993.
- M. Reddy and Y. Leclerc, Enabling Geographic Support in Virtual Reality Modeling with Geo-VRML, Journal of Cartography and Geographic Information Science, 26(3), 1999.
- D. Hearn and P. Baker, Computer Graphics, 2nd edn. Prentice Hall, New York, 1994.
- C. Gonzalez and R. Woods, Digital Image Processing, 2nd edition. Prentice Hall, New York, 2002.
- V. Alexandrov and N. Gorsky, Image Representation and Processing: A Recursive Approach, Kluwer Academic Publishers, London, 1985.