International Journal of Computing and Digital Systems

A Video Complexity Index for Cluster Computing

Demóstenes Zegarra Rodríguez, Renata Lopes Rosa, and Graça Bressan

Department of Computer Science and Digital Systems, University of São Paulo, SP, Brazil

E-mail address: demostenes@larc.usp.br, rrosa@usp.br, gbressan@larc.usp.brgbressan@larc.usp.br

Received 27 Oct 2013, Revised 8 Nov 2013, Accepted 13 Nov 2013, Published 1 Jan 2014

Abstract: Entertainment applications that work with images and videos processing use the Cluster computing to decrease their rendering time. A complexity level can be experimentally determined according to the time spent to render a video. This paper study the time spent for rendering 3D videos on a Beowulf cluster computing with PovRay software on a GNU/Linux environment. For another hand, the video complexity is determined by a novel metric named Video Complexity Index (VCI) that considers both, the spatial and temporal video characteristics, the metric is tested with videos of different complexity. Experimental results with the cluster number increasing demonstrate that VCI metric successfully classifies the video complexity according to the number of nodes used to render them, considering the processing time consumed for a cluster. So, VCI metric can be very useful to find the minimal number of nodes depending of the 3D video complexity.

Keywords: Cluster Computing, video complexity index, spatial and temporal video characteristics, 3D video complexity.

I. Introduction

An image or video can have different complexity level, and this complexity influences your rendering process time and compression/reduction [1].

Although a sequence of complex images or videos slow to be processed by one node, it does not occur if the renderization is done with multiple computers/nodes.

Cluster Computing is an important concept to be used in image and video processing, because it can use many low cost computers to simulate a supercomputer.

Nowadays, a person can have a 3D television in your home and this fact has motivated the entertainment applications to increase 3D images and video modeling. If an image or video complexity can be calculated, it is interesting for many areas such as to have an idea of the final compression size of an image to know the processing time of it.

Sequential images with the moving sensation can form a video, so a video can be defined as a temporal sequence of images or video frames. As stated in ITU-T recommendation P.910 [2], videos can be characterized using the following index: Temporal Information (TI) and Spatial Information (SI). The video complexity can be determined if the video frames complexity is measured and the video temporal information is know.

In this paper, the use of a cluster with a ray tracer of high quality is studied to create stunning three-dimensional graphics. A POV 3D image file format is used because the files have small sizes and they are easy to understand with the description of the shape of each object, the light and other characteristics. The method to measure each video frame complexity is based on [3], which is a simple method that reads the image file and uses an algorithm to indicate the image complexity degree and use the video complexity measurement, considering the spatial and TI complexity of video frames.

This work differs from [2] [3] [4] [5] and [6] because in this paper is used a method to measure video complexity by the association of the spatial complexity and TI of video sequences obtaining a Video Complexity index.

The tests are applied in different videos, of low and high complexity.

This paper is organized as follows. After this Introduction, Section II presents the Theoretical review; Section III describes the Video Complexity Measurement Method associated with the temporal information method implemented in this paper. Section IV presents the experimental results; and finally Section V concludes the paper and proposes some future works.

II. THEORETICAL REVIEW

A. Cluster Computing

Cluster computing means the association of computers that can represent a single supercomputer. So, a number of computers are used together to do a mathematical calculus or a rendering process, the associated computers appear to be a single one, it is considered a cluster. Clusters can be used for High Performance Computing (HPC) [7], where the associated nodes can form a higher computational processing computer.

Beowulf Cluster [8] is academic cluster based on computers with low processing power, on a private network, with GNU/Linux, where the computers are similar and it is used in this paper. The computational processing increases with the number of machines, until reaches a processing limit.

Beowulf cluster is the union of a number of computers (compute nodes) interconnected in a network with the following characteristics:

- The nodes have to be dedicated to the Cluster use and they serve to the same purpose.
- The network is dedicated to the cluster use.
- The nodes with lower processing power can be used in cluster computing.

All de nodes run Open Source software.

B. Image Rendering with POV-Ray

Image rendering is the process to generate an image from a file containing data about cameras, objects, geometry, texture, lighting, and shading data.

Movies, TV, and design visualization, all of them use the rendering process, each employing different features and techniques [9] [10].

Images can be rendered by the software Persistence of Vision Ray tracer (POV-Ray) [11] that is a high-quality, a freeware software, to create three-dimensional graphics [11].

The image contains information about your structure, and during its performing, the software can generate a TGA (Targa) file with use of ray tracing. POVRay reads a file and as output gives a 3D computer image. The textual file contains the type of light, camera position, texture of the object, atmospheric conditions and outlook of the object. POVRay supports primitives such as the iso-surface [12], the polynomial primitive, the julia fractal [13], the super quadratic ellipsoid and the parametric primitive.

It represents objects using mathematical definitions, all POV-Ray primitive objects can be described by mathematical information.

Fig. 1 presents the contents of a file named image1.pov with data of light, camera and the objects (spheres) with theirs color.

POV-Ray works with objects described through spheres, surfaces and cylinders are mathematically. POV-Ray parameters are simple to define. For example, in POV-Ray, a sphere is described by its center and radius.

```
global_settings { radiosity { count 35 distance_maximum 504 }}
background {<1.0, 1.0, 1.0>}
#declare Fin = finish {phong 0.1 phong_size 30 ambient 0.2 diffuse 0.5 reflection 0 }

light source {<576, 432, 100 > color rgb <1, 1, 1> area_light <50,0,0><0,0,50>,5,5
adaptive 1 jitter}
light_source {<288, 216, 252> color rgb <0.5, 0.5, 0.5> area_light <20,0,0><0,20,0>,3,3
adaptive 1 jitter}
camera {location < 288, 216, 252> look_at < 288, 216,0> angle 90}
sphere {<1, 1, -143>,0.87 pigment { rgbf < 0.44, 0.44, 0.42, 0 > } finish { Fin }}
sphere {<4, 1, -164>,0.7 pigment { rgbf < 0.36, 0.36, 0.34, 0 > } finish { Fin }}
sphere {<7, 1, -159>,0.74 pigment { rgbf < 0.38, 0.38, 0.36, 0 > } finish { Fin }}
sphere {<10, 1, -146>,0.84 pigment { rgbf < 0.43, 0.43, 0.41, 0 > } finish { Fin }}
sphere {<13, 1, -135>,0.93 pigment { rgbf < 0.47, 0.47, 0.45, 0 > } finish { Fin }}
sphere {<16, 1, -135>,0.93 pigment { rgbf < 0.47, 0.47, 0.45, 0 > } finish { Fin }}
sphere {<19, 1, -139>,0.9 pigment { rgbf < 0.46, 0.46, 0.44, 0 > } finish { Fin }}
sphere {<22, 1, -134>,0.94 pigment { rgbf < 0.48, 0.48, 0.46, 0 > } finish { Fin }}
sphere {<25, 1, -125>,1.01 pigment { rgbf < 0.52, 0.51, 0.49, 0 > } finish { Fin }}
sphere {<28, 1, -126>,1.0 pigment { rgbf < 0.51, 0.51, 0.49, 0 > } finish { Fin }}
sphere {<31, 1, -126>,1.0 pigment { rgbf < 0.51, 0.51, 0.49, 0 > } finish { Fin }}
```

Figure 1. Content of file image1.pov

C. Ray Tracing

The Ray tracing method generates a 3-dimensional image based on mathematical description. It is used on cluster computing, where can work with high visual quality images [14].

Fig. 2 shows the rays of light from the eye through the image into the scene, when the rays go through the objects of the scene. If the ray misses all objects, then the pixel is shaded with the background color. Ray tracing reads shadows, reflections, and texture in a very easy manner.

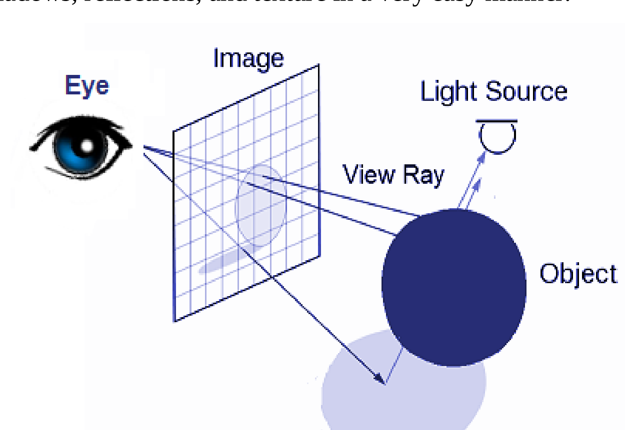


Figure 2. Ray tracing of an object

Ray tracing is so used for interactive and animated rendering.

The traditional problem with Ray tracing has been the computation time, which can be hours or days per image (considering the complexity of the image and the amount of detail desired), and the use of a cluster is useful for saving time in Ray tracing of images.

D. ITU-T recommendation P.910

Recommendation ITU-T P.910 describes non-interactive subjective assessment methods for evaluating video quality.

The P.910 also shows the characteristics of the source sequences, as duration, kind of content, number of sequences, and others.

This Recommendation defines the following terms:

- Gamma: describes the difference between the grey level on a visual display.
- Optimization tests: subjective tests that are typically done during either the development or the standardization of a new system.
- Qualification tests: subjective tests to compare the performance of systems or equipment.
- Spatial Perceptual Information (SI): measure that present, in a picture/video, the number of spatial detail of it. It is higher for more spatially complex scenes.
- Temporal Perceptual Information (TI): measure that present, in a video sequence, the number of temporal changes of it. It is usually higher for high motion sequences.
- Transparency: it shows the performance of a system in relation to an ideal transmission system.
- Replication: repetition of the same conditions for the same subject.
- Reliability of a subjective test:
- a) Intra-individual: refers to the combination about repeated ratings of the subject on the same test condition;
- b) Inter-individual: refers to the combination about different ratings of subject on the same test condition.
- Validity of a subjective test: accordance about the mean value of ratings obtained in a test and the true value which the test is measured.
- Reference Conditions: conditions added to the test conditions in order to anchor the evaluations coming from different experiments.
- Explicit Reference: condition used by the assessors as reference to give their opinion, when the degradation category rating (DCR) method is used.
- Implicit Reference: condition used by the assessors as reference to give their opinion on the test material, when the adaptive controlled routing (ACR) method is used [15].

If is used a small number of test sequences, it is important to choose sequences that reach a large portion of the spatial-temporal information plane. With a sequence test number, one might wish to choose a sequence from each of the number of quadrants of the spatial-temporal information plane.

E. Spatial perceptual information measurement

The spatial perceptual information (SI) is a measurement based on the Sobel filter, that each video frame at time n (Fn) is first filtered with the Sobel filter [Sobel(Fn)]. The standard deviation over the pixels (stdspace) in each Sobel-filtered frame is then computed. This is done for each frame in the video sequence. The maximum value in the time series (maxtime) is chosen to represent the spatial information content of the scene. This process is shown in the following equation:

$$SI = max_{time}\left\{std_{space}\left\lfloor Sobel(Fn)\right\rfloor\right\} \quad (1)$$

F. Temporal perceptual information measurement

The temporal perceptual information (TI) is based upon the motion difference feature, Mn(i, j), which is the difference between the pixel values at the same location in space but at successive times or frames.

The measure of temporal information (TI) is computed as the maximum over time (maxtime) of the standard deviation over space (stdspace) of Mn(i, j) over all i and j.

More motion in adjacent frames will result in higher values of TI.

III. The Proposed Video Complexity Measurement Method

This section describes the video complexity measurement that considers the spatial and TI complexity of video frames.

The variation of frames is represented by TI, considering the values of the pixels, in terms of luminance, located in the same space, and corresponding to consecutive frames of a video in the time domain. The movement function Mn(i,j) is defined and represented by (2).

$$\mathrm{M}_{\mathrm{n(i,j)}} = Q_{n(i,j)} - Q_{n-1(i,j)} \quad (2)$$

Where Qn(i,j) is the pixel in the ith row and jth column of the video frame at the nth time.

The TI index is calculated as the mean value corresponding to the standard deviations for all the Mn (i,j), as shown in (3).

$$\mathrm{TI} = Mean\{std[M_{n(i,j)}]\} \quad (3)$$

The complexity of a video frame can be defined by its background homogeneity and foreground clutter. An image with a high quantity of foreground clutter is considered more complex.

In this paper, the spatial complexity of each video frame is determined using an algorithm based on the image file information. This algorithm uses as input, the parameters of form, size of the object, texture, pigments and intersections between objects. As a result, a Global Spatial Complexity (GSC) index is defined, and it is mean of the spatial complexity of each frame.

Finally, considering the normalized values of GSC and TI indexes, the Video Complexity Index (VCI) is defined by (4).

$$\mathrm{VCI} = \| TI \| + \| GSC \| \tag{4}$$

IV. THE EXPERIMENTAL RESULTS

Seven different videos were analyzed, each one lasted 30 seconds, three videos were simpler and four others were more complex; the videos were chosen in agreement with the rendering time in Blender software, the more the video took to render the more it was considered complex. The frames were analyzed one by one, generating the images that will be analyzed by the PovRay software that works only with .POV file extension. In order to generate the .POV files was used tools that translate between several 3D file formats including and that converts animation files from Anim8or (.an8), MilkShape3d (.ms3d and .txt), ASE (.ase), AC3D (.ac) to PovRay files. It is used a free software to generate the videos (.avi). The PovRay software was installed on Linux environment.

When the POV file is generated, the image rendering process has been performed eight times to reach results with low standard deviation.

In the experimental work, nine computers core i5 were used, with GNU/Linux, RAM memory of 4 GB, where one node was selected as master node and more eight computers were slaves, so the processing occurs in the eight nodes and the master one only distribute the job between the slaves. The Beowulf cluster was used in this experiment, where the nodes are connected via an Ethernet switch equipment of 100 Mbps. It was used the Parallel Virtual Machine (PVM) and the Message Passing Interface (MPI) through a library on Linux, a folder is shared via Network File System (NFS) where the videos are allocated and is used for all nodes to collect and render the videos.

A bash script was implemented based on the VCI index defined in (4). This script classifies the video complexity levels from 0 to 3, in which level 0 indicates the lowest video complexity and the level 3 indicates a video with the highest complexity level.

The relation between the level complexity and the cluster nodes number is shown in the Table 1. So, the greater the complexity of the scene, more cluster nodes is necessary for rendering the image.

The methodology used to obtain this relation was through experiments using different videos and nodes cluster.

Fig. 3 shows the first frame of a video with a low complexity level used as test material, named Smile with teeth. The TI index is low, because there is no drastic variation between consecutive frames. The GSC is also low because of the simple object, its texture, shadows, and black background and also generated with resolution of 640 x 480 pixels. Indeed, the video complexity level obtained by the bash script for the video was 1. Therefore, its processing time consumption in a cluster was relatively low.

TABLE I. LEVEL COMPLEXITY AND NODES NUMBER

Level Complexity	**Nodes Number**
3	08
2	05
1	03
0	01

It was done an experiment using TI and GSC alone and the results were of 1 for TI and 0 to GSC. But the value of 0 of GSC do not represent the true value, because is necessary more than one node to render the video.

Figure 3. Simple frames of the videos named Smile with teeth

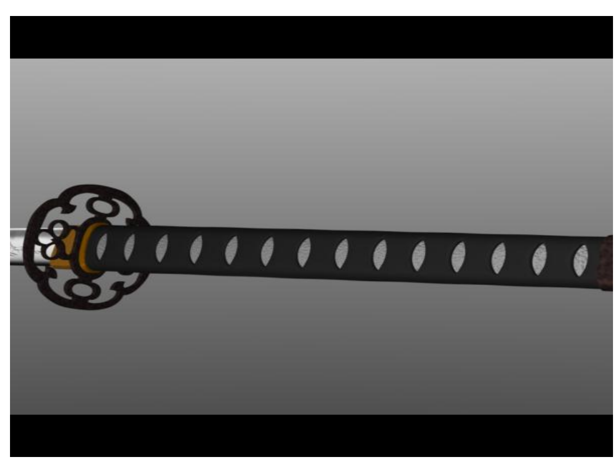


Figure 4. Simple frames of the videos named Sword

Fig. 4 presents a frame of a video with a low complexity level too, named Sword. The TI indexes are low. The GSC is also low because of the simple object, simple colors, its texture, shadows, and simple background and also generated with resolution of 640 x 480 pixels. Indeed, the video complexity level obtained by the bash script for the video was 0. Its processing time consumption in a cluster was relatively low. This was the simplest frame used in the tests, obtained minimal complexity level.

Figure 5. Complex frame of the video named Elephants Dream

Fig. 5 shows an image with a high complexity level, corresponding to the frame of a video named Elephants Dream. The processing time consumed is also high because of its texture, shadows, and the two personages. Also, the image was generated in full 1920×1080 HD resolution. Indeed, the complexity level obtained by the bash script was 2.

It was done an experiment using TI and GSC alone and the results were of 1 for TI and 1 to GSC. But the value of 1 of both do not represent the true value, because is necessary more than three nodes to render the video.

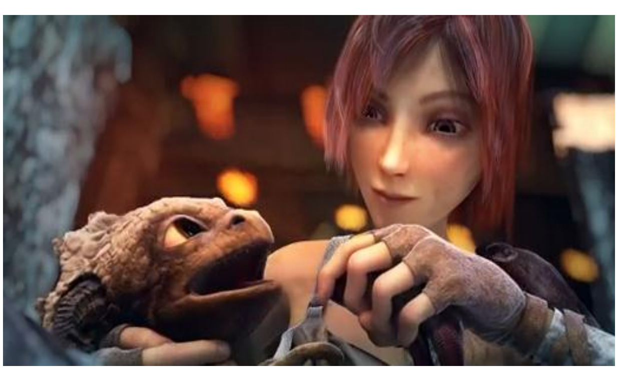
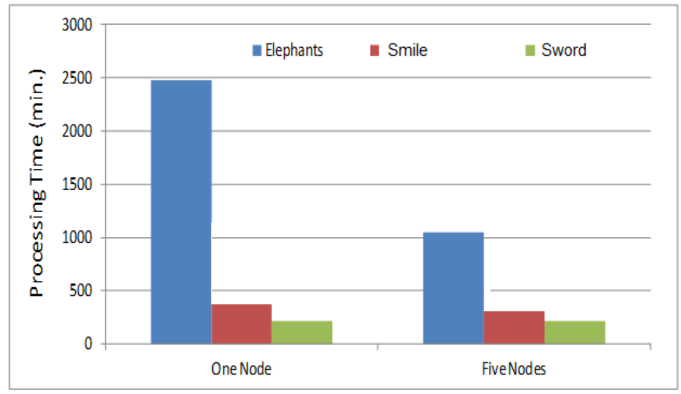


Figure 6. Complex frame of the video named Sintel

Fig. 6 presents an image with a high complexity level. The complexity level obtained by the bash script was 3.

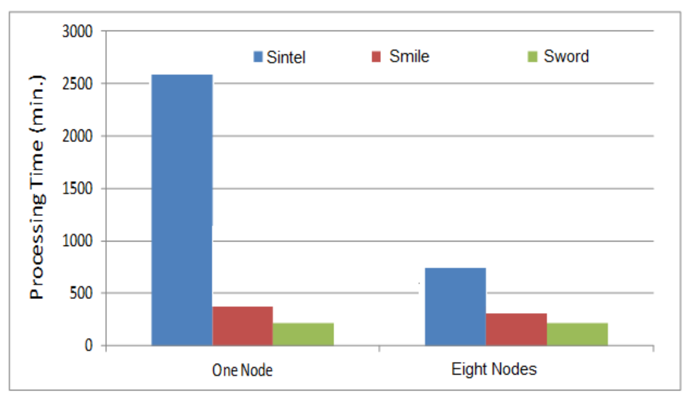
Fig. 7 shows the processing time consumed for rendering the videos presented before using 1 and 5 nodes. It can be observed that if the video complexity is high, the number of cluster nodes need to be increased to get a better time. Conversely, for low complexity videos, one node is enough, that is, it is not necessary to use a cluster.

Fig. 8 shows the processing time using 1 and 8 nodes, the time to rendering the video named Sword has increased slightly, because its low complexity. So, in case of low complexity the time to render the images on multiple nodes can cause a delay time of full rendering.



Processing Time consumed for 1 and 5 cluster nodes

The video named Sintel presents a complexity level higher than the video named Elephants, the increased for 8 nodes shows the reducing in the processing time of Sintel more than the processing time of Elephants with five nodes that appears in Fig. 8.



Processing Time consumed for 1 and 8 cluster nodes

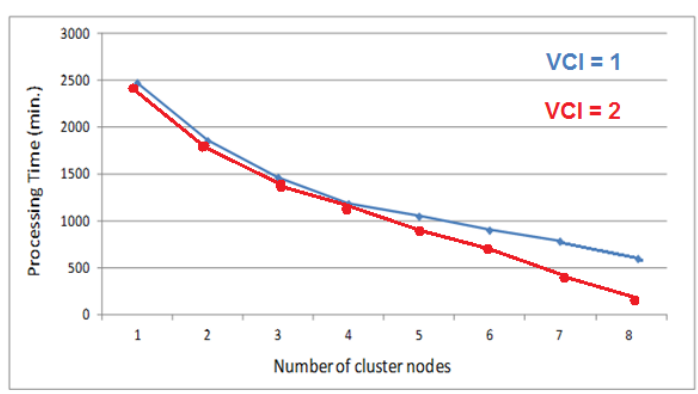


Figure 7. Processing Time consumed for different number of cluster nodes

Fig. 9 shows the processing time consumed to render a video with a VCI equal to 1, shown in blue line, and presents a VCI equal to 2, shown in red line, using different number of nodes. The results show that a VCI equal to 2 needs more nodes to be rendered, generating a higher decrease in rendering time than the VCI equal 1.

V. Conclusions

In this paper it is presented the VCI that takes into account the spatial and temporal characteristics of the video. Experimental results show that VCI metric can successfully classify videos according their complexity.

In the experiments the TI and GSC was calculated together because both helps to find the Video complexity, and consequently could identify the right number of cluster nodes needed for each video. If only the TI or the GSC is calculated alone, it will not be considered the time or the spatial variation, and a video is the association of both parameters, that is, a spatial pixel variation is different from a variation in a given time.

The Blender software that performs 3D modeling and animation was used to render the test videos, and through this software is possible to measure the video complexity level, the results with the proposed metric in this work have a relationship with time spent to render the videos through Blender, and so this was used to validate the results. But the time to render a video through the Blender in one single computer can take hours.

VCI metric can be used to determine the right number of nodes that are necessary to process videos based on their complexity in a fast way. Low complexity videos can be processed using only one node and regardless the number of cluster nodes used, the time to render a low-complexity video does not decrease; high complexity videos require more cluster nodes.

Acknowledgment

The authors thank the University of São Paulo by the motivation to researches in the area of Computer and Telecommunication Systems.

References

[1] J. S. Kaleka, "Comparative Performance Analysis of Haar, Symlets and Bior wavelets on Image compression using Discrete Wavelet Transform". International Journal of Computers & Distributed Systems, [S.l.], v. 1, n. 2, p. 11-16, aug. 2012.

[2] ITU-T, "Subjective Video Quality Assessment Methods for Multimedia Applications", Tech. Rec. P.910, Geneva, Abr. 2008.

[3] D. Z. Rodríguez , R. L. Rosa, E. Alfaia, "A Simple Method to Measure the Image Complexity on a Fault Tolerant Cluster Computing", in Proc of AICT, pp.549-554, Barcelona, Spain, 2010.

[4] S. Skala and N. Pavkovic, "Image Programming for Scientific Visualization by Cluster Computing", in Proc. of Autonomic and Autonomous Systems and International Conference on Networking and Services, vol. 23, pp. 93, October, 2005.

[5] M. I. C. Murguía, and A. C. Sáenz, "A Fuzzy Approach on Image Complexity Measure", Revistas Científicas de América Latina y el Caribe, España y Portugal, 2007.

[6] M. Chacon, D. Alma, S. l Corra, "Image complexity measure: a human criterion free approach", in: Proc. NAFIPS 2005, 2005, pp. 241--246.

[7] C. Severance and K. Dowd, "High Performance Computing", Oreilly, July, 1998.

[8] *Beowulf Cluster*, http://www.beowulf.org, 12.10.2013.

[9] K. Miller, D. Przybyla, and M. Pegah, "*Soothing the* many-headed beast: improving the rendering process in the *world of 3D animation"*, Proceedings of the 32nd annual ACM SIGUCCS conference on User services, 2004, pp. 381-384, New York, USA.

[10] N. Li and Z. Huang, "Feature-guided painterly image *rendering"*, International Conference on Image Processing, pp. 653-656, v. 1, 2002.

[11] *PovRAY*, http://www.povray.org, 12.10.2013.

[12] P. Thevenaz and M. Unser, "Precision Isosurface Rendering of 3-D Image Data", IEEE Transactions on Image Processing, vol. 12, no.7, July, 2003.

[13] H. Zhang, J. Hu, G. Wang and Y. Zhang, "A Steganography Scheme Based on Fractal Images", Second International Conference on Networking and Distributed Computing (ICNDC), pp. 28-31, 2011.

[14] T. Schiffer, "Ray Tracing: Lessons Learned And Future Challenges". IEEE Potentials, vol. 32, pp. 34-37, 2013.

[15] J. Regnier, F. Bedard, J. Choquette, A. Caron, "Dynamically controlled routing in networks with non-DCR-compliant switches", Communications Magazine, IEEE , vol.33, no.7, pp.48,52, Jul 1995.

Demóstenes Zegarra Rodríguez received his B.S. degree in Electronic Engineering from the Pontifical Catholic University of Peru and the M.S. degree from University of São Paulo - USP (2009). Nowadays, he is a Ph.D. student in Electronic Systems at USP, with solid knowledge in Telecommunication Systems and Computer Science based on 12 years of professional experience in important companies. His current interest includes, QoS and OoE in Multimedia services, Digital TV and architect solutions in Telecommunication Systems.

Renata Lopes Rosa received her B.S. degree in Science Computer from UNIFEI, Brazil, and the M.S. degree from University of São Paulo - USP (2009). She is a Ph.D. student at Escola Politécnica of the University of Sao Paulo (EPUSP). Her current research interest includes computer networks, quality of experience of multimedia service, social networks and recommendation systems.

Graça Bressan was granted her PhD in Electronic Engineering (1986) by Escola Politécnica of the University of São Paulo (EPUSP). Her current research interests include computer networks and digital television focusing on aspects of distributed systems, distributed middleware, QoS mechanisms, collaborative virtual environment, middleware for Digital TV, interactive digital TV, video-conferencing, modeling and performance analysis of networks, and applications in distance education.