FILTERING AND OPERATIONS WITH GEOMETRIC PROCESSING OF GREY LEVEL PICTURES/IMAGES

Sri Lakshmi U*1, Mr. Siva Rama Krishna K*2, Dr.Raja Prakash Rao P*3

1. *II.M.Tech-CSE, TRREC*
2. *Associate Professor, Dept of CSE, TRREC*
3. *Professor & Head, Dept of CSE, TRREC*

Abstract: *This paper describes an algorithm to implement image filtering and enhancement operations by processing adaptive triangular meshes that represent gray-level images. Experimental results show that these operations are significantly more efficient when they are performed upon triangular meshes than by sequentially processing all the pixels in the given images.*

1. INTRODUCTION

Image compression has undergone a notorious development due to the increasing need to store and transmit large digital images. Conversely digital image processing is still performed up on uncompressed images, no matter how big and redundant they are, since the commonly used compression formats were not originally devised for applying further processing. Nonetheless some researchers have managed to apply a limited number of basic operations ,such as arithmetic, feature extraction and scaling ,up on such compressed representations (e.g.,[1][2]).

Alternatively, digital images may be represented with geometric models, such as adaptive triangular meshes. Triangular meshes are advantageous since a single triangle is able to approximate the gray levels of a certain area of pixels. For example, a large white background, which may comprise thousands of pixels in an image, can be represented by a couple of triangles in an adaptive mesh. Several techniques can be utilizes to obtained triangular meshes from gray-level images (e.g., [3][4][5][6][7]).

Taking advantage of their representation benefits, triangular meshes have been proposed to accelerate image segmentation algorithms (e.g.,[8][9]). However, their suitability for other image processing and computer vision operations is still an open issue. In this respect, this paper explores the utilization of adaptive triangular meshes to accelerate image filtering and enhancement operations, and shows that these simple operations can run more efficiently in the geometric domain than when they are implemented by sequentially processing all the pixels of the image.

Section 2 describes the proposed image filtering and enhancement technique. Conclusions and future work are given in section 3.

**** Sri Lakshmi U***
II.M.Tech-CSE, TRREC

2. IMAGE FILTERING AND ENHANCEMENT FROM ADAPTIVE TRIANGULAR MESHES

Two techniques for filtering and enhancing gray-level images represented with adaptive triangular meshes are described in this section. A technique for computing histograms from triangular meshes is also presented. The triangular meshes utilized in this paper were generated with the algorithm presented in [7].

Adaptive triangular meshes representing gray-level images are defined in a 3D space in which coordinates X and Y represent the columns and rows of the image space , and coordinate Z represent the gray level. Every vertex of a triangular mesh is associated with a *polygonal patch*, the letter being the set formed by that vertex and its adjacent neighbors sorted in counter-clock wise order. Two vertex of the mesh are adjacent if there exists an edge between them.

2.1. FILTERING OF AN IMAGE WITH TRIANGULAR MESHES

The proposed filtering technique computes a new triangular mesh from the given one. The new mesh is the same as the original one except from the Z coordinates of its vertices.

Figure1. (Left) original image (House: 256x256). (Middle) Adaptive triangular mesh that approximates the original image 1,649 vertices. (right) approximating image obtained from the previous mesh, RMS error equal to 10.6 for 256 gray levels.

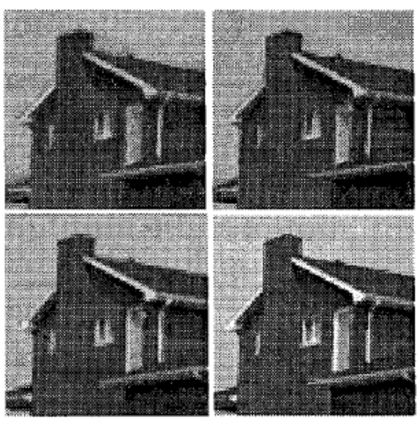

Figure 2. (top row) filtered images, with the proposed technique : (left) mean filter,(right) median filter.(bottom row) filtered images with CVIPtools: (left) mean filter,(right) median filter.

The Z coordinate of each vertex of the new mesh is computed as a function of the original Z coordinates of the vertices contained in the polygonal patch associated with the vertex. This process mimics the classical convolution technique applied to filtering digital images. Both a mean and a median filter have been implemented.

The mean filter computes the Z coordinate of every vertex in the new mesh as the average of the original Z coordinates of the vertices that belong to the vertex's polygonal patch. The median filter sorts out the Z coordinates of every polygonal patch. The new Z coordinate of each vertex whose associated sorted list has

an even number of values is the average of the two values in the middle of the list. If the number of values in the sorted list is add, the new Z coordinate is just value in the middle of the list. Fig. 1 shows (*left*) an original 256x256x8 image, *(middle*) an adaptive triangular mesh that approximates this image and (*right*) the approximating image obtained from that mesh through a sampling algorithm described in section 2.2. Fig. 2 shows the approximating images corresponding to the triangular meshes obtained by applying the proposed (*top-left*) mean and (*top-right*) median filters to the mesh in Fig. 1... Fig 2(*bottom*) shows equivalent filters applied to the original image with a conventional image processing algorithm *(CVIPtools*, [9]), which processes the given image pixel by pixel.

The original triangular mesh was filtered in 0.8ms (mean) and 3.8ms (median) with the proposed technique. Similar filters applied to the original image with CVIPtools took 290ms (mean) and 170ms (median)

2.2 Histogram generation

The three enhancement techniques presented in this paper modify the histograms associated with the original gray-level images. This section describes a technique for generating histograms from triangular meshes. Those histograms are not utilized for the enhancement techniques themselves but for verifying their results.

In order to generate a histogram from an adaptive triangular mesh, an array with as many counters as gray-levels is utilized. All the counters are initially set to zero. Then the triangles of the given mesh are considered sequentially.

For each triangle, it is necessary to determine which pixels of the original image are contained in the projection of that triangle on the XY plane, considering that coordinates X and Y in the mesh space are equivalent to the column and row coordinates in the image space. This is done by, first determining the pixels contained in the triangle's bounding box. From those pixels, the ones contained in the triangle are then selected. Finally, the Z Coordinates of those pixels are obtained by evaluating the plan equation associated with that triangle at the x and Y coordinates that correspond to the selected pixels.

Each Z coordinate represents a gray-level determined by the triangular mesh. The counter associated with that gray-level is then increased. When the previous process has been applied to all the triangles of the mesh, the array of counters keeps the image histogram. This histogram is normalized by dividing the counters by the total number of evaluated pixels.

The triangle sampling process described above is also utilized to generate the gray-level image corresponding to a triangular mesh (approximating image). Several approximating images generated by applying the previous process, as well as their corresponding histograms, are shown in fig.3 and Fig.5.

2.3 image Enhancement

This section describes three enhancement algorithms that modify the gray-level distribution of a given triangular mesh by applying a mapping function that either stretches, shrinks or slides the histogram corresponding to that mesh. This histogram can be computed with the technique described in the previous section.

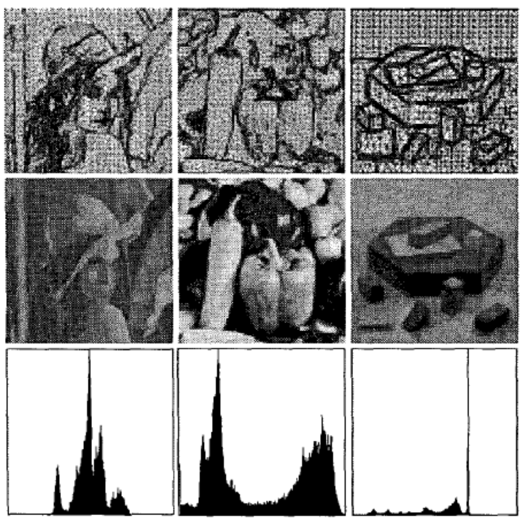

Figure 3. (top-row) adaptive triangular meshes: (left) 7, 492, (middle) 7, 370 and (right) 1, 593 vertices.(middle-row) approximating images from the previous meshes.(left)512x512 (middle)512x512 and (right)256x256

pixels.(*bottom-row*)histogram computed from the given meshes in 2.87,2.46 and 0.87 sec respectively.

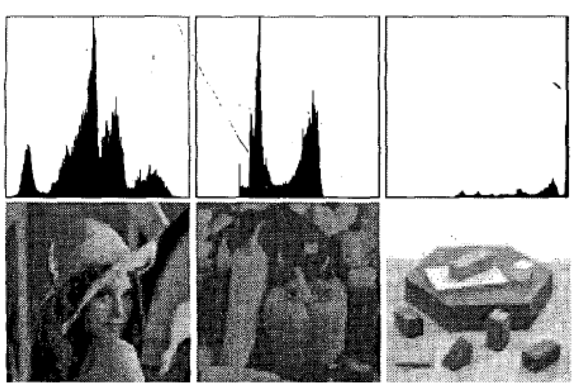

Figure 4. Image enhancement with the proposed technique. (top-row) histograms after applying :(left) stretching (middle) shrinking and (right) sliding. (Bottom-row) Approximating images after :(left) stretching, (middle) shrinking to the range [60,180] and (right) sliding.

The histogram stretching process computes a new Z coordinate for every vertex of the given triangular mesh by applying the following equation [10]:

$$Z' = Z - Zmin \div Zmax(MAX - MIN) + MIN$$

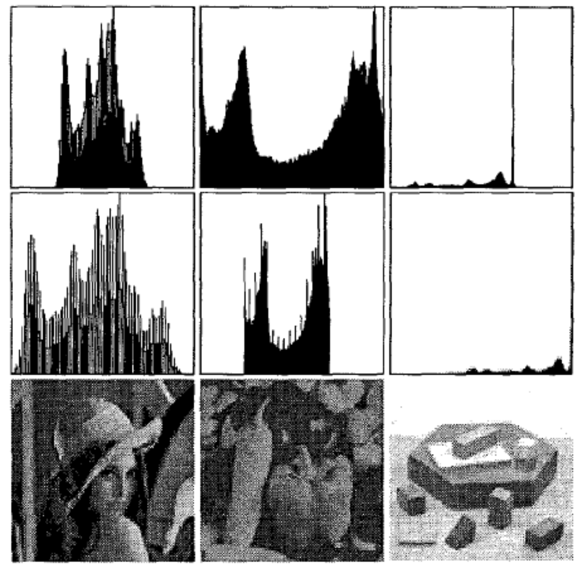

Figure 5. Image enhancement with CVIPtools. (top-row) histograms computed from the input image. (middle-row) histograms after applying :(left) stretching (middle) shrinking and (right) sliding. (bottom-row) Resulting images after: (left) stretching, (middle) shrinking to the range [60, 180] and (right) sliding.

Where Zmax and Zmin are the largest and smallest Z coordinates of the given mesh, while MAX and MIN are the maximum and minimum gray levels corresponding to the current pixel depth (255 and 0 for 8-bit image) shows the histogram and approximating image obtained after applying the stretching procedure to the triangular mesh shown in Fig.3 (*left*).

The histogram shrinking process consists of applying the same previous equation to all the vertices of the triangular mesh, although, in this case, MAX and MIN are the maximum and minimum gray levels desired for the compressed histogram. An example that shows the final histogram and its approximating after applying the shrinking procedure is given in Fig.4 (*middle*), considering the triangular mesh shown in Fig. 3(*middle*)

Finally , the mapping function for the histogram sliding process is defined as [10]:Z'=Z+offset, with the offset being the gray-level displacement applied to the histogram associated with the original triangular mesh shown in Fig.4(*right*).Shows the histogram and resulting approximating image corresponding to the application of the sliding process to the triangular mesh shown in Fig. 3(*right).*

The CPU time for both the stretching and shrinking processes in the previous examples was 1.3ms.qualitatively similar results were obtained with CVIPtools in 190ms. The sliding process took 0.1ms with proposed technique and 10ms with CVIPtools. Fig .5 shows the results obtained by applying the three image enhancement processes with CVIPtools.

3. EXPERIMENTAL RESULTS

The proposed techniques have been tested upon several triangular meshes obtained from gray-level images. The CPU times have been measured on a SIG indigo II workstations with a 175MHz R10000 processor. Results for 4 of these images are presented in this section: house (256x256x8), Lenna (512x512x8), peppers (512x512x8) and blocks (256x256x8). These images were approximated with adaptive triangular meshes obtained by applying the technique presented in [7]. These meshes contained: (House) 1,649, (Leena) 7,492, (peppers) 7,370 and (Blocks) 1,593 vertices respectively.

The same filtering and enhancement operations have been applied to the original gray-level images by using the public image processing library CVIPtools [10]. Table 1 shows the CPU times for computing, the mean and median filtering operations with both the proposed technique and with CVIPtools (the latter with 5x5 masks):

Technique	**Image**	**Mean Filter**	**Median Filter**
Proposed	House	0.0008	0.0038
CVIPtools	House	0.29	0.17
Proposed	Leena	0.0123	0.0255
CVIPtools	Leena	1.25	0.66
Proposed	Peppers	0.0123	0.025
CVIPtools	Peppers	1.25	0.66
Proposed	Blocks	0.0008	0.0037
CVIPtools	Blocks	0.29	0.17

Table 1: CPU times in seconds of filtering operations with the proposed technique and with CVIPtools.

Table 2 shows the CPU times for computing the stretching and sliding operations with both the proposed technique and with CVIPtools:

Technique	**Image**	**Stretch**	**Shrink**	**Slide**
Proposed	House	0.00024	0.00024	0.00014
CVIPtools	House	0.05	0.05	0.01
Proposed	Leena	0.0013	0.0013	0.00057
CVIPtools	Leena	0.19	0.19	0.04
Proposed	Peppers	0.0013	0.0013	0.00057
CVIPtools	Peppers	0.19	0.19	0.04
Proposed	Blocks	0.00024	0.00024	0.00014
CVIPtools	Blocks	0.05	0.05	0.01

Table 2: CPU times in seconds of enhancement operations with the proposed technique and with CVIPtools.

In the previous example, the execution upon triangular meshes is two orders of magnitude faster when working upon triangular meshes than when the original images are proposed. The obvious reason is that, in the first case, only the vertices of the triangular meshes must be sequentially proposed while, in the second case, all the pixels of the gray-level images must be considered.

4. CONCLUSIONS

This paper presents basic algorithms to perform image filtering and enhancement operations directly working upon adaptive triangular meshes that represent gray-level images. Experimental results show that these operations are significantly more efficient than their pixel-to-pixel counterparts. We are currently implementing and evaluating other image processing operations that work upon adaptive triangular meshes, including algorithms to perform image thresholding and labeling [11], segmentation, and logic and arithmetic operations.

5. REFERENCES

P2P integration (peer to peer) –It is a decentralized approach where data is mutually shared and integrated at every peer location. This is entirely dependent on integration functionality available at all the peer location.

1. B. Smith and L. Rowe.Alorithms for manipulating compressed images.*IEEE Computer Graphics and applications*, 13(5), September 1993, pp...34-42.

2. S.-F. Chang. Compressed –domain techniques for image/video indexing and manipulation. IEEE Int.conf.on image Processing, Washington DC, October 1995.

3. M.Vasilescu and D.Terzopouls. Adaptive meshes and shells: irregular triangulation,. Discontinuities and hierarchical subdivision. IEEE Int. Conf. on Computer vision and Pattern recognition, Champaign, USA, June 1992, pp.829-832.

4. L.DeFloriani, a pyramidal data structure for triangle-based surface description. .*IEEE Computer Graphics and applications*, 9(2), March 1989, pp...67-78.

5. D.Terzopoulos and M.Vasilescu sampling and reconstruction with adaptive meshes. IEEE Int. Conf. on Computer vision and Pattern recognition Hawaii, USA, June 1991, pp.70-75

6. M.A.Garcia, B.X.Vintimilla and A.D.Sappa. Efficient approximation of gray-scale images through bounded error triangular meshes. IEEE Int.conf.on image Processing, Kobe, japan, October 1999.

7. M.A.Garcia, B.X.Vintimilla and A.D.Sappa. "Approximation and processing of Intensity Images with Discontinuity Preserving adaptive triangular meshes". Accepted for publication in sixth European conference on computer vision, Dublin, Ireland, July 2000.

8. M.A.Garcia, L.Basanez, Fast extraction of surface primitives form range images, 13th IAPR Int. Conf. on Pattern recognition, vol. III: application and Robotic systems, Vienna, August 1996, pp.568-572.

9. T.Gevers and A.Smeulders. Combining region splitting and edge detection through guided Delaunay image subdivision . IEEE Int. Conf. on Computer vision and Pattern recognition, Puerto Rico, June 1997, pp.1021-1026.

10. S.E. Umbrage. Computer Vision and Image Processing Prentice-Hall International Editions, 1998.

11. M.A.Garcia, B.X.Vintimilla. "Acceleration of Threshoulding and Labeling operations Through Geometric proposing of Gray level Images". Accepted for publication in 15th IAPR. IEEE Int. Conf. on Pattern recognition, arcelona, Spain, September 2000.