Progressive Curve Representation Based on Reverse Subdivision

Faramarz F. Samavati[1], Mai Ali Nur[1], Richard Bartels[2], and Brian Wyvill[1]

[1] Department of Computer Science
University of Calgary,
Calgary, Canada
{samavati, mnur, blob}@cpsc.ucalgary.ca
[2] Department of Computer Science
University of Waterloo,
Waterloo, Canada
hbartel@uwaterloo.ca

Abstract. A progressive curve representation based on reverse subdivision is introduced. Multiresolution structures for common subdivision rules that have both banded reconstruction and decomposition filters are produced. Multiresolution filters are usually applied to the whole curve uniformly, while progressive curves are based on collapse and split operations that can be applied locally on any portion of a curve. In this work, firstly, small width multiresolution filters are constructed based on the reverse of the cubic B-spline subdivision. The collapse and split operations are replaced by a local decomposition and reconstruction process. Second, an efficient algorithm and data structures are presented to allow for the resulting progressive curve. Third, both a user-controlled and an automatic method to select a portion of the curve for reconstruction or decomposition are described. The technique introduced has various applications such as view-dependent rendering, flexible editing and progressive transmission.

1 Introduction

Curves are in many applications in CAD/CAM and computer graphics. Curves can be found as the basis for high quality font design, artistic sketches, data plots, 3D modeling and animation to manipulate object design and motion [14]. As described in [6], any flexible curve representation should allow for effective tools that include editing, smoothing and scan conversion. In this research, the problem of applying the operations listed below, on a segment of a curve are addressed. These operations have applications in such tools. The operations are:

– **Simplifying a segment of a curve (smoothing):** scanned data can often be replaced by simpler representations with less points. Current methods allow the user to reconstruct and decompose the whole curve [7, 6]. No method exists to apply reverse subdivision on a curve locally. Two approaches would be useful, a user-controlled method and an automated approach for segment

simplification. Possible applications include flexible editing, wherein, the user can lower the resolution of a curve to allow for easier editing, since there are fewer points to manipulate. To view the final result, a finer sequence of points can then be generated, for the curve segment (Figure 3).
- **View dependent rendering:** sometimes it is desirable to refine a region of a curve selectively. For view-dependent rendering, a segment of a curve can be shown close-up, as a finer set of points, while the remainder of the curve can be kept at a lower resolution (Figure 4).
- **Progressive Transmission:** when a curve is transmitted over a network, a low resolution curve segment followed by correction information are incrementally sent to get the higher resolution portion.

A multiresolution representation provides a uniform framework that addresses all these problems, if applied to the whole curve. It is useful to be able to manipulate the complete curve. An alternative approach is a progressive curve representation that is based on edge collapse and vertex split operations. This approach can be applied locally to change and enhance a portion of a curve. However, it is not based on subdivision curves, that are very important in computer graphics.

In this work, we introduce a new framework that replaces vertex-split and collapse operations based on reverse subdivision with reconstruction and decomposition operations. To achieve this, local filters are used and applied non-uniformly to curves, to do reverse subdivision. This has the advantage that the resulting curve can be created with high and low resolution segments simultaneously. In this work, local clusters with many points are replaced by clusters with fewer points, approximated using least-squares, that are a geometric good fit to the original group of points.

Section 2 describes previous work. Section 3 outlines details of how decomposition and reconstruction matrices are derived. Section 4 demonstrates applications of this work. Section 5 illustrates the data structure and algorithms used. Section 6 shows results. Section 7 summarizes the main concepts of this research and proposes some possible future work.

2 Previous Work

In this section, some relevant background work will be described, which includes subdivision, reverse subdivision and progressive curves.

2.1 Subdivision Curves

Subdivision curves start with a set of coarse points $\mathbf{c^k}$ and generate a larger set of points $\mathbf{c^{k+1}}$, using a subdivision matrix P. This process is repeated a finite number of times to produce the finer points. By successive application of subdivision, a hierarchy of curves can be obtained, which converge to a smooth curve. Subdivision can be stated by:

$$P\mathbf{c^k} = \mathbf{c^{k+1}} \tag{1}$$

Chaikin [3], Faber[4], Cubic B-Spline[16], Dyn-Levin-Gregory curves [5] are all example of subdivision curve schemes. The first three subdivisions are spline-based and are all examples of uniform knot insertion. For these curve schemes, subdivision reversal can be studied as uniform knot removal. There is other interesting work related to general knot removal [11].

An example of cubic B-spline subdivision scheme will be described. Given an initial control polygon $\mathbf{c^k}$. A new refined control polygon $\mathbf{c^{k+1}}$ is created with new points on the edges of the given polygon and with the given polygon vertex points in adjusted positions. The new points on the edge of the original control polygon are called c_{2i+1}^{k+1}. c_{2i}^{k+1} are the vertex points of the control polygon. c_i^{k+1}, c_{2i}^{k+1} and the cubic subdivision matrix, P, are:

$$
\begin{aligned}
c_{2i+1}^{k+1} &= \frac{1}{2}c_i^k + \frac{1}{2}c_{i+1}^k \\
c_{2i}^{k+1} &= \frac{1}{8}c_{i-1}^k + \frac{3}{4}c_i^k + \frac{1}{8}c_{i+1}^k
\end{aligned}
\qquad
P = \begin{bmatrix} & & \vdots & & \\ \dots \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \dots \\ \dots \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 & 0 \dots \\ \dots 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 \dots \\ \dots 0 & \frac{1}{8} & \frac{3}{4} & \frac{1}{8} & 0 \dots \\ & & \vdots & & \end{bmatrix}
$$

The limit curve of this subdivision is $\mathbf{C^2}$ and it will be exactly the cubic uniform B-Spline curve defined by the initial control polygon[17]. This is one of the most important curve schemes. The work in this research builds on the cubic B-Spline subdivision scheme. Nevertheless, the approach may be built on other subdivision schemes in a similar way.

2.2 Multiresolution Curves

Multiresolution (MR) is a representation which allows the user to change a high resolution to a lower one, in such a way that the original data can be reconstructed correctly. MR can be considered as a generalization of subdivision. A conventional approach to obtain an MR representation is based on wavelets [15]. Another approach to construct MR is using reverse subdivision [9, 2, 1]. which converts a high resolution approximation to a lower one, while simultaneously storing approximation-error (detail) information in a space-efficient manner. The high resolution approximation is recoverable by subdividing the lower resolution approximation and adding detail information. Both their operations are simple and fast. Figure 1, shows an MR structure. It shows an outline of an owl. The left-most image is a given fine set of points, while the middle and right-most images are successively coarser approximations. The user can easily switch between the fine and coarse images when there is an MR representation.

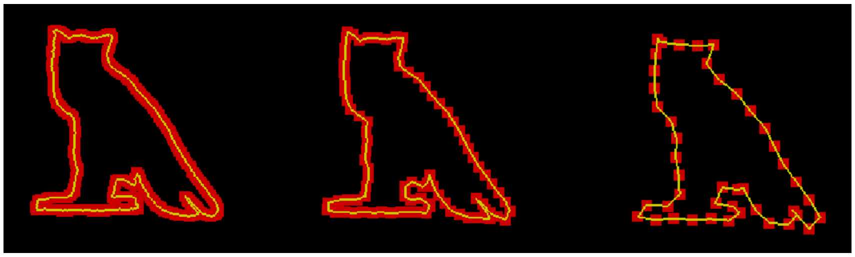

Fig. 1. Subdivision and reverse subdivision

To create MR structures, four matrices A, B,P and Q must be found, whose rows provide filters for decomposition and reconstruction. Assume that the points for a curve are given. Denote them by $\mathbf{c^{k+1}}$ (n points). They will be referred to as the fine data. It may be of interest to find an approximate set of m coarse points $\mathbf{c^k}$ (where $m < n$). By applying A, $\mathbf{c^k}$ is obtained:

$$\mathbf{c^k} = A\mathbf{c^{k+1}} \tag{2}$$

A is an $m \times n$ matrix. In order to have an accurate reconstruction of $\mathbf{c^{k+1}}$, the error terms of the approximation must be stored completely, as details. The details ($\mathbf{d^k}$) are captured as:

$$\mathbf{d^k} = B\mathbf{c^{k+1}} \tag{3}$$

A and B provide the decomposition filters. Decomposition is the process of splitting fine data into a low resolution part and details. The original data can be recovered using two matrices P and Q, providing the reconstruction filters. The reconstruction phase is as follows:

$$P\mathbf{c^k} + Q\mathbf{d^k} = \mathbf{c^{k+1}} \tag{4}$$

2.3 Progressive Curves

A progressive mesh [7] is an approach for constructing levels of details of meshes. It can also be used for curves. It is based on edge-collapse and vertex-split operations. An initial fine sequence of points can be simplified into coarse points by applying a sequence of edge-collapse operations. The inverse of a collapse is the vertex-split, which reproduces the fine mesh from a coarse one. An important issue that distinguishes the work presented in this work from a simple edge collapse operation is that a number of points are replaced by a new point, which is a least-square approximation. Simple edge collapse chooses a new point that is just conveniently chosen on the collapsed edge.

Progressive curves have many similarities to multiresolution curves since both of then store simpler structures and details. Both approaches allow the user to change between coarse and fine representations. A detailed discussion of the difference between both, in relation to global MR can be found in [7]. The framework presented in this research, is a combination of MR and the progressive structure. Like progressive structures, the user can apply the decomposition and reconstruction operation on any portion of the curve. However, the decomposition and reconstruction operations in this work are based on subdivision and its reverse rather than vertex split and collapse. As a result, a non-uniform distribution of points can easily be obtained, if needed, as shown in figure 2(b) all MR points are obtained by local approximation based upon least squares in a local area. By contrast, conventional MR filters must be applied uniformly on all the data points.

3 Progressive Structures Based on Multiresolution

In this research, a decomposition and reconstruction approach has been applied using MR filters to a portion of a curve. We have selected a MR method consistent with the cubic B-spline subdivision [2] for this paper, but the approach can be used for other subdivisions. In this case, the minimum non-trivial length of curve portions to collapse is five for this work, but the approach can be used for other subdivisions. A sequence of five points ($\mathbf{c^{k+1}}$) from the fine points are changed to three coarse points ($\mathbf{c^k}$), during decomposition:

$$\begin{aligned}\mathbf{c^{k+1}} &= [\mathbf{c_0^{k+1}}, \mathbf{c_1^{k+1}}, \mathbf{c_2^{k+1}}, \mathbf{c_3^{k+1}}, \mathbf{c_4^{k+1}}] \\ \mathbf{c^k} &= [\mathbf{c_0^k}, \mathbf{c_1^k}, \mathbf{c_2^k}]\end{aligned}$$

To preserve continuity between segments, $\mathbf{c_0^k}$ must be equal to $\mathbf{c_0^{k+1}}$ and $\mathbf{c_2^k}$ must be equal to $\mathbf{c_4^{k+1}}$. $\mathbf{c_1^{k+1}}$, $\mathbf{c_2^{k+1}}$ and $\mathbf{c_3^{k+1}}$ must be collapsed to a new point $\mathbf{c_1^k}$ located in a best least squares position determined from $\mathbf{c^{k+1}}$ points. This conversion means that two points are removed for any one of the curve segments. The procedure may be repeated for each curve part, as needed. To allow for full reconstruction, it is necessary to keep some additional information, as details. In this case, two details, $\mathbf{d_0^k}$ and $\mathbf{d_1^k}$ must be stored:

$$\mathbf{d^k} = [d_0^k, d_1^k] \tag{5}$$

For this scheme, the four matrices A, B, P and Q, described in section 2.2, need to be found. In order to construct these partial and local filters for the specific portion of the data, we use and manipulate the general technique that appeared in [2]. P is chosen as a small subdivision matrix for cubic B-spline that converts three coarse points to five fine points. This first row and last row of P have been selected as the identity rows to keep the first and the last points unchanged ($\mathbf{c_0^k}$ and $\mathbf{c_3^k}$). The other rows come from standard subdivision matrices.

Consequently, P is known and other matrices A, B and Q must be constructed consistent with P. In order to guarantee the full reconstruction of fine data from coarse data and details, the matrices must satisfy the bi-orthogonality condition [15] which is:

$$\begin{bmatrix} A \\ B \end{bmatrix} \begin{bmatrix} P & Q \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix}$$

In addition, it is preferred that the A matrix in Equation 2 produces $\mathbf{c^k}$ as a minimizer of the problem:

$$\min_{\mathbf{c^k}} ||\mathbf{c^{k+1}} - P\mathbf{c^k}|| \tag{6}$$

This guarantees that the details stored are small values. All these conditions are transformed to linear equations [2] and the solutions provide the matrices as:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \frac{-11}{36} & \frac{5}{12} & \frac{7}{9} & \frac{5}{12} & \frac{-11}{36} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad B = \begin{bmatrix} \frac{-11}{72} & \frac{5}{24} & \frac{7}{18} & \frac{-19}{24} & \frac{25}{72} \\ \frac{25}{72} & \frac{-19}{24} & \frac{7}{18} & \frac{5}{24} & \frac{-11}{72} \end{bmatrix}$$

$$P = \begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{8} & \frac{3}{4} & \frac{1}{8} \\ 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 1 \end{bmatrix}, \quad Q = \begin{bmatrix} 0 & 0 \\ 0 & -1 \\ \frac{15}{28} & \frac{15}{28} \\ -1 & 0 \\ 0 & 0 \end{bmatrix}$$

These matrices together with equations (2), (3) and (4) form a local MR representation that can be used for the decomposition and the reconstruction operations on the points and detail information of $\mathbf{c^{k+1}}$, $\mathbf{c^k}$ and equation (5).

4 Applications

The given progressive structure that is based on multiresolution has several applications such as scan conversion, flexible editing, view dependent rendering and curve compression. There are other techniques available for these applications, however our technique has the flexibility of the progressive structure, as well as consistency with multiresolution and subdivision methods (Figure 2(c,d)). This flexibility is due to decomposition and reconstruction operations that are based upon least squares (Equation 6). It is possible to choose any set of five successive points to collapse, consequently, the resulting coarse curve doesn't necessary have a uniform distributed set of points (Figure 2 (f)), however the details are meaningful (as can be seem by comparing Figure 2(c) to 2(d)), since they come from multiresolution filters. An important advantage over simple collapse/split progressive curves is that there is the possibility of using the subdivision scheme

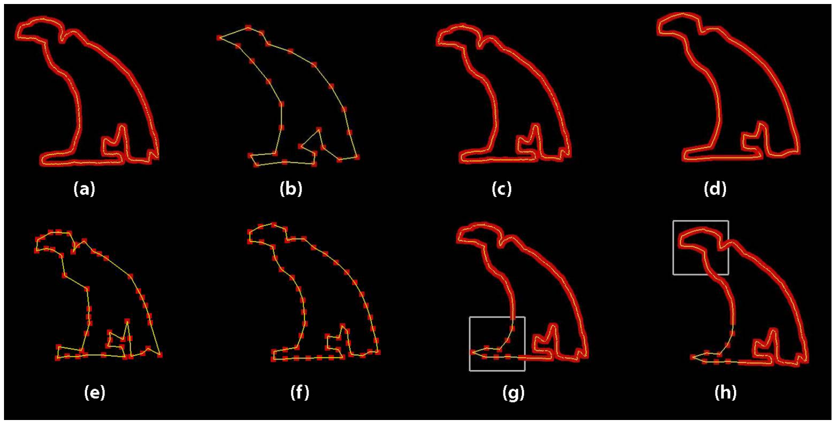

Fig. 2. **(a)** An eagle created with 3156 points. **(b)** A decomposed eagle. **(c)** Image (b) reconstructed using details. **(d)** Image (b) reconstructed by subdivision alone without details. **(e)** Image (a) after removing points using the metric described in section 4.2. The total number of points are 50. **(f)** Image (a) with 50 points, generated using the simple remove one and keep one algorithm. **(g)** A portion of image (a) is decomposed. **(h)** This image consists of three different segments, a high resolution part, a lower resolution part and a portion that was decomposed and reconstructed without details (the area selected by the window)

to partially enhance the curve (Figure 2(d)), because the local multiresolution filters are constructed from the reverse subdivision.

A crude, simple approach to choose points to decompose, would be to select them at random. However, using a more sophisticated approach will more likely guarantee that the resulting curve represents the initial scanned points. Another useful consideration is to provide a user-controlled method as well as an automated approach. The two methods used in this work, to eliminate redundant points in scanned data, include the window selection and metric approach.

4.1 Decomposing a Curve Segment Using Window Selection

This method allows the user to control which portion of a curve is decomposed. All points in this part of the curve are divided sequentially to groups of five points. Then the decomposition operation is applied to them. This process is repeated until a certain number of points desired, is reached. An example is shown in figure 3. This diagram illustrates an application to edit Arabic fonts. The user moves a window over the portion of the letter that is to be edited. This segment has lots of control points, which make it tedious to select a specific point. The points in the selected region are then decomposed. The user can then manipulate the control points and elongate the letter. The edited segment of the font is then reconstructed to finer points, to visualize the final resulting letters. That is, $\mathbf{c^{k+1}}$ are decomposed to $\mathbf{c^k}$ and $\mathbf{d^k}$. The $\mathbf{c^k}$ points are edited to produce altered points $\mathbf{e^k}$, a new version of the fine curve is produced as $P\mathbf{e^k} + Q\mathbf{d^k}$.

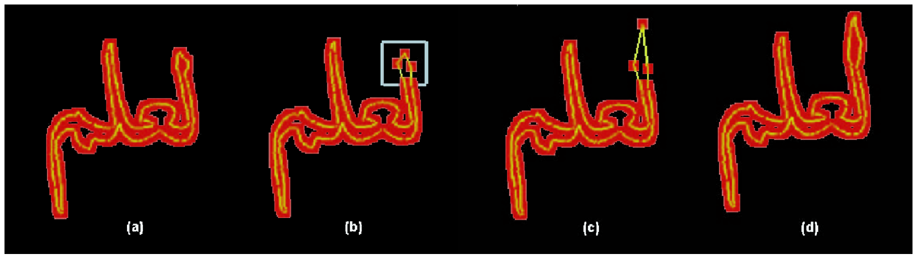

Fig. 3. (a) Arabic font created with 3095 points. (b)The user selects a portion of the font. The number of points in the selected region are reduced. (c) The letter is modified in the area with reduced points. (d) The decomposed segment of the curve is reconstructed.

4.2 Metric Ordering

This is an automated approach, to reduce points that represent a segment of a curve, using a metric. Examples of possible metrics can be found in [8, 7]. The metric used in this research is based on curvature. Areas of lower curvature are decomposed and higher energy portions are preserved. The curvature is calculated by finding the distances and angle between a point and it's neighbors. The metric values m_{c1} calculated for point $\mathbf{c_1}$ can be written as:

$$m_{c1} = w_\theta|\theta| + w_e(e_0 + e_1) \tag{7}$$

e_0 and e_1 are the Euclidean distances between points c_1 and it's neighbors c_0 and c_2. The weights w_θ and w_d allow the user to tweak the metric as desired, to give either distance or angle criteria, more importance. All weights w_θ and w_d, used in this research were set to 0.5.

Figure 2(e) shows results using the metric. The original eagle is made up of 3156 points. Using the metric, segments of the curve are repeatedly decomposed until 50 points remain. For comparison purposes, a simple scheme is used to reduce the original eagle to 50 points (Figure 2(f)). This simple scheme keeps one point and removes the following point. Again the process is repeated until 50 points remain. As can be seem from the figure, using the metric in equation 7 means that fine details such as the sharp hump on the eagle's back, beak details, the bottom of the wing parts, are all preserved. All these details are lost using the simple scheme (Figure 2(f)).

5 Algorithms and Data Structures

In this section the algorithms and data structures that are used in the progressive curve are outlined. To implement the window selection case in section 4.1, it is possible to use a simple data structure, such as an array. However, the data structure used, has been designed to work with the metric ordering in section 4.2. The essential demand of data management, aside from correctly associating $\mathbf{d^k}$ with $\mathbf{c^k}$ information, is to keep track of the location and order at which decompositions take place so that reconstructions can take place in strictly reverse order to decompositions. The main data structure used, consists of a linked-list of elements (Element List). Each element consists of the following components:

- **Position[3]**: an array representing the x,y and z co-ordinates of a point.
- **Details**: a data structure that stores details, for the correct reconstruction.
- **Metric Value**: a double that is calculated from equation 7.

Details are stored as a ternary tree structure. For the three decomposed elements, two details must be stored, as described in Section 3 and a hierarchy of all the details is stored. This allows any of the curves, in any level of the hierarchy to be regenerated correctly. The data structure for the details is given by:

- **$\mathbf{d_0}$ and $\mathbf{d_1}$**: two doubles representing first and second details
- **dleft, dmiddle and dright** : three pointers, one to a new left detail, middle detail and a right detail structure.

dleft, dmiddle and dright store details for the fine points, $\mathbf{c_1^{k+1}}$, $\mathbf{c_2^{k+1}}$, $\mathbf{c_3^{k+1}}$. Two more data structures are needed:

1. **A pointer to the Element List**: this list is sorted based on the metric defined in section 3. It is needed for the decomposition stage. The first pointer in this list, points to an element that should be decomposed with its left and right neighbors. During the decomposition stage, these three points are removed. The pointer list is then updated locally by inserting a new point at the suitable location. This process is repeated, until a certain percentage of points is deleted.
2. **A stack of indices of the coarse elements**: as each element is removed, the index of the new coarse element is stored in a stack. This is needed to reconstruct a curve correctly. As described in section 4.2, the decomposition stage selects points based on a metric. Any five points can be chosen to be decomposed to three coarse points. During the reconstruction phase, it is crucial that the same three points be used, in order to return to the original five points, this is guaranteed with the stack structure.

6 Results

To demonstrate the progressive curve, several examples are shown. For Figure 1, the structure has been reverse subdivided globally as in [2]. The left diagram is a fine mesh and the coarse representation is shown in the middle and a coarser representation is shown in the right-most figure. Figure 2 (b) shows the eagle outline. The figure shows fine set of points, coarse points, a reconstructed curve using details ($P\mathbf{c^k}+Q\mathbf{d^k}$) and without details ($P\mathbf{c^k}$ alone). The figure also shows a representation created with 50 points, generated by using the metric described in section 4.2 and by using the simple keep one and remove one approach for comparison (Figure 2 (e) and (f)). As described in section 4, results in 2 (e) are superior to results in 2 (f). Lastly the use of the window as a selection medium is shown. Figure 2(h) shows a curve created with three different segments. Figure 3 illustrates flexible editing, using the window selection feature. Figure 4 show the stages to view-dependent rendering. Lastly, an example of scanned data is shown in Figure 5.

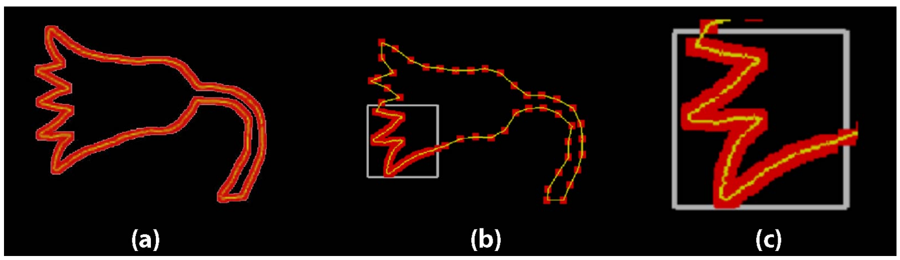

Fig. 4. View dependent rendering. (a) The lotus flower with a fine set of points. (b) The flower was decomposed and an area selected, using a window was reconstructed. (c) A close-up of the curve component selected by the window in (b)

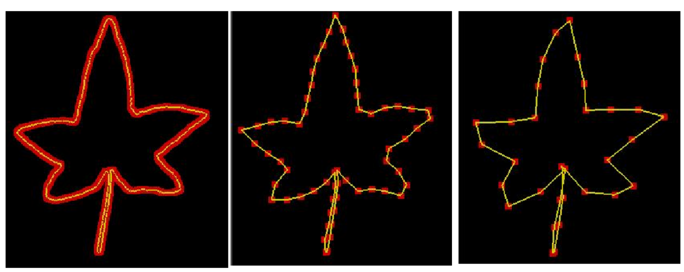

Fig. 5. High and lower resolution of scanned data

7 Conclusions

This work combines both multiresolution and progressive structures together. A multiresolution system has been constructed using the cubic B-spline subdivision approach. The decomposition and reconstruction operations are derived from the multiresolution filters. The efficiency of the method is presented. Although, this work is based on the cubic B-spline subdivision, The approach can be applied to any other curve subdivision scheme such as Chaikin and Dyn-Levin-Gregory.

Generalization of this work to surfaces can be considered as a possible direction of future work.

8 Acknowledgments

We would also like to thank the Department of Computer Science of the University of Calgary. This work is partially sponsored by the Natural Sciences and Engineering Research Council of Canada.

References

1. Faramarz F. Samavati, N. Mahdavi-Amiri and R. H. Bartels: Multiresolution Representation of Surface with Arbitrary Topology by Reversing Doo Subdivision, Computer Graphic Forum, Vol.21, No.2, 121–136, (2002)
2. Richard H. Bartels and Faramarz F. Samavati: Reversing subdivision rules: local linear conditions and observations on inner products. Journal of Computational and Applied Mathematic, (2000) 29–67 ,Vol. 119
3. G. Chaikin: An algorithm for High Speed Curve Generation. Computer Graphics Image Processing 3 (1974) 346–349
4. G. Faber: Über Stetige Functionen, Math. Ann. 66 (1909) 81–94
5. N. Dyn, D. Levin and J. Gregory: A 4-point Interpolatory Subdivision Scheme for Curve Design. Computer Aided Geometric Design 4 (1987) 257–268
6. Adam Finkelstein and David H. Salesin: Multiresolution Curves. Proceedings of Siggraph (1994) 261–268
7. Hugues Hoppe: Progressive Mesh. Proceedings of Siggraph (1996) 99–108
8. Hugues Hoppe and Steve Marschner: Efficient Minimization of New Quadratic Metric for Simplifying Meshes with Appearance Attributes. Technical Report MSR-TR-2000-64 (2000), Addendum to IEEE Visualization 1999 paper.
9. Faramarz F. Samavati and Richard H. Bartels: Multiresolution Curve and Surface Representation by Reversing Subdivision Rules by Least-Squares Data Fitting Computer Graphics Forum (1999) 97–120,Vol. 18, number 2
10. Samuel Hornus and Alexis Angelidis and Marie-Paule Cani: Implicit Modelling Using Subdivision-curves. The Visual Computer, 2002
11. Matthias Eck and Jan Hadenfeld: Knot Removal for B-Spline Curves. Computer Aided Design, (1995) 259–282, Vol. 12
12. Denis Zorin and Peter Schroder: Subdivision for Modeling and Animation. ACM Computer Graphics (Course Notes 2000)
13. Thomas Strthotte and Stefan Schlechtweg: Non-Photorealistic Computer Graphics Modeling, Rendering and Animations. Morgan Kaufmann (2002), Magdeburg, Germany
14. James D. Foley, Andries Van Dam, Steven K. Feiner and John F. Hughes: Computer Graphics Principles and Practice, Second Edition. Addison-Welsey Publishing Company (1990), Massachusetts, United States
15. E. J. Stollnitz, T. D. DeRose and D. H. Salesin: Wavelets for Computer Graphics. Morgan Kaufmann Publishers (1996)
16. J. Warren and H. Weimer: Subdivision Methods for Geometric Design. Morgan Kaufmann Publishers (2002)
17. Richard H. Bartels , John C. Beatty , Brian A. Barsky: An introduction to the use of splines in computer graphics. Morgan Kaufmann Publishers Inc. (1986)