

INFORMATION THEORY ASSISTED DATA VISUALIZATION AND
EXPLORATION

by

EKREM SERIN

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

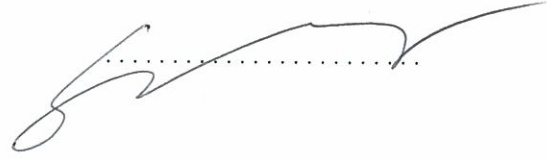
Sabanci University

January 2012

INFORMATION THEORY ASSISTED DATA VISUALIZATION AND
EXPLORATION

APPROVED BY:

Assist.Prof.Dr.Selim Balcısoy
(Dissertation Supervisor)



Assoc.Prof.Dr.Berrin Yanıkođlu



Prof.Dr.Mustafa Ünel



Prof.Dr.Tanju Erdem



Assoc.Prof.Dr.Yücel Saygın



DATE OF APPROVAL: 18.01.2012

© EKREM SERIN 2012
All Rights Reserved

Abstract

This thesis introduces techniques to utilize information theory, particularly entropy for enhancing data visualization and exploration. The ultimate goal with this work is to enable users to perceive as much as information available for recognizing objects, detecting regular or non-regular patterns and reducing user effort while executing the required tasks.

We believe that the metrics to be set for enhancing computer generated visualizations should be quantifiable and that quantification should measure the information perception of the user. The proper way to solve this problem is utilizing information theory, particularly entropy. Entropy offers quantification of the information amount in a general communication system. In the communication model, information sender and information receiver are connected with a channel. We are inspired from this model and exploited it in a different way, namely we set the information sender as the data to be visualized, the information receiver as the viewer and the communication channel as the screen where the visualized image is displayed. In this thesis we explore the usage of entropy in three different visualization problems,

- Enhancing the visualization of large scale social networks for better perception,
- Finding the best representational images of a 3D object to visually inspect with minimal loss of information,
- Automatic navigation over a 3D terrain with minimal loss of information.

Visualization of large scale social networks is still a major challenge for information visualization researchers. When a thousand nodes are displayed on the screen with the lack of coloring, sizing and filtering mechanisms, the users generally do not perceive much on the first look. They usually use pointing devices or keyboard for zooming and panning to find the information that they are looking for. With this thesis we tried to present a visualization approach that uses coloring, sizing and filtering to help the users recognize the presented information.

The second problem that we tried to tackle is finding the best representational images of 3D models. This problem is highly subjective in cognitive manner. The best or good definitions do not depend on any metric or any quantification, furthermore, when the same image is presented to two different users it can be identified differently. However in this thesis we tried to map some metrics to best or good definitions for representational images, such as showing the maximum faces, maximum saliency or combination of both in an image.

The third problem that we tried to find a solution is automatic terrain navigation with minimal loss of information. The information to be quantified on this problem is taken as the surface visibility of a terrain. However the visibility problem is changed with the heuristic that users generally focus on city centers, buildings and interesting points during terrain exploration. In order to improve the information amount at the time of navigation, we should focus on those areas. Hence we employed the road network data, and set the heuristic that intersections of road network segments are the residential places. In this problem, region extraction using road network data, viewpoint entropy for camera positions, and automatic camera path generation methods are investigated.

Özet

Bu tez bilgi kuramı, özellikle entropiden yararlanarak veri görselleştirmesi ve araştırmasını geliştirecek teknikleri tanıtmaktadır. Tez ile nesnelere tanımda, olağan veya olağan dışı örüntülerin tespit edilmesinde ve gerekli görevlerin yerine getirilmesinde kullanıcıların mevcut olan en fazla bilgiyi algılamalarını sağlamak amaçlanmıştır.

Bilgisayar tarafından üretilmiş görselleştirmelerin iyileştirilmesinde konulmaya çalışılan metriklerin sayısallaştırılabilir olması ve sayısallaştırmanın algılanan bilgi miktarını ölçmesi gerektiğine inanılmaktadır. Bu problemi çözmenin uygun yolu, bilgi kuramı, özellikle entropiden yararlanmaktır. Çünkü entropi genel iletişim sistemindeki bilgi miktarını ölçmeyi önermektedir. İletişim modelinde, bilgi göndericisi, bilgi alıcısı ile bir kanal vasıtasıyla bağlantı halindedir. Bu modelden esinlenerek, farklı bir yaklaşım ile bilgi göndericisi görselleştirilmeye çalışılan bilgi, bilgi alıcısı izleyici, kanal ise görselleştirmenin sunulduğu ekran olarak değerlendirilmiştir. Bu tezde, entropinin üç farklı görselleştirme probleminde kullanılabilirliği araştırılmıştır,

- Büyük çaplı sosyal ağların algılanmasını iyileştirebilmek amacıyla görsellenmesi,
- 3B'lu nesnelere görsel olarak en az bilgi kaybı ile incelemek amacıyla en iyi temsili resimlerin bulunması,
- Arazi üzerinde en az bilgi kaybı ile otomatik gezinme.

Büyük çaplı sosyal ağların görselleştirilmesi bilgi görselleştirmesi araştırmacıları için halen önemli bir sorundur. Bin düğümün renk, boyutlandırma, ve filtreleme mekanizmalarından yoksun olarak sergilenmesi durumunda, kullanıcılar ilk bakışta pek birşey algılayamazlar. Genellikle aradıkları bilgiye ulaşabilmek amacıyla işaret cihazları veya klavyeyi yakınlaştırma ve kaydırma amacıyla kullanırlar. Bu tez ile kullanıcılara sunulan bilginin algılanabilmesini sağlayabilmek amacıyla filtreleme, renklendirme ve boyutlandırmayı kullanan bir görselleştirme yaklaşımı sunulmaya çalışılmıştır.

İkinci olarak 3B'lu modellerin en iyi temsili resimlerinin bulunması problemi çözülmeye çalışılmıştır. Bu problem bilişsel anlamda öznedir. İyi ve en iyi tanımlamaları herhangi bir metrik ve sayısallaştırmaya dayanmamaktadır, ayrıca aynı

resmin iki farklı kullanıcıya sergilenmesi durumunda farklı olarak değerlendirilmesi olasıdır. Bu çalışmada, iyi ve en iyi tanımlamaları temsili resimler için en fazla nesne yüzü, en fazla belirgenlik veya bunların kombinasyonu şeklinde eşlelemeye çalışılmıştır.

Çözüm bulmaya çalışılan üçüncü problem ise en az bilgi kaybı ile otomatik arazi gezinmesidir. Burada arazinin yüzey görünürlüğü sayısallaştırılmaya çalışılmıştır. Ancak görünürlük problemi, kullanıcıların genellikle şehir merkezlerine, binalara ve ilginç noktalara odaklandıkları buluşsalı ile farklı hale getirilmiştir. Gezinme esnasındaki bilgi miktarını artırabilmek amacıyla, anılan alanlara yoğunlaştırılması hedeflenmiştir. Bu maksatla, yol bilgisinden yararlanılmış ve yol kesişimlerinin iskan bölgeleri olabileceği buluşsalı ortaya konulmuştur. Bu problemde, yol bilgilerinden alan çıkarımı, kamera noktaları için bakış noktası entropisi ve otomatik rota üretimi metodları araştırılmıştır.

Acknowledgements

I would like to express my sincere gratitude to my advisor Dr. Selim Balcısoy for the continuous support of my PhD study and research. Without his patience, motivation, enthusiasm, and deep knowledge this thesis would not come true. His guidance helped me in all steps of this research and writing of this thesis.

Besides my advisor, I would like to thank internal and external committee of my dissertation: Prof. Mustafa Ünel, Prof. Tanju Erdem, Assoc.Prof. Berrin Yanıkođlu and Assoc.Prof. Yücel Saygın for their insightful comments.

Further, I would like to express my sincere thanks for the endless moral support from my colleagues, Cihat Eryiđit, Tolga Önel, Serhat Özener, Selçuk Öztürk, and Barıř Aktop.

Finally, I would like to heartily thank my wife, Sibel, whose constant encouragement, patience, and support that I have relied throughout the period of my research.

This research is partially supported by Turkish Scientific and Technological Research Council (TUBITAK) research grant 109E022.

TABLE OF CONTENTS

Abstract	iv
Acknowledgements	viii
List of Tables	xii
List of Figures	xiii
1 Introduction	1
1.1 Problem Statement	4
1.2 Contribution	5
1.3 Thesis Structure	5
2 Overview on Information Visualization	7
2.1 Introduction	7
2.2 What Is Good Visual Representation ?	8
2.3 History of Information Visualization	8
2.4 Information Visualization Techniques	11
2.4.1 Graph Drawing	11
2.4.2 TreeMap	13
2.4.3 HeatMap	13
2.4.4 Parallel Coordinates	16
2.4.5 Flowmap	16
2.5 Information Visualization Problems	16
3 Literature Survey	19
3.1 Social Network Analysis and Visualization	19
3.2 Viewpoint Generation	20
3.3 Camera Control	22
4 Sensitivity Analysis and Visualization of Social Networks	25

4.1	System Overview	26
4.2	Social Network Centralities	27
4.2.1	Degree Centrality	27
4.2.2	Betweenness Centrality	28
4.2.3	Closeness Centrality	29
4.3	Sensitivity Analysis	30
4.3.1	Degree Entropy	30
4.3.2	Betweenness Entropy	31
4.3.3	Closeness Entropy	32
4.3.4	Combined Approach	32
4.4	Discussion and Visualization	33
4.4.1	Discussion	34
4.4.2	2D Visualization of Social Network	36
4.5	Conclusion	40
5	Object Exploration	43
5.1	Viewpoint Entropy	43
5.2	Mesh Saliency and Mean Curvature	45
5.3	Information Coverage	48
5.3.1	Greedy N-Best View Selection	49
5.3.2	Viewpoint Mesh Saliency Entropy	54
5.3.3	Combined Approach	57
5.4	Results and Statistical Output	57
5.5	Usability Study	63
5.6	Conclusion	64
6	Automated Terrain Navigation	66
6.1	Scene Analysis and Path Generation	67

6.1.1	Region Extraction	67
6.1.2	Terrain Rendering	68
6.1.3	Best Viewpoints	70
6.2	Camera Path Planning	72
6.2.1	Traveling Salesman Problem	72
6.2.2	Path Planning for Intra-Regions	74
6.2.3	Path Planning for Inter-Regions	76
6.2.4	Final Camera Trajectory	76
6.3	Tour Presentation in Google Earth	77
6.3.1	Camera in KML	77
6.3.2	Tour Generation in KML	78
6.3.3	Importing the Tour Using Google Earth API	79
6.4	Results	79
6.5	Conclusion	80
7	Conclusion	85
7.1	Future Work	86

List of Tables

4.1	Centrality and sensitivity entropy values for the example network. Cent. denotes the centrality and Ent. denotes the entropy sensitivity analysis. Note that difference between columns shows the change reflected by sensitivity analysis, and the difference between rows highlight the ratio emphasized.	34
5.1	Cumulative face coverage contribution ratio of the viewpoints for different models using Greedy N-Best View Selection and taking surface area entropy into account.	60
5.2	Cumulative face coverage contribution ratio of the viewpoints for different models using Greedy N-Best View Selection and taking combined entropy into account.	62
5.3	Cumulative saliency coverage contribution ratio of the viewpoints for different models using Greedy N-Best View Selection.	62
6.1	Timing for non-optimized application. Note that all the values are total duration of the corresponding steps.	80

List of Figures

2.1	The continuum of understanding, according to Nathan Shedroff [1]. . .	7
2.2	Napoleon’s Russian Campaign of 1812	9
2.3	Playfair’s chart	10
2.4	Florence Nightingale’s rose diagram.	10
2.5	The 1854 London Cholera Epidemic	11
2.6	(a)Tube map of 1908, (b)The modern tube map, based on the simplified topological design invented by Beck.	12
2.7	A Periodic Table Of Visualization Methods [2]	12
2.8	A Co-citation Map of Graph Drawing Articles(1990-2003) [3]	13
2.9	The Flare Dependency Graph is a ring-based layout showing the dependencies between classes in the Flare library [4]	14
2.10	Treemap of soft drink preference in a small group of people.	14
2.11	Heatmap Visualization	15
2.12	Parallel coordinates for 730 elements with 7 variant attributes [1]. . .	16
2.13	Flowmap: Outgoing Migration Map from Colorado for 1995-2000 [5] .	17
4.1	Social Network Visualization System Overview	26
4.2	Hue (from red=0 to blue=max) shows the node betweenness.	29
4.3	An example social network	34
4.4	Node size mapping (a)Degree centrality, (b)Degree entropy sensitivity analysis	35
4.5	Node size mapping (a)Betweenness centrality, (b)Betweenness entropy sensitivity analysis	36
4.6	Node size mapping (a)Closeness centrality, (b)Closeness entropy sensitivity analysis	37

4.7	Social network visualizing combined information	37
4.8	Default presentation of collaboration network	38
4.9	Collaboration network visualized using degree centrality	39
4.10	Collaboration network visualized using key actor discovery	40
4.11	Collaboration network visualized using sensitivity analysis of degree entropy	41
4.12	Collaboration network visualized using sensitivity analysis of betweenness entropy	42
4.13	Collaboration network visualized using sensitivity analysis of closeness entropy	42
5.1	Hand model shown with unique colors for each face, used for viewpoint entropy calculations. Four of initial camera points are also presented.	45
5.2	Surface normal, tangent plane and principal curvatures of the surface.	46
5.3	Greedy approach for best view selection. <i>CF</i> stands for covered faces, <i>E</i> for entropy and <i>Lat-Lon</i> for latitude and longitudes over the sphere. Three dots show the continuous call of the algorithm till the termination. In the initial step algorithm is called with empty set, hence \emptyset . In the following steps <i>CF</i> includes all faces covered so far.	50
5.4	Teapot is displayed with five viewpoints using the approach from [6] and [7] compared to our greedy method. Images (a)-(e) cover 813 faces of total 2256 faces. However our method shown in (f)-(j) covers 2200 faces with provided views.	51
5.5	Stanford Bunny is displayed with five viewpoints using the approach from [6] and [7] compared to our greedy method. Images (a)-(e) cover 63748 faces of total 69743 faces. However our method shown in (f)-(j) covers 68674 faces with provided views.	52
5.6	Armadillo is displayed with five viewpoints using the approach from [6] and [7] compared to our greedy method. Images (a)-(e) cover 20103 faces of total 50000 faces. However our method shown in (f)-(j) covers 42009 faces with provided views.	52

5.7	Dragon model is displayed with five viewpoints using the approach from [6] and [7] compared to our greedy method. Images (a)-(e) cover 36965 faces of total 49755 faces. However our method shown in (f)-(j) covers 41911 faces with provided views.	53
5.8	Hand model is displayed with five viewpoints using the approach from [6] and [7] compared to our greedy method. Images (a)-(e) cover 8976 faces of total 18905 faces. However our method shown in (f)-(j) covers 18406 faces with provided views.	53
5.9	Mesh saliency for a hand model shown in (a). HSV color model shown in (b) is used to mark the saliency of the vertices. Hot colors(red) $Hue=0$ shows the highest saliency, and $Hue=240$ for the lowest. <i>Saturation</i> and <i>Value</i> are kept fixed in distribution.	55
5.10	An example of triangulated surface for vertex to face saliency distribution.	56
5.11	Hand shown from five viewpoints using face area maximization maximization (a)-(e), saliency coverage maximization (f)-(j) and combined approach (k)-(o). For each approach the figures are ordered from the most contribution to the least.	59
5.12	Heart shown from five viewpoints using face coverage maximization (a)-(e), saliency coverage maximization (f)-(j) and combined approach (k)-(o). For each approach the figures are ordered from the most contribution to the least.	60
5.13	Brain shown from five viewpoints using face coverage maximization (a)-(e), saliency coverage maximization (f)-(j) and combined approach (k)-(o). For each approach the figures are ordered from the most contribution to the least.	61
5.14	Dragon shown from five viewpoints using face coverage maximization (a)-(e), saliency coverage maximization (f)-(j) and combined approach (k)-(o). For each approach the figures are ordered from the most contribution to the least.	61
5.15	Stanford Bunny shown from five viewpoints using the combined approach i.e face and salient point coverage are maximized. The figures are ordered from the most (a) contribution to the least (e).	62
5.16	Hand model shown with red spheres used for visually queueing user selected points.	63

5.17	Mean saliency of the user selected points and surface saliency mean are shown. The surface saliency mean is denoted by the red circle and user values with blue. Note that the user selected points are higher than the surface mean which does not contradict with the knowledge in literature about user tendencies for salient points.	64
6.1	An automatically generated path by our algorithm for San Francisco shown in Google Earth framework.	67
6.2	Road network and terrain data	68
6.3	The region extraction algorithm steps are visualized. In (a) An example road network is shown, (b) Intersection points are marked with red square. In (c) the result of convex hull determination algorithm is presented. The extracted bounding circle is shown in (d)	69
6.4	Sketch of uniquely colored texture mapping to a grid	70
6.5	Sketch of CLOD algorithm on a grid. Camera is shown with a turquoise circle. Note that camera move changes the triangulation.	70
6.6	Wireframe mode for a region of terrain is shown in (a). When the camera gets closer vertex popping phenomenon occurs. In (b) the uniquely colored texturing is applied to the elevation data	71
6.7	In (a), extracted regions in San Francisco are shown by circles using Google Maps. With the aid of these regions, path is generated on the terrain(b)	82
6.8	Sketch for the generated path. Note that intra-region camera path resembles circles however not exact, they are on sphere and the connection between them is an arc. Straight lines show the path for inter-regions, however the start and finish points may not be on the same plane.	83
6.9	Extracted regions are presented by the spheres using Google Earth framework.	83
6.10	Inter-region tour shown with connecting lines using Google Maps framework. Placemarks represent the region centers.	84
6.11	Heights(m) of viewpoints in first 3 regions for the path generated on San Francisco.	84

1 INTRODUCTION

Information Visualization is a wide research area and there will always be a need for visualizing information as the information continued to be produced. The purpose of information visualization is to convey useful and helpful information to the user where it can ease the tasks that users do on daily basis, which is considered as one of the aims of computers in general.

In this research we tried to exercise and find metrics to enhance the computer generated visualizations where the established metrics are used to form the basis for color, and size of objects visualized on the screen as well as to find good camera positions for improving the perception of the displayed image.

We believe that the metrics to be set for enhancing computer generated visualizations should be quantifiable and that quantification should measure the information perception of the user. The proper way to solve this problem is to utilize information theory, particularly entropy for enhancing data visualization and exploration. Shannon's entropy model offers the quantification of the information amount for a general communication system [8]. In that model there are information sender and information receiver connected with a communication channel. We are inspired from this model and exploited it in a different way; we set the information sender as the data to be visualized, the information receiver as the viewer and the communication channel as the screen where the visualized image is displayed.

We studied the usage of entropy for improving visualizations in three different problem domains,

- Enhancing the visualization of large scale social networks for better perception,
- Finding the best representational images of a 3D object to visually inspect with minimal loss of information,

- Automatic navigation over a 3D terrain with minimal loss of information.

The first problem that we tried to present a solution is analysis and visualization of large scale social networks, which is still a major challenge for researchers. When a thousand nodes are displayed on the screen with the lack of coloring, sizing and filtering mechanisms, the users generally do not perceive much on the first look. They usually use pointing devices or keyboard for zooming and panning to find the information that they are looking for. With this thesis we tried to present a visualization approach that uses coloring, sizing and filtering to help the users recognize the presented information.

In our approach social network is considered as a communication system and the entropy change of the system by actor removal using centrality measures such as degree, betweenness and closeness is employed. We provided a visualization system where a conventional node-link diagram is used. However the node-link diagram is enhanced by the means of size, and colors of actor representations where they are mapped from conducted analyses.

The social network used in this work is a scientific collaboration network extracted from DBLP [9] database including submissions for IEEE Transactions on Visualization and Computer Graphics(TVCG) between 2005-2009. We conducted sensitivity analysis for the collaboration network using degree, betweenness and closeness entropies. In order to present the aggregate or combined entropy change, each centrality measure entropy vector is normalized before combination process. Key actor discovery [10] is also integrated into the application. The visualization system provided with this work exploits the centrality, the centrality entropy and aggregate entropy change measures to differentiate the actors using the sizing technique. Furthermore, the color information is utilized to convey the groups and subnetworks information using graph clustering analysis.

The second problem that we tried to tackle is to find the best representational images of 3D models, where the images are generated by the help of camera control in 3D object exploration context. These concepts have been actively studied in

recent years [11, 12, 13, 14] and have applications in many areas including medical analysis and training, robotics, image based rendering, virtual reality and scientific visualizations. Finding the best representational images of 3D objects is a highly subjective problem in cognitive manner and the “best” or “good” definitions do not depend on any metric. However in this thesis we tried to map some metrics to those definitions for representational images.

Representational images of 3D objects are created by projecting their surfaces onto the screen or any artificial plane. The projection process depends on parameters such as camera position, camera vector, up vector, and clipping plane positions. We tried to find such camera positions that the 3D object is projected in “good” or “best” way where those subjective definitions are mapped to *Information Theoretical* measure. Information Theory helped us to quantify two displayed information of a model, the faces of the model and its salient features. In this work Viewpoint Entropy introduced by Vazquez et al. [6] is employed and *Mesh Saliency Entropy* is presented as novel view descriptor. Viewpoint Entropy is an information theoretical measure which is used to determine the amount of information from a viewpoint using the projected faces of the model. The newly introduced view descriptor, *Mesh Saliency Entropy* depends on the idea by Lee et al. [15], is an information of regional importance which is considered as the salient feature of the model or the graphics meshes. We map the good or best definition as a camera position where the perception of two defined information is maximized. The maximization approach is done by our *Greedy N-Best View Selection* algorithm which creates a viewing sphere around the object explored and tries to find a camera point where the viewer can receive the maximum information amount. The details about the techniques and algorithms introduced with this work will be presented in subsequent chapters.

The third problem that we tried to come upon with a solution is the automatic terrain navigation with minimal loss of information. Automatic navigation requires camera control methods which is still a challenging task that includes viewpoint calculation, path planning and editing necessities. An excellent survey by Christie

et al. [16] explains the motivation and methods of camera control in virtual space. Although the methods are developed to solve the requirements of different domains, they share common problems and difficulties such as degrees of freedom, computation complexity and lack of generic measures. Camera control techniques vary from user input reaction based ones to fully automated controls. The approaches and the techniques presented do not provide a solution for a camera control in large terrain dataset.

In this work we propose a novel technique to control the camera for large terrain dataset visualization where the calculated viewpoints can be used as initial starting points for navigation. The proposed camera point set contains the best views in the extracted subregions and the framework can be integrated into 3D game engines or urban visualization systems to give quick glimpse or tour of the environment for the users.

Our proposed navigation in virtual space depends on information and a measure to quantify it. The information on navigation problem is taken as the surface visibility of the terrain. However the surface visibility problem is changed with the heuristic that users generally focus on cities, buildings and interesting points during terrain exploration. In order to improve the perceived information amount, we employed road network data, where we set a heuristic that intersection of road network segments are the residential places and we should focus on those areas at the time of navigation. Here we borrow the concept of *Viewpoint Entropy* and use our *Greedy N-Best View Selection* technique for descriptive and informative view determination in sub-regions of the terrain surface. In order to connect the calculated viewpoints an evolutionary programming approach is used where a single objective function, i.e. distance, is minimized.

1.1 Problem Statement

The aim of a visualization is conveying some helpful information to one that looks at it. We tried to develop presentation systems to show useful and meaningful infor-

mation to the user with this work. Here we exploited Shannon’s entropy model as information amount inferred from a system, and tried to improve the user perception by conducting visibility, saliency and sensitivity analyses. We believe Shannon’s entropy model is a promising way to solve view related problems by providing a measure to *quantify* the information on the communication channel between the user and the visual world.

1.2 Contribution

Our contributions in this thesis are,

- A novel approach for the sensitivity analysis of a social network and a visualization system that conveys the quantified information,
- An efficient greedy choice algorithm that selects high coverage of 3D object faces (N-Best View Selection),
- Introduction of a novel view descriptor called *Mesh Saliency Entropy*, and combining viewpoint and mesh saliency entropies in view selection for minimal loss of information,
- Conducting visibility analysis of large scale terrain using road network data and employment of evolutionary programming approach for camera path generation.

1.3 Thesis Structure

The remainder of this thesis is structured as follows :

- **Chapter 2:** Gives a quick overview about the techniques and metaphors used in Information Visualization. It presents the history, and the state of art approaches and discusses the issues and problems that we are facing in visualization systems.

- **Chapter 3:** Presents a review about social network analysis and visualization, view descriptors used for object exploration, and methods for camera control in virtual environments.
- **Chapter 4:** Techniques to analyze and visualize a social network using Shannon's entropy model are discussed in this chapter. Degree, betweenness and closeness entropy measures are introduced to conduct network sensitivity analysis. A visualization application where the social network is displayed using sizing, filtering and colorization to improve the perception is presented.
- **Chapter 5:** This chapter introduces our novel view descriptor entitled *Mesh Saliency Entropy* and a greedy choice algorithm for selecting high coverage of faces, high coverage of mesh saliency and high coverage of combined information.
- **Chapter 6:** Navigation in 3D terrain is discussed in this chapter. Camera control techniques, region extraction from road network data, viewpoint generation, connecting the viewpoints using evolutionary programming approach, and integrating the generated path to Google Earth framework are detailed.
- **Chapter 7:** Concluding remarks and future work are discussed in this chapter.

2 OVERVIEW ON INFORMATION VISUALIZATION

2.1 Introduction

Information visualization is the interdisciplinary study of “the visual representation of large-scale collections of non-numerical information, such as files and lines of code in software systems, library and bibliographic databases, networks of relations on the internet, and so forth” [17]. Its interdisciplinary approach includes Computer Graphics, Human-Computer Interaction, Visual Design, and Psychology. It has many applications vary from scientific research, to data mining and crime search.

The question “why do we visualize data or information ?” may arise, we can answer this question with Nathan Shedroff’s “continuum of understanding” [18]. He analyzes the process of understanding and describes it as a continuum that produces information from data, where information is transformed into knowledge and finally into wisdom. According to Mazza, Information Visualization lays between data and information when Nathan’s model is taken into account [1]. We visualize data to produce information where that information is transformed into knowledge with the help of user’s experience. Visual representations help us to ease that process.

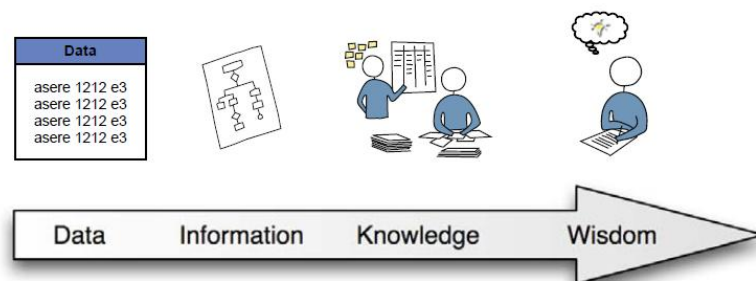


Figure 2.1: The continuum of understanding, according to Nathan Shedroff [1].

2.2 What Is Good Visual Representation ?

This is one of the hardest questions and many researchers are trying to find criteria by setting this as a research challenge for themselves. Edward Tufte, one of the most prominent researchers with two excellent milestone books points out some criteria to describe that a visual representation is effective. According to Tufte, a good picture is a well-built presentation of “interesting” data [19, 20]. It is something that brings together substance, statistic, and design. It aims to clearly, precisely, and efficiently present and communicate complex ideas. More generally, it aims to provide the viewer with “the greatest number of ideas, in the shortest time, using the least amount of ink, in the smallest space” [1].

Ben Shneiderman’s information visualization mantra is ”Overview, zoom and filter, details on demand” [21]. This approach can show researchers a roadmap for good visual representations.

2.3 History of Information Visualization

In this section we will briefly describe the work done in history to better state that information visualization is not a research area after the invention of computers and many methods were developed before the computer era.

In Figure 2.2 the French engineer, Charles Minard (1781-1870), illustrated the disastrous result of Napoleon’s Russian campaign of 1812. The size of Napoleon’s army is shown as the width of the band in the map, starting on the Russian-Polish border with 422,000 men. By the time they reached Moscow in September, the size of the army dropped to 100,000. Eventually, only a small fraction of Napoleon’s original army survived [3]. It is considered as the best statistical graphic ever drawn.

In Figure 2.3 the Scottish engineer William Playfair’s (1759-1823) “The Commercial and Political Atlas”, published in 1786, is shown. He is generally viewed as the inventor of most of the common graphical methods of statistics and display. Line plots, bar chart and pie chart are first introduced by Playfair. In this visualiza-

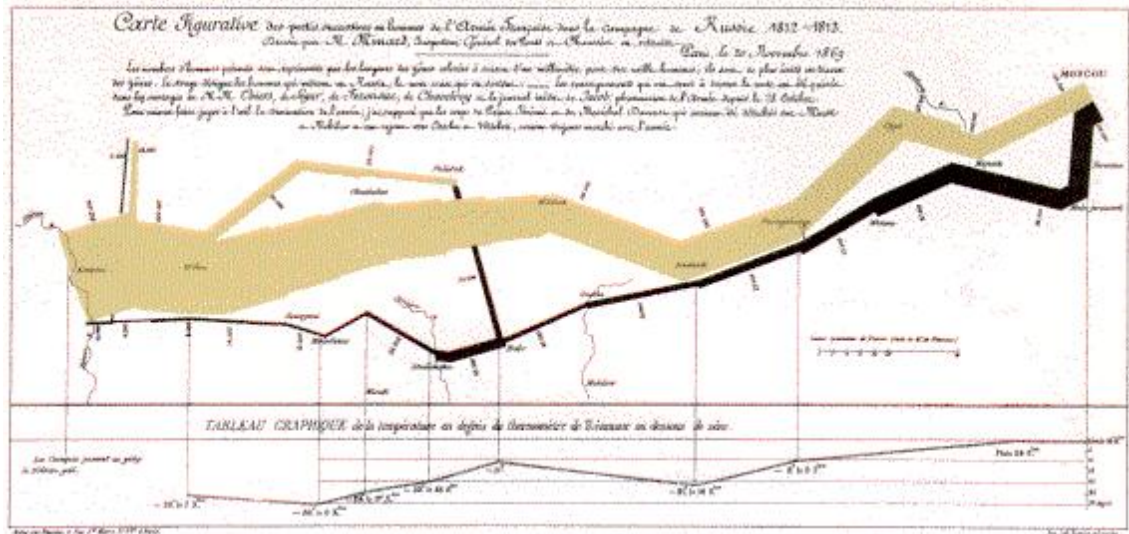


Figure 2.2: Napoleon’s Russian Campaign of 1812

tion, the area between two time-series curves was emphasized to show the difference between them, representing the balance of trade [20].

In Figure 2.4 Florence Nightingale’s polar area diagram is presented. Florence Nightingale (1820-1910) is described as “a true pioneer in the graphical representation of statistics”, and is credited with developing a form of the pie chart now known as the polar area diagram, or usually the Nightingale rose diagram, equivalent to a modern circular histogram, in order to illustrate seasonal sources of patient mortality in the military field hospital she managed [22].

In Figure 2.5 Dr. John Snow’s spot map is presented. Snow is considered to be one of the fathers of epidemiology. He traced the source of a cholera outbreak in Soho, England, in 1854. He used a spot map to illustrate how cases of cholera clustered around the pumps. He also made use of statistics to illustrate the connection between the quality of the source of water and cholera cases [20]. Snow plotted deaths by dots and water pumps by crosses.

In Figure 2.6 both the tube map of 1908 and the modern tube map is presented [23]. The modern version of map is based on the topological design of Harry Beck in 1933. Harry Beck’s classic schematic design of the London underground map shows us that a good design is not necessarily built on geometric details even

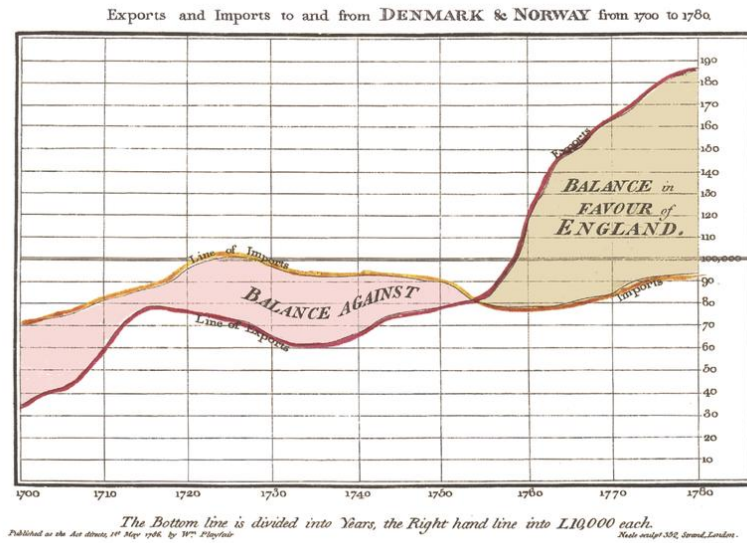


Figure 2.3: Playfair's chart

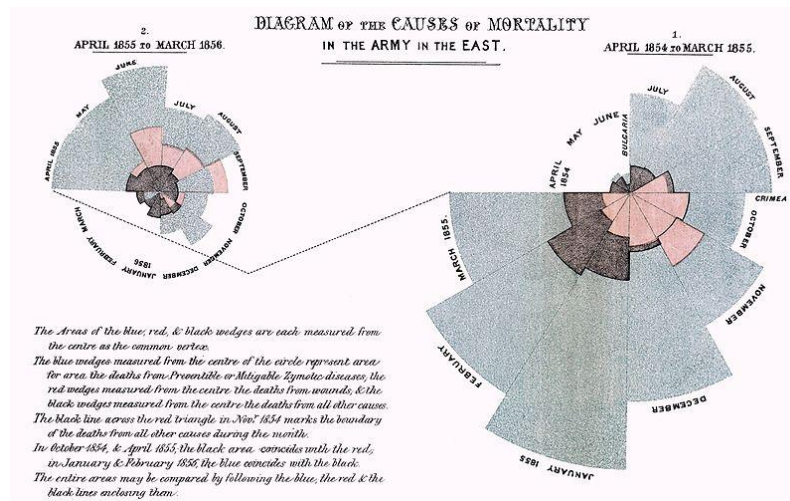


Figure 2.4: Florence Nightingale's rose diagram.

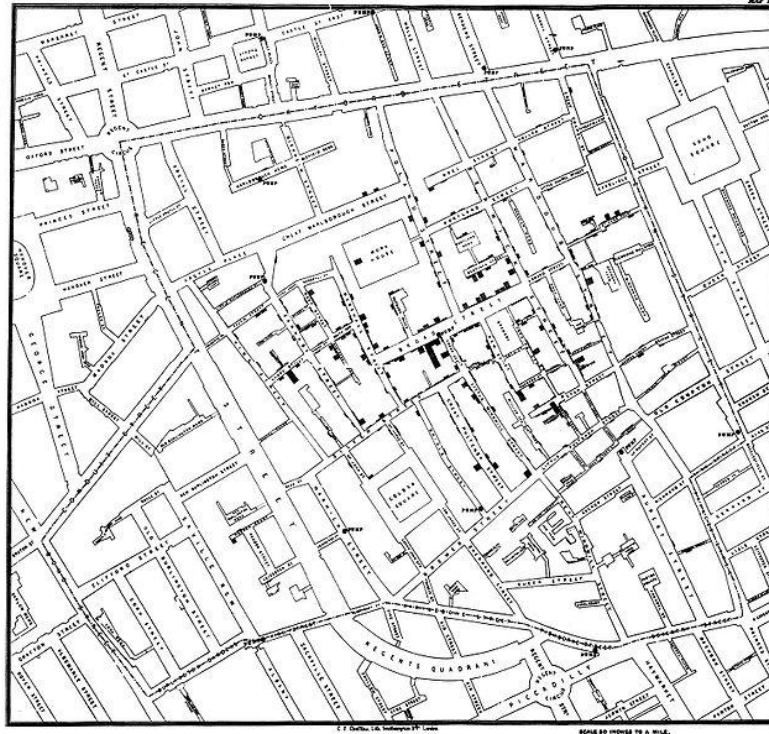


Figure 2.5: The 1854 London Cholera Epidemic

those details come with the data [3].

2.4 Information Visualization Techniques

In this section we will demonstrate some techniques heavily used in information visualization. The techniques and methods presented here do not cover all the state of art visualization approaches; instead an overview is presented. In Figure 2.7 a periodic table is shown [2]. In this figure the state of art data or information visualization methods are presented in visual form.

2.4.1 Graph Drawing

A drawing of a graph or network diagram is basically a pictorial representation of the vertices and edges of a graph [24]. Graph drawing techniques have been used in information visualization, as well as in VLSI design and software visualization. An example of graph drawing is shown in Figure 2.8. In this figure a co-citation

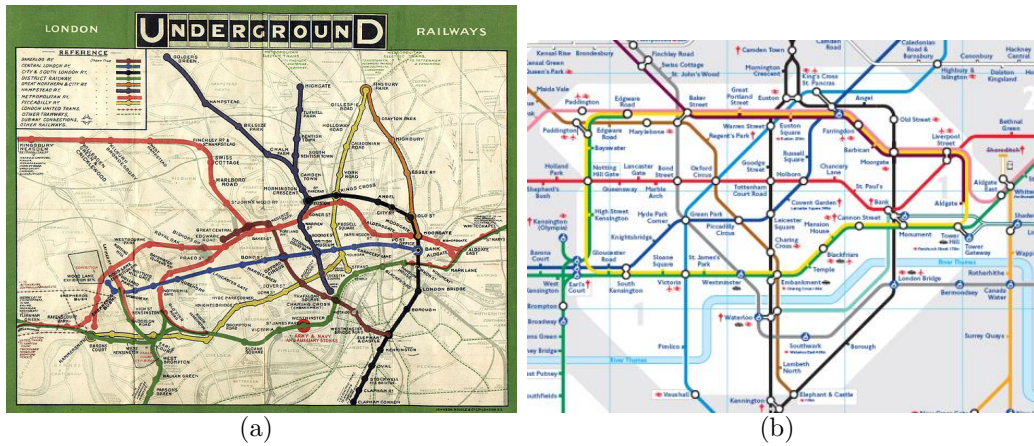


Figure 2.6: (a) Tube map of 1908, (b) The modern tube map, based on the simplified topological design invented by Beck.

A PERIODIC TABLE OF VISUALIZATION METHODS

€ cartesian		Data Visualization Visual representations of quantitative data in schematic form (either with or without axes)		Strategy Visualization The systematic use of complementary visual representations in the analysis, development, formalization, communication, and implementation of strategies in organizations.		G graphic facilities	
Tb table		Ca cartesian coordinates		Information Visualization The use of interactive visual representations of data to amplify cognition. This means that the data is transformed into an image, is mapped to screen space. The image can be changed by users as they proceed working with it.		Metaphor Visualization Visual Metaphors position information graphically to organize and structure information. They also convey an insight about the represented information through the key characteristics of the metaphor that is employed.	
Pi pie chart		L line chart		Concept Visualization Methods to elaborate (mostly) qualitative concepts, ideas, plans, and analyses.		Compound Visualization The complementary use of different graphic representation formats in one single scheme or frame.	
B bar chart		Ac area chart		R radial chart		Pa radial coordinates	
Hi histogram		Sc scatterplot		Sa safety diagram		In information tree	
Th tree plot		Sp specification		Da data map		Tp taxonomy	
Hy Structure Visualization		Overview		Detail		Detail AND Overview	
Overview		Detail		Detail AND Overview		Divergent thinking	
Divergent thinking		Convergent thinking		Convergent thinking		Convergent thinking	

Note: Depending on your location and connection speed it can take some time to load a pop-up picture.
© Ralph Lengler & Martin J. Eppler. www.visual-literacy.org version 1.5

Su supply demand graph	Pc performance charting	St strategy map	Oc organization chart	Ho house of quality	Fd feedback diagram	Ft failure tree	Mq milk quality diagram	Ld life-cycle diagram	Po porter's five forces	S s-curve	Sm stakeholder map	Is ishikawa diagram	Tc technology roadmapping
Ed edgewise box	Pf parallel diagram	Sg signal flow diagram	Mz matrix organization	Z zebra's psychological box	Ad advertising diagram	De decision discovery diagram	Bm log matrix	Stc strategy canvas	Vc value chain	Hy hyper-plot	Sr scenario rating map	Ta top	Sd strategy diagram

Figure 2.7: A Periodic Table Of Visualization Methods [2]

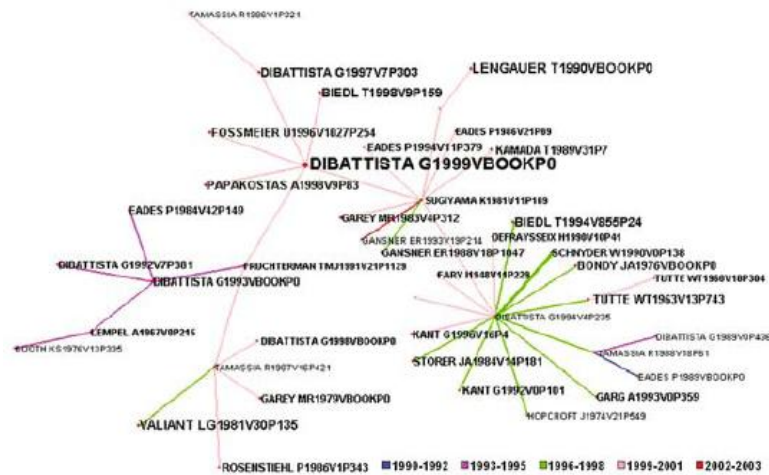


Figure 2.8: A Co-citation Map of Graph Drawing Articles(1990-2003) [3]

network is represented by a graph, where each node is a published article or book.

In Figure 2.9 a visualization that shows the dependencies among classes within the Flare library is presented. The classes in package are positioned along the circle and links that indicate the dependency between the classes are represented by lines. Chen lists several challenges and some good heuristics with graph drawing [3]. The scalability of layout algorithms which can output readable and understandable visualization is one of the most important challenges in graph drawing.

2.4.2 TreeMap

Treemapping is a visualization method for displaying hierarchical data by using nested rectangles. It utilizes a space-filling algorithm that fills recursively divided rectangle areas with components of a hierarchy. A tree map example which shows drink preferences in a small group of people is presented in Figure 2.10.

2.4.3 HeatMap

A heatmap is a graphical representation of data where the values taken by variables are represented as colors in two-dimensional table. The representation can be a 2D matrix as well as a geospatial map. In Figure 2.11.a a 2D matrix representation of a heatmap is shown, in Figure 2.11.b a geospatial heatmap is presented [25].

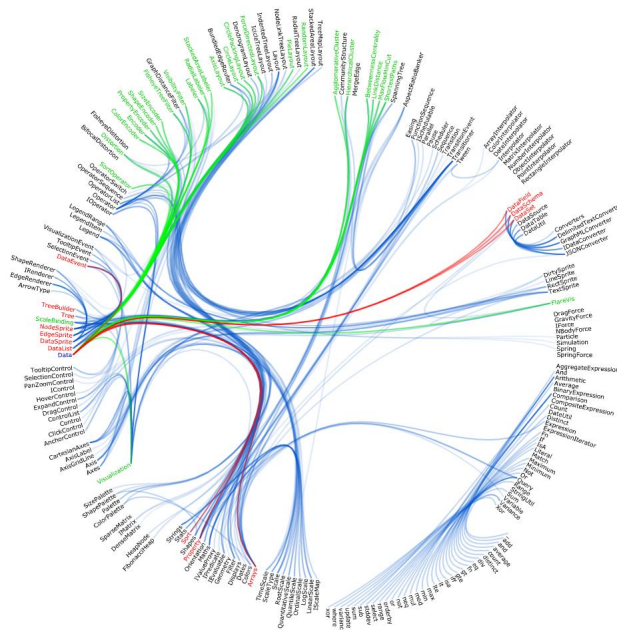
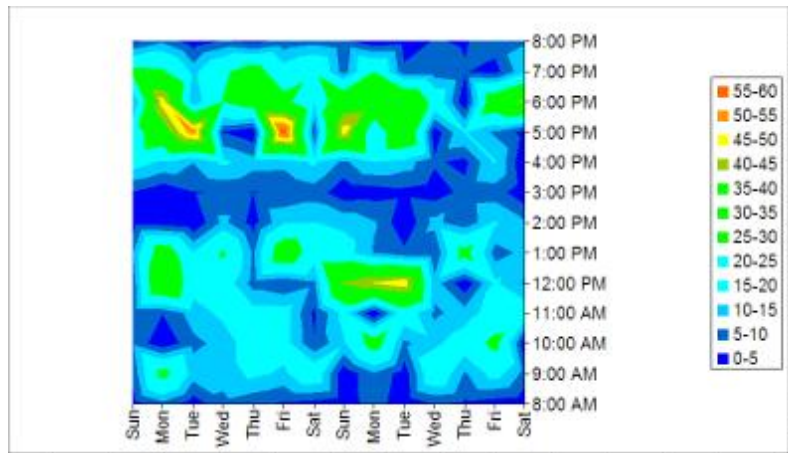


Figure 2.9: The Flare Dependency Graph is a ring-based layout showing the dependencies between classes in the Flare library [4]



Figure 2.10: Treemap of soft drink preference in a small group of people.



(a)



(b)

Figure 2.11: Heatmap Visualization

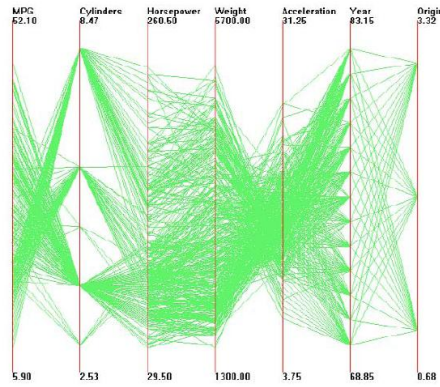


Figure 2.12: Parallel coordinates for 730 elements with 7 variant attributes [1].

2.4.4 Parallel Coordinates

Parallel coordinates is an intuitive way of visualizing high-dimensional or multivariate data. In this technique the attributes are represented by the axis, where they are parallel and equally spaced. Each record in dataset is depicted with a line segment where the values on axes are connected. In Figure 2.12 an example of parallel coordinates with 7 variant attributes is shown. Although parallel coordinates is a powerful technique, it lacks scalability. For large dataset the visualization can be dense and non-distinguishable.

2.4.5 Flowmap

Flowmap is a displaying method of flow data. This type of data contains two different locations and a connection item that represents trucks, people, items or communications. The data item is specific about where the flow starts and a destination where the flow ends. In Figure 2.13 a flowmap is shown which visualizes the outgoing migration from the Colorado state.

2.5 Information Visualization Problems

Although many visualization techniques for different problem domains exist today, there are still major problems with information visualization methods. When a visualization method is analyzed in depth, we see several problems with it. For

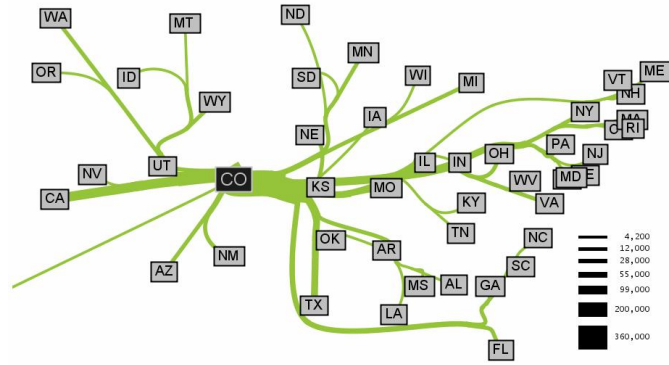


Figure 2.13: Flowmap: Outgoing Migration Map from Colorado for 1995-2000 [5]

instance in graph drawing many layout algorithms work nice with tens of node or up to a hundred nodes, when the node size goes several hundreds or thousands layout algorithms tend to break due to instability. In the case layout algorithm does not loose its stability, then the issues such as aesthetics, readability, understandability or perception usually come into play.

Chen lists the visualization problems in his article entitled “Top 10 Unsolved Information Visualization Problems” in 2005 [26]. When we examined the problems identified by Chen in detail, we realize that we still face those issues; however many ongoing information visualization researches are trying to solve or tackle them. Some problems can be stated as user centered, some problems are technical challenge or “need tackling at the disciplinary level”. The problems identified by Chen vary from usability to understanding elementary perceptual-cognitive tasks, from scalability and quality measures to aesthetics.

Keim et al., in their notable paper entitled “Visual Analytics: Scope and Challenges” break down the information visualization challenges into two categories: “Application Challenges” and “Technical Challenges” [27]. In application challenges category they talk about the use of information visualization in diverse domains and the challenges presented by these domains. In technical challenges they list 10 technical challenges varying from problem solving to user acceptability, from data quality and uncertainty to scalability.

We refer the reader to these two excellent articles for further and detailed explanations about the information visualization domains, application areas, and the scopes as well as the challenges that arise both from the nature of domain and the techniques.

3 LITERATURE SURVEY

This chapter will be discussed in three different subsections. The first one will discuss about the techniques for social network analysis and visualization, the second one will elaborate on the viewpoint generation, informativity and quality of views, and the third one will present the camera control techniques used in virtual environments.

3.1 Social Network Analysis and Visualization

In recent years many methods have been developed for social network analysis to rank nodes, to discover hidden links, to deduce meaningful information by the help of statistical, dynamic or visual perspective analyses [28]. The context of social network analysis varies from dark networks [29], to collaboration networks [30] or to networks in biological sciences.

Statistical analysis of social networks uses statistical properties of graphs including clustering, degree distributions or centrality measures to deduce useful information. Centrality measures determine the relative importance of a node in a network and the most common ones are degree, betweenness and closeness [31]. A more complex measure i.e. Markov centrality [32] treats the social network as a Markov chain and helps to discover significant facilitators in that network.

Choosing the right centrality for a specific problem is usually a hard task and common approach is comparing different centralities for the same network and building hypothesis about the discovered central nodes [33].

One of the pioneers in exploring key actors for dark networks Sparrow [34] used six centrality measures for their relevance in revealing the mechanics and vulnerabilities of criminal enterprises. Hussain et al. [10] used degree centrality measure to set Bayesian Posterior Probabilities for entropy change calculations to locate key

actors in social networks. Newman [30] defined a different set of statistical measures such as number of authors, mean papers per author, mean authors per paper, number of collaborators, and average degrees of separation for scientific collaboration networks. Crnovrsanin et al. [35] used Markov centrality metric to discover and highlight meaningful links.

Another aspect of social network analysis is to discover the dynamic behaviors of the network which usually takes domain of time into account. Dynamic analysis can include network recovery by multiple representations from longitudinal data to model the evolving network, network measurement of deterministic, probabilistic and temporal aspects and statistical analysis such as continuous Markov model, and Cox regression analysis for determining significant nodes.

Kaza et al. [29] used multivariate survival analysis of Cox regression for significant facilitator discovery. Falkowski et al. [36] proposed a technique to detect the evolution of subgroups and analyzing subgroup dynamics in manner of stability, density, cohesion and distance using temporal and statistical analyses.

3.2 Viewpoint Generation

In recent years many methods have been developed for measuring the quality of the views and have tried to describe the optimum point to place a camera on a scene where it can be viewed the best way. Unfortunately the translation of the term “best” or “good” into measures or numbers is not an easy task. Kamada-Kawai [37] were one of the pioneers in defining a good position to place a camera in a 3D scene. They define a parallel projection of a scene to be good, if the number of surface normals orthogonal to the view direction is minimal. The method has several drawbacks, first it does not guarantee that user will see as much details as possible and will fail when comparing equal number of degenerated faces.

Barral et al. [38] use a modification of the coefficients introduced by Kamada-Kawai in order to cope with perspective projection. They introduce different exploration coefficients, that are combined to determine the quality of a perspective

projection. However, they can not find a good weighting scheme for those factors. The algorithm fails for objects of genus one and larger.

Vazquez et al. [6] propose a metric based on the entropy of the scene. They define the best viewpoint as the one with the highest entropy, i.e. the one that sees the maximum of information. They apply the ratio of the projected area of each face to the area covered by the projection of all faces in the scene. Vazquez et al. suggested the technique in 2001 and made improvements in following years.

Vazquez [39] proposes a new technique to select the views automatically by using depth-based stability analysis. In this work he introduces a new view descriptor which uses depth maps to have three-quarter oblique views for 3D objects. He claims that psychophysical experiments have shown that users often prefer oblique views between frontal and profile views as representative views for 3D objects.

Skolov and Plemenos [40] propose a high level technique and claim the techniques presented above as low-level. They step in the direction of semantic description of a 3D scene and use hierarchical decomposition of them. They define the viewpoint quality as the sum of observation qualities of each decomposed object.

Mesh Saliency is also actively studied for viewpoint selection and mesh simplification. Salient features such as luminance, pixel colors or geometry are used. Koch and Ullman [41] suggest that salient locations in 2D images will be different from its neighbors. Itti et al. [42] propose a method for the calculation of the saliency map using 2D images. They combine information from center-surround mechanisms applied to different feature maps and assign a saliency value to each pixel. Lee et al. [15] propose a geometrical approach for calculation of mesh saliency in 3D models. Their method uses the curvature attribute of the object and Itti et al.'s center-surround mechanism to highlight the regions that are different from their surroundings. Takashi et al. [43] propose a method to locate optimal viewpoints for volumetric objects by decomposing the entire volume into a set of feature components. Bordoloi and Shen [44] use view goodness, view likelihood and view stability concepts to locate viewpoints for volume rendering where viewpoint goodness mea-

sure is based on entropy that uses the visibility of the voxels. Bulbul et al. [11] use the concept of saliency and apply it to the animated meshes with material properties. They compute multiple feature maps including geometry, material and motion where the calculated maps are combined into a cumulative feature map. Liu et al. [45] uses mesh saliency to extract critical points by the help of Morse theory and claims that their technique is more satisfactory and results with the lower number of critical points.

3.3 Camera Control

The camera control can be classified into four different categories or schemes; direct control, through the lens control, assisted control and automated control [16]. The key issues for researchers include the management of the control in the high degrees of freedom, handling of exponentially growing computation complexity and finding effective and reactive measures to avoid the occlusions in the scene.

The direct control is a reactive control type that responses back for the user inputs. Ware and Osborne present possible input mappings for direct camera control metaphors in their review including *eyeball in hand*, *world in hand*, *flying vehicle*, and *walking metaphor* [46]. In *eyeball in hand* metaphor the position and orientation parameters of the camera is directly manipulated by the input device in the user. In *world in hand* metaphor the rotational and positional parameters of the camera is fixed or constrained but in this case the world parameters are manipulated by the input device i.e. the arcball concept introduced by [47]. In flying vehicle metaphor the camera is treated as a flying object and user inputs control the rotational and translational velocities of the camera. This metaphor is widely exercised in 3D games and considered to be the intuitive way of the exploration. However the major concern for players is being lost in environment. Hanson and Wernert [48] present a constrained based navigation system to avoid obstacles in the scene. Turner et al. [49] present an exploration of physics based camera control where the user inputs are treated as forces acting on a weight(in this case it is the virtual camera). Xiao

and Hubbard [50] present the use of vector fields for avoiding the cluttered views while directing the users into the object of interest.

In through the lens control metaphor the camera is controlled by the change of the positions of objects in the environment. Gleicher and Witkin [51] present this paradigm in their seminal paper where they recompute camera parameters to match the user's requested location. The difference between the screen and desired location is considered as velocity. And the relationship between the velocity and displacement of points is expressed through the Jacobian matrix which represents the perspective transformation of the scene.

Assisted camera control technique exploits local or global knowledge about the environment to assist the users through their navigation. It can be classified into two metaphors such as object aware and environment aware assistances depending on their knowledge type [16]. In object aware assistance the proximal object inspection is used for collision avoidance such as ray casting, and in environment aware assisted camera control metaphor the global knowledge about the scene is used to avoid obstacles or direct the user to interesting parts. Elmqvist et al. [52] use scene voxelization, connectivity graph and TSP-like algorithm to assist the user in their guided navigation framework. Andjar et al. [53] exploit the concept of Viewpoint Entropy for indoor navigation. They use cell and portal decomposition together with the calculated viewpoints in each cell. This work resembles the most to our work however, instead of indoor portals, our environment is large scale terrains, we use our Greedy N-Best View Selection algorithm for calculations in the regions extracted by the help of road network data. We also utilize the evolutionary programming paradigm to find the path between the calculated viewpoints. The details of our approach will be discussed in subsequent sections.

In automated camera control, the transformation and rotational attributes of the camera is directly computed using either the generated image, or the fitness function that needs to be optimized. Visual servoing or target tracking is one example of automated camera control using the image analysis technique. Visual servoing uses

the feedback information extracted from a vision sensor to control the motion of a robot [54]. In optimization based automated camera control, the deterministic or non-deterministic optimization methods are employed to find the camera configuration. For instance Bares et al. [55] propose the use of a complete search space as an optimization approach. In our technique we employ the divide and conquer metaphor. We calculate camera positions for sub-regions of the terrain and utilize a non-deterministic approach such as population-based genetic TSP to calculate the final camera path.

4 SENSITIVITY ANALYSIS AND VISUALIZATION OF SOCIAL NETWORKS

This chapter introduces a technique to analyze and visualize a social network using Shannon's entropy model.

Social network analysis [35, 10, 29, 56] has applications in many areas including organizational studies, social psychology and information science. The goal is to distinguish and detect regular or non-regular patterns, tendencies, mutual interests and reveal hidden information to execute the required tasks by perceiving the information presented.

In this work we presented a visualization approach that uses coloring, sizing and filtering to help the users perceive the presented information. We used degree entropy and presented novel measures such as betweenness and closeness entropies to conduct network sensitivity analysis by means of evaluating the change of graph entropy via those measures. We integrated the result of our analyses into a visualization application where the social network is presented using conventional node-link diagram.

The visualization provided in this work uses general mantra of information visualization where the *size* of visual representation of an actor depends on the amount of change in system entropy caused by the actor and the *color* information is mapped from the graph clustering or conducted sensitivity analyses. *Filtering* of edges and nodes is also provided to ease and improve the perception of complex graphs. The main contribution of this study is a visualization where the information communicated from a social network is enhanced by the help of clustering and sensitivity analyses.

The rest of the chapter is organized as follows: in Section 4.1 we describe the system architecture, input and outputs of the processing components and the system flow for visualization of the social network data. In Section 4.2 we review the commonly used social network centralities, in Section 4.3 we present entropy based sensitivity analysis of a social network, in Section 4.4 we discuss about the visualization and analyze the outputs. Section 4.5 concludes our work.

4.1 System Overview

The visual display of social network data using entropy enhancement requires several steps as shown in Figure 4.1. One of them is to create a social network data or a social network graph. In order to accomplish this task we employed the DBLP [9] data and filtered the papers published in ACM SIGGRAPH conference and journals in TVCG (IEEE Transactions on Visualization and Computer Graphics (TVCG)) between years 2005 and 2009. The filtered publications form the basis for the collaboration network creation.

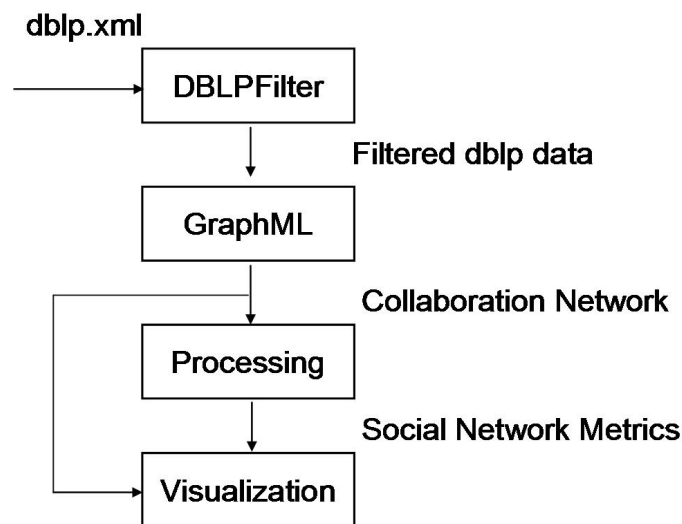


Figure 4.1: Social Network Visualization System Overview

The second step is achieved by creating a social network graph via the help of filtered publications. This task includes creating a node for each author and connecting the links between authors with the papers published together. This

graph is defined as a collaboration graph. The collaboration graph is an XML file which uses GraphML file format [57]. In the processing step the collaboration graph is analyzed by means of sensitivity and social graph metrics where the produced output is used to derive the visualization of the network. In the visualization step we provided a 2D presentation that maps the calculated metrics to the color and size of the actors displayed on the screen.

The metric creation and techniques for sensitivity analysis will be explained on the next section.

4.2 Social Network Centralities

There are various measures of the centrality of a node within a graph that determine the relative importance of a node. For example the centrality measure for a social network can map to solve how important a person is within that social network or the effect of a person in the connectivity of the social network. Many of the centrality concepts were first developed in social network analysis, and used in terms to reflect a sociological origin.

4.2.1 Degree Centrality

Degree centrality is defined as the number of links incident on a node. If the network is directed, indegree and outdegree centralities are defined. Indegree is a count of the number of links directed to the node, and outdegree is the number of links that the node directs to others. For relations such as friendship, indegree is interpreted as popularity, and outdegree as gregariousness. For the social network in our domain, the graph is undirected and degree of a node is the number of all incident links.

In order to find the degree centralities of the nodes, the number of incident links are counted and recorded. The recorded values are processed to normalize the values to $[0, 1]$. The equation used for normalization is shown in (4.1). In this equation $C_d(v)$ denotes the the degree centrality of vertex v , $min(C_d)$ is the minimum, and $max(C_d)$ is the maximum of the degree centrality of the network, and $norm(C_d(v))$

is the normalized degree centrality of the vertex v .

$$norm(C_d(v)) = \frac{C_d(v) - \min(C_d)}{\max(C_d) - \min(C_d)} \quad (4.1)$$

4.2.2 Betweenness Centrality

Betweenness is a centrality measure of a node within a graph. It was introduced as a measure for quantifying the control of a human on the communication between other humans in a social network by Freeman [58]. In his conception, nodes that have a high probability to occur on a randomly chosen shortest path between two randomly chosen nodes, have a high betweenness.

For a graph $G := (V, E)$ with n nodes, the betweenness $C_b(v)$ for vertex v is computed as follows:

1. For each pair of nodes (s, t) , compute all shortest paths between them.
2. For each pair of nodes (s, t) , determine the fraction of shortest paths that pass through the vertex in question (here, vertex v).
3. Sum this fraction over all pairs of nodes (s, t) .

The formula to calculate the betweenness centrality is shown in equation(4.2) [59].

$$C_b(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}} \quad (4.2)$$

where σ_{st} is the number of shortest paths from s to t , and $\sigma_{st}(v)$ is the number of shortest paths from s to t that pass through a vertex v . This may be normalized by dividing through the number of pairs of nodes not including v , which is $(n - 1)(n - 2)$ for directed graphs and $(n - 1)(n - 2)/2$ for undirected graphs. This scaling emphasizes the highest possible value where a node is crossed every shortest path. In this work we used the normalization method shown in equation(4.3) where the betweenness centrality values are mapped $C_b(v) \in [0, 1]$.

$$norm(C_b(v)) = \frac{C_b(v) - \min(C_b)}{\max(C_b) - \min(C_b)} \quad (4.3)$$

An example figure for graph betweenness is presented in 4.2 where Hue shows the node betweenness.

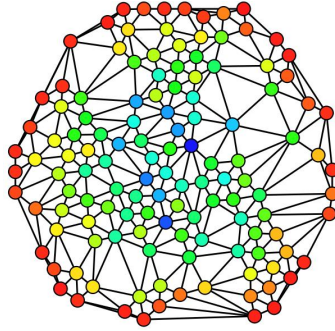


Figure 4.2: Hue (from red=0 to blue=max) shows the node betweenness.

4.2.3 Closeness Centrality

Closeness centrality is a natural distance metric between all pairs of nodes defined by the length of the shortest distance between them. It is the inverse of farness where the farness for a node s is defined as the sum of distances to all other nodes [31].

Closeness can be regarded as a measure of how long it will take to spread information from s to all other nodes sequentially. Thus when a node is the more central the lower its total distance to all other nodes. The closeness $C_c(v)$ for a vertex v is the reciprocal of the sum of geodesic distances to all other vertices of V as shown in equation(4.4):

$$C_c(v) = \frac{|V| - 1}{\sum_{t \in V \setminus v} d_G(v, t)} \quad (4.4)$$

The closeness centrality values are mapped to $[0, 1]$ using the normalization equation(4.5).

$$\text{norm}(C_c(v)) = \frac{C_c(v) - \min(C_c)}{\max(C_c) - \min(C_c)} \quad (4.5)$$

4.3 Sensitivity Analysis

The sensitivity of an actor in the social network reveals the importance of relation between the actor and all other participants. Here we present an analytical approach using centrality entropy distributions which can be considered as good indicators of network sensitivity. We define three centrality entropy distributions, degree entropy, betweenness entropy and closeness entropy. Combined information is presented by the normalization of centrality entropy distributions discussed in this work. Subsections will describe the centrality entropies via the help of Shannon Entropy.

4.3.1 Degree Entropy

The Shannon entropy [8] of a discrete random variable X with values in the set $\{x_1, x_2, \dots, x_n\}$ is defined as

$$\mathbf{H}(\mathbf{x}) = - \sum_{i=1}^n p(x_i) \log_b p(x_i) \quad (4.6)$$

In equation (4.6) $p(x_i)$ is the probability mass functions of state x_i , for a system with n different states. In our context the probability mass function set is the degree distribution of the actors in the social network and n is the number of distinct actors. Hence we defined the the probability mass function $p(x_i)$ of the node x_i using the degree centrality as shown in equation(4.7)

$$p_d(x_i) = \frac{\text{norm}(C_d(x_i))}{\sum_{j=1}^n \text{norm}(C_d(x_j))} \quad (4.7)$$

Safar et al. [60] defines a similar probability equation in their evolutionary programming inspired cyclic entropy maximization. They use Barabasi et al. [61]'s generation algorithm for experimenting on scale-free networks, where the degree based distribution is used to link nodes for finding an optimal distribution where the total entropy of the network is maximized. However we use degree centrality

distribution to calculate the entropy of the social network by interpreting the actors as the states of a system.

In order to conduct sensitivity analysis using degree entropy, the initial information amount, hence degree entropy is recorded including all the actors in the network. An actor is removed from the network and the system entropy is recalculated for the remaining actors. To calculate the system entropy we use the largest connected component of the subgraphs if the actor disconnects the network. The calculated entropy value is recorded and actor is connected back to the network. This sequence is applied to all actors in the social network.

The system entropy change analysis for each actor is performed by taking difference of initial system entropy and remaining system entropy. Hence the entropy is defined as the quantification of information amount, the change between initial and remaining system entropy is defined as the amount of change caused by the actor.

The recordings of sensitivity analysis are normalized during the output to be processed by the visualization system provided in this work.

4.3.2 Betweenness Entropy

The betweenness entropy is defined as the information amount revealed by the graph using betweenness centrality. We exploit the same concept mentioned before. We specified the system with n different states as a social network with n different actors.

The probability mass function set is interpreted as the betweenness distribution of the actors in that social network. The distribution is created by using the normalized betweenness centralities shown in equation(4.8):

$$p_b(x_i) = \frac{\text{norm}(C_b(x_i))}{\sum_{j=1}^n \text{norm}(C_b(x_j))} \quad (4.8)$$

The sensitivity analysis using betweenness entropy is done similar to the degree entropy analysis. The initial system entropy using betweenness probability mass

function is calculated and recorded, and each actor is removed from the network where the betweenness entropy is calculated for the social network with remaining actors. The change between the initial entropy and remaining is recorded as the change caused by the actor and actor is connected back to the network. After the recordings, the values are normalized.

4.3.3 Closeness Entropy

The closeness entropy is defined as the information amount revealed by the graph using closeness centrality. In this sense the social network with n actors is interpreted as a system with n different states. The information measure that needs to be quantified is closeness in this case.

$$p_c(x_i) = \frac{\text{norm}(C_c(x_i))}{\sum_{j=1}^n \text{norm}(C_c(x_j))} \quad (4.9)$$

We used the values calculated in equation(4.9) as the probability mass function for the equation(4.6) to compute closeness entropy for the social network. The sensitivity analysis is done using the sequence presented in previous sections; however in this case closeness entropy is used as probability mass function.

4.3.4 Combined Approach

Degree, betweenness and closeness entropies are combined to measure the aggregate sensitivity of each actor in the network. The combination approach can be either product or summation of the values. Since we have normalized sensitivity (i.e. change information); the summation operation would be a reasonable approach. However to favor the actors that have high values jointly in degree, betweenness and closeness entropy changes, we selected the product as the aggregation method. With this scheme we can emphasize those actors in the final visualization.

Either summation or product operation is used, the aggregation helps to incorporate three centrality change information into a single, measurable and displayable

value.

$$\mathbf{Combined}(\mathbf{v}) = I_d(v).I_b(v).I_c(v) \quad (4.10)$$

In equation(4.10), $I_d(v)$ denotes degree change information, $I_b(v)$ denotes betweenness change information and $I_c(v)$ is closeness change information where we treat information as the system entropy. The user can select any of them as well as the combined one for further analysis using the visualization system provided in this work.

4.4 Discussion and Visualization

There are many techniques found in literature [62] for social network visualization varying from node-link diagrams, to tree-maps, from adjacency matrix representations [63] to sophisticated 3D visualizations, however we believe that node-link diagrams are most suitable presentation of social networks for human perception.

In this work, we provide a visualization application that demonstrates the social network with conventional node-link diagram. Centrality measures, centrality measure entropy changes i.e. sensitivities are conveyed to the user via drawn nodes. For instance if an actor changes the system entropy more than the other actors, that actor is represented with a greater ellipse. The layout and clustering analysis is done using the energy-based minimization model presented by Noack [64].

The sensitivity analyses using centrality measure entropies show the changes to the system entropy caused by the actors in the network. The cause of change differs by the amount of information decreased from the initial information quantity calculated for the system. The change is sensitive to two factors, the number of disconnected nodes caused by the actor after removal, and the centrality measure entropy amount of the disconnected actors, which actually complies with the aim of sensitivity analysis that is revealing importance of relation between the actor and all other participants in the system.

4.4.1 Discussion

Here we discuss about a hand generated social network example shown in Figure 4.3. In this example 13 actors are collaborating in a tree shaped flow, where the actor-0 is on the center of this collaboration, hence the root node.

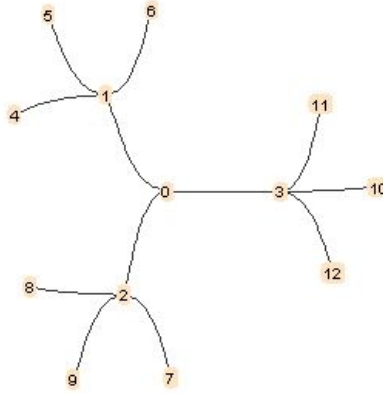


Figure 4.3: An example social network

Actor	Degree		Betweenness		Closeness		Combined
	Cent.	Ent.	Cent.	Ent.	Cent.	Ent.	
0	0.667	1.0	1.0	1.0	1.0	1.0	1.0
1	1.0	0.265	0.625	0.188	0.555	0.395	0.0197
2	1.0	0.265	0.625	0.188	0.555	0.395	0.0197
3	1.0	0.265	0.625	0.188	0.555	0.395	0.0197

Table 4.1: Centrality and sensitivity entropy values for the example network. Cent. denotes the centrality and Ent. denotes the entropy sensitivity analysis. Note that difference between columns shows the change reflected by sensitivity analysis, and the difference between rows highlight the ratio emphasized.

When we analyzed the degree centrality of each actor we observe that actors 1,2, and 3 have four links, actor-0 has three links, and the rest has only one link. The normalized degree centralities of the actors are shown in Table 4.1, actors that do not have value, such as 0, are not presented. The degree centrality values are 1.0 for actors 1,2, and 3 and 0.667 for the actor-0. If we solely analyze this centrality measure, the values indicate that the actor-0 does not have that significance compared

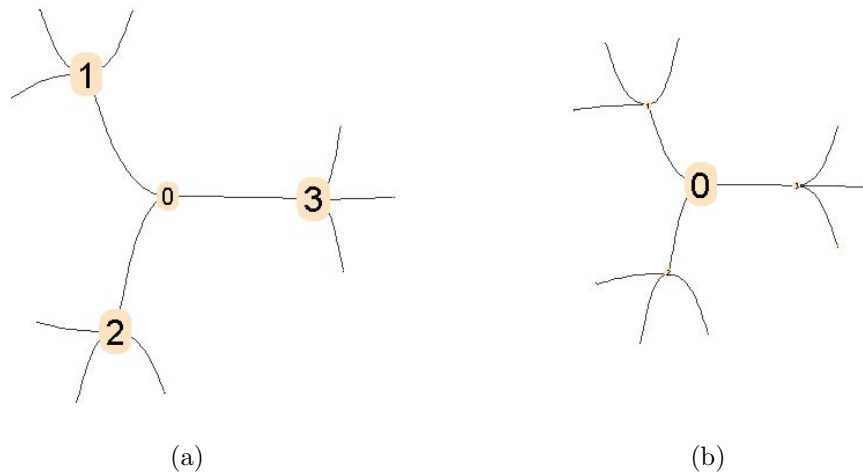


Figure 4.4: Node size mapping (a)Degree centrality, (b)Degree entropy sensitivity analysis

to the actors 1,2, and 3, which does not reveal that the presented social network is more sensitive to the root actor hence it disconnects the whole network. The result of sensitivity analysis of degree entropy is also shown in Table 4.1. This analysis reveals that the change caused by actor-0 is the greatest, hence it has normalized value 1.0, furthermore the difference between the actor-0 and actors 1,2, and 3 is emphasized as shown in table.

In Figure 4.4 the mapping of degree centrality measure and sensitivity analysis are conveyed with the size of displayed actors. We can observe the differences about the actor sizes and emphasized output in Figure 4.4.b.

The result of betweenness centrality and the sensitivity analysis using betweenness entropy are shown in Table 4.1. Note that the normalized values of both measures are 1.0 for actor-0, who has the highest betweenness centrality as being the central node on each shortest path calculation. Although we do not observe any difference between the values of betweenness centrality and its sensitivity analysis using entropy, we can distinguish that the difference amount between actor-0 and the other actors is emphasized in sensitivity analysis. This can also be observed from the Figure 4.5.

The results for analysis of closeness centrality measure and its sensitivity using

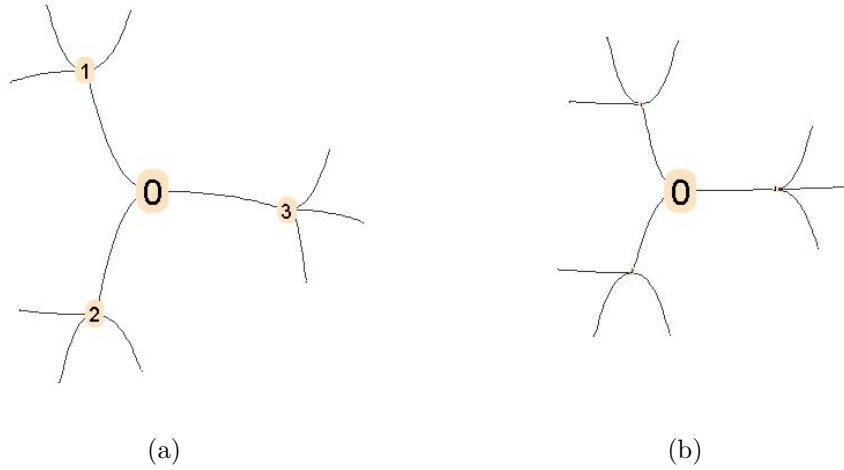


Figure 4.5: Node size mapping (a)Betweenness centrality, (b)Betweenness entropy sensitivity analysis

entropy are presented in Table 4.1 and Figure 4.6 respectively. The same argument made in analysis using betweenness centrality applies here as well. Although the normalized values for centrality measure and its entropy based sensitivity analysis do not differ, the sensitivity analysis emphasizes the difference between the actor-0 and actors 1,2, and 3.

We present the result of combined information change caused by actors in Table 4.1 and Figure 4.6 respectively. It can be observed from the values shown and its mapping to the visualization that the difference between actor-0 and all other actors is emphasized. This information reveals that the effect caused by the actor-0 is the highest and the displayed social network is highly sensitive to this actor.

4.4.2 2D Visualization of Social Network

TVCG collaboration network is visualized using different information and filtering applied to nodes and edges. Figure 4.8 shows the default presentation of the network, no information except the connectivity of the actors is conveyed to the user.

In the following subsections we will analyze the use cases and present the visualizations for them. In order to enhance the perception we filtered the nodes associated with each visualization where their normalized value is less than 0.35 and

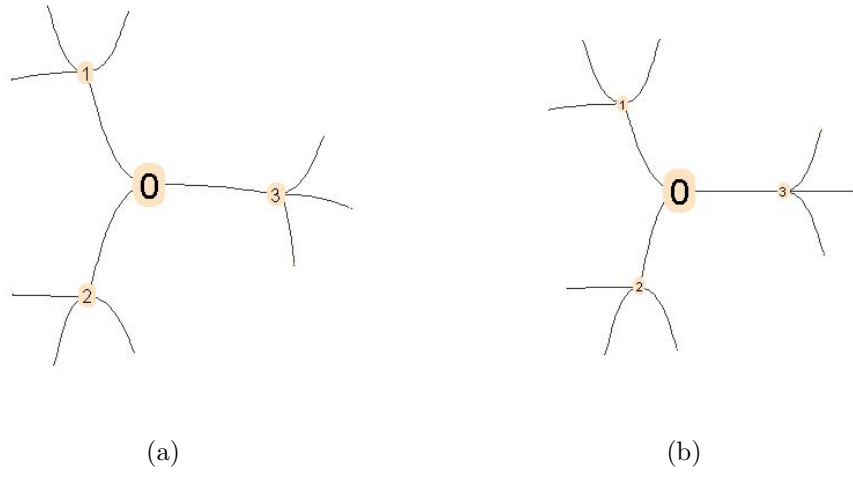


Figure 4.6: Node size mapping (a)Closeness centrality, (b)Closeness entropy sensitivity analysis

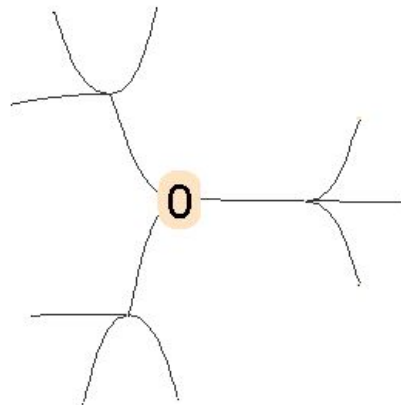


Figure 4.7: Social network visualizing combined information

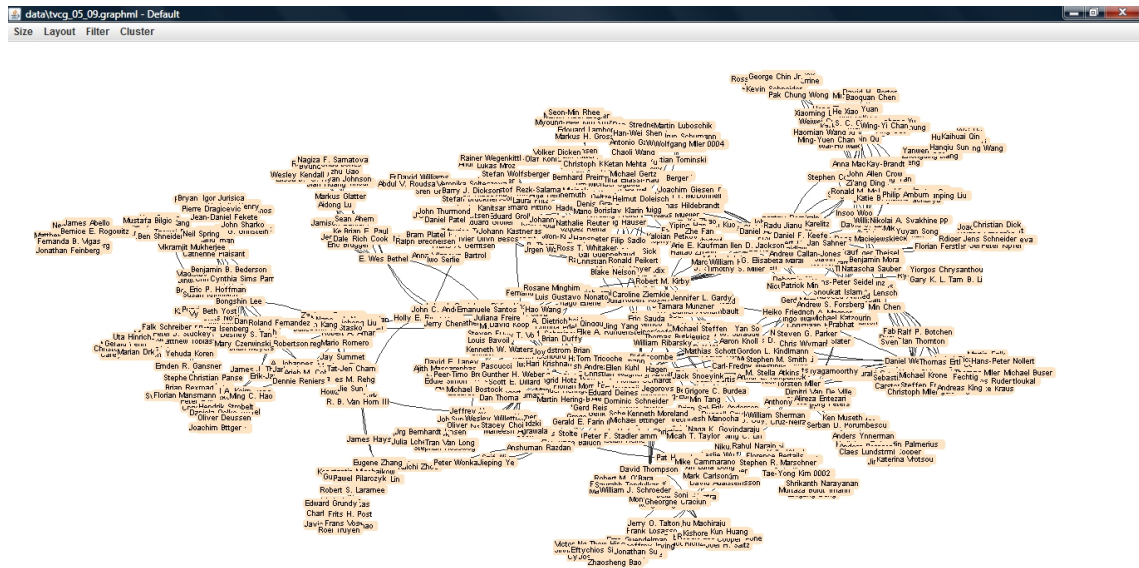


Figure 4.8: Default presentation of collaboration network

colored to convey the cluster analysis information.

Use Case: Popularity

The popularity of the actors can be found with two conducted analyses, hence *degree centrality measure* and *key actor discovery algorithm* presented by Hussain and Arroyo [10]. The results for these two algorithms are shown in Figure 4.9 and Figure 4.10 respectively. Here we can deduce the importance or popularity of an actor from size of the ellipse that denotes the actor.

Use Case: Which actor disconnects the most ?

This question can be answered by the result of sensitivity analysis of degree entropy. It also associates with another important problem “*finding the number of nodes that must be removed from the network before it disconnects into separate networks*”. These problems are robustness indicators of the networks. In social network case, it can be used to target the actor who will give the most damage to the network. Figure 4.11 presents the result of sensitivity analysis of degree entropy applied to the nodes sizes, the user can deduce each actor’s effect using degree entropy to the whole system, hence the actors shown with a greater ellipse change the system

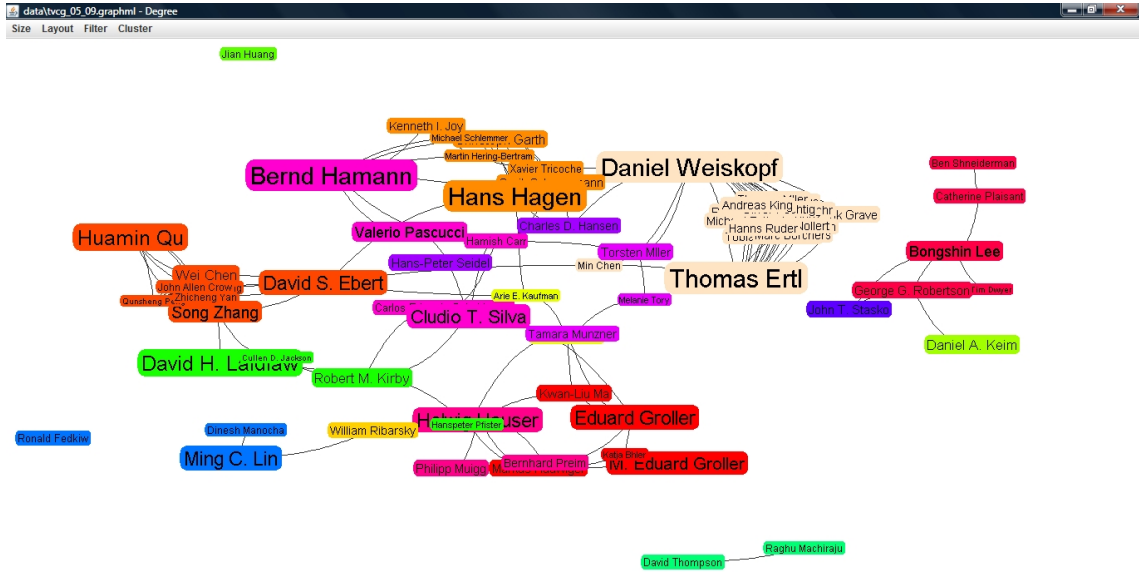


Figure 4.9: Collaboration network visualized using degree centrality

entropy more than the other actors due to the degree centrality.

Use Case: Which actor affects the data flow most?

This question can be answered by the result of sensitivity analysis using betweenness entropy. When a node has higher betweenness centrality it has higher probability of being on many shortest paths of the network. We will exploit this knowledge to legitimate our claim about disrupting the data flow. Data flow in a social network can be affected in two ways : due to disconnection in the network, or removal of an actor who has high betweenness centrality. Our analysis reveals these two cases in a single visualization. In Figure 4.12 the result of sensitivity analysis using betweenness entropy is presented. The actors shown with a greater ellipse change the system entropy more than the other actors.

Use Case: Which actor affects the global connectivity most?

This question can be answered by the result of sensitivity analysis of closeness entropy. The global connectivity in the network is revealed by the closeness of its actors. This connectivity can be disrupted either by causing a disconnection in the network or removing the nodes with high closeness centralities. Our sensitivity

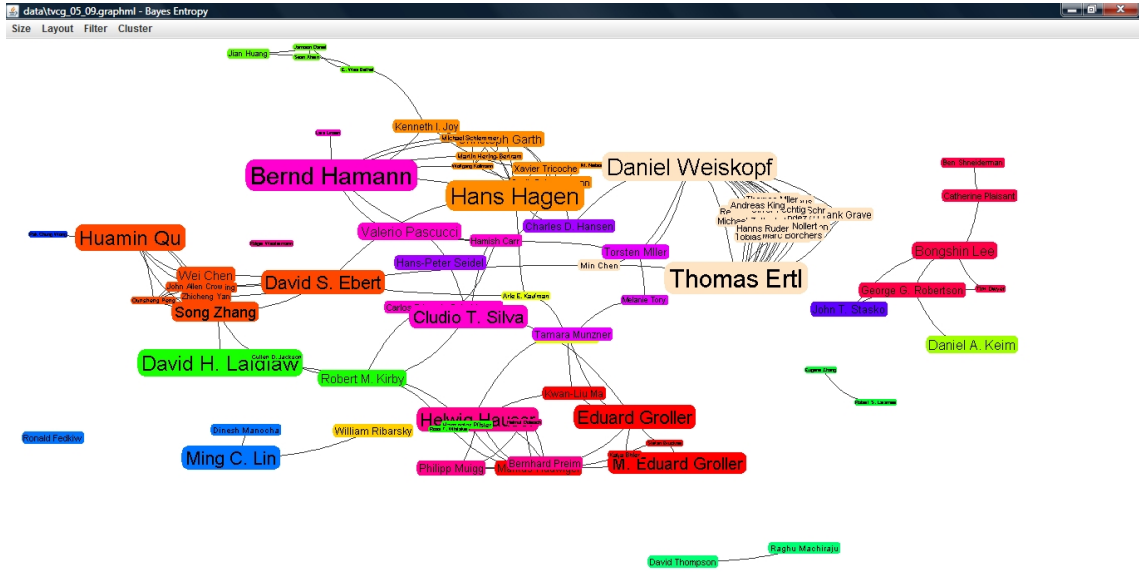


Figure 4.10: Collaboration network visualized using key actor discovery

analysis of closeness entropy reveals these cases. In Figure 4.13 the result of sensitivity analysis using closeness entropy is presented. The actors shown with a greater ellipse change the system entropy more than the other actors hence revealing the information about the effect on global connectivity.

4.5 Conclusion

In this chapter a technique for analyzing and visualizing a social network using Shannon’s entropy definition is presented. We used the three most common centrality measures such as degree, betweenness and closeness to define centrality measure entropies. Centrality measure entropies are utilized to conduct the sensitivity analysis of system employing entropy changes of the actors in the social network.

We tried to enhance the information communicated from a social network by the help of analyses and visualization techniques provided in this work. Experiments are performed using different datasets varying from hand generated to collaboration data extracted from various sources. A social network example TVCG collaboration data is presented here to show the results of our work.

Our experiments have shown that Shannon’s entropy model is a promising tech-

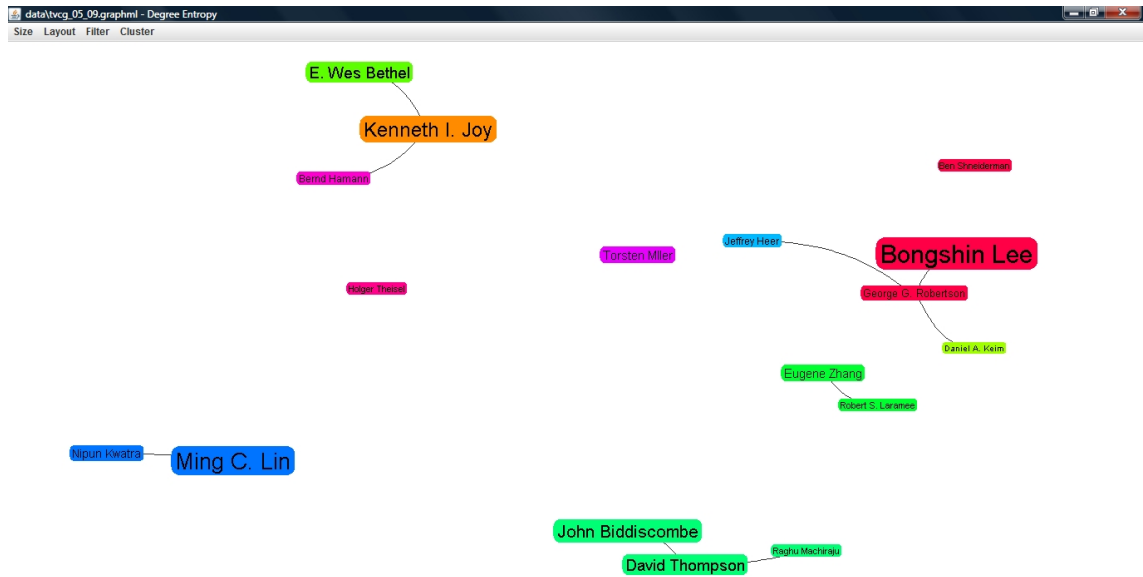


Figure 4.11: Collaboration network visualized using sensitivity analysis of degree entropy

nique for social network analysis and visualization.

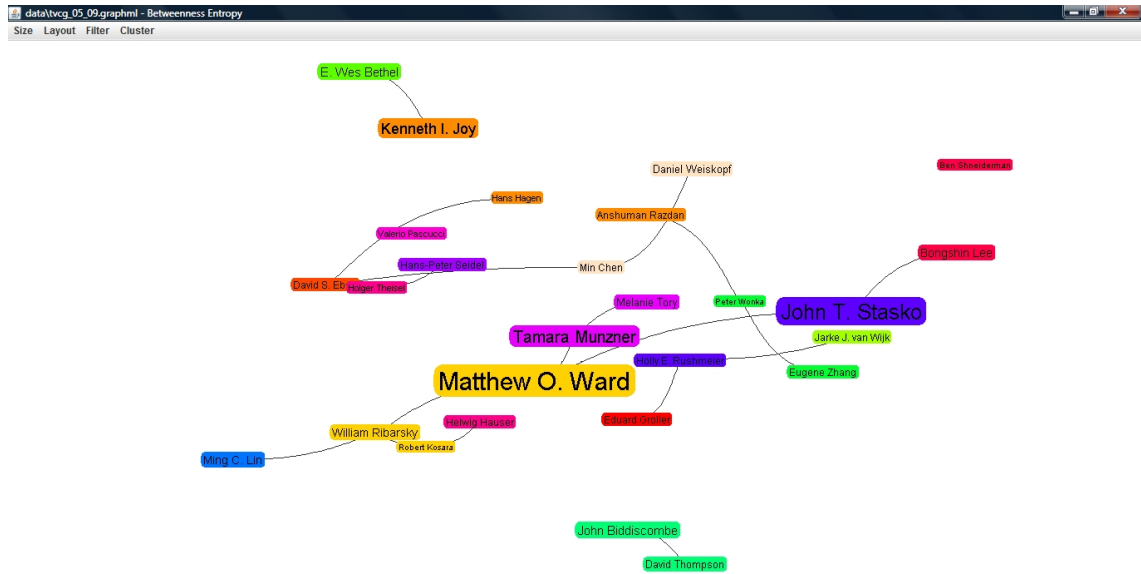


Figure 4.12: Collaboration network visualized using sensitivity analysis of betweenness entropy

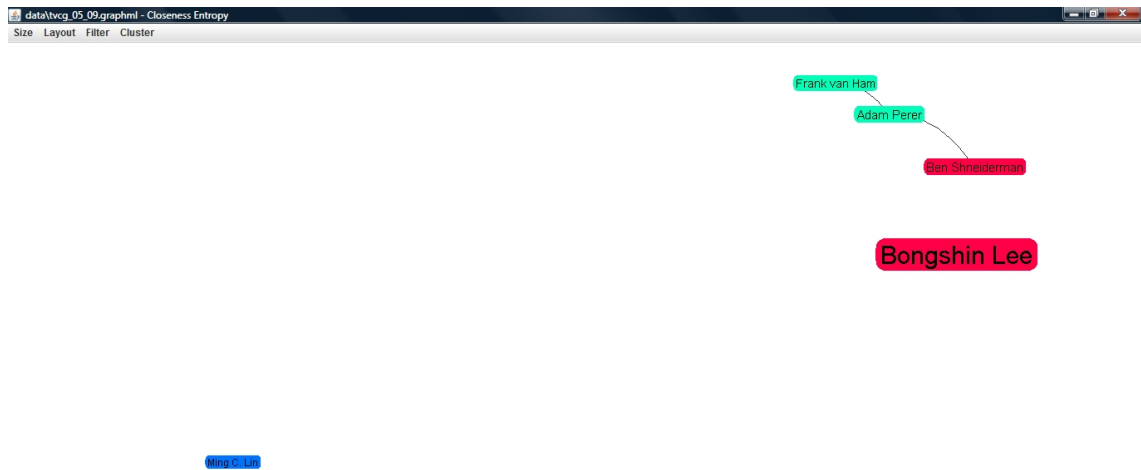


Figure 4.13: Collaboration network visualized using sensitivity analysis of closeness entropy

5 OBJECT EXPLORATION

We introduce a technique to visually inspect a 3D object in a scene with minimal loss of information. We exploit the concept of the viewpoint entropy and introduce a novel view descriptor called mesh saliency entropy for virtual object exploration. The viewpoint entropy is an information theoretical measure which is used to determine the amount of information seen from a certain viewpoint. When the model geometry is considered with no coloring and texturing, the object can only communicate its surface or volume to the viewer by the output of rendering stages. Hence it is the solely information that can be perceived from the model. In this work present a technique to perceive the maximum information from a 3D model by finding a minimal set of camera points which can be defined as the best viewpoints.

In this chapter we also discuss about the evaluation of the entropy as a metric for information coverage, and a usability study to measure the strength of the techniques we provided.

The rest of the chapter is organized as follows: in Section 5.1 and Section 5.2 we give a theoretical background about Viewpoint Entropy, Mesh Saliency and Mean Curvature, in Section 5.3 we present Greedy N-Best View Selection and Mesh Saliency Entropy algorithms and discuss the differences with the methods presented in [6, 65, 66, 7]. In Section 5.4 we examine the statistical results, in Section 5.5 we present a usability study and its outcome, and Section 5.6 concludes this chapter with our remarks.

5.1 Viewpoint Entropy

The entropy [8] of a discrete random variable X with values in the set $\{x_1, x_2, \dots, x_n\}$ is defined as in equation(4.6). Even though the entropy is expressed as a function

of the random variable X , it is actually a function of the probability distribution p of the variable X over the number of distinct symbols N . Entropy function has following two important properties [44];

1. For a given number of symbols N , the maximum entropy occurs for the distribution p_{eq} , where $\{p_0 = p_1 = \dots = p_{N-1} = 1/N\}$.
2. Entropy is a concave function, which implies that the local maximum at p_{eq} is also the global maximum. It also implies that as we move away from the equal distribution p_{eq} , along a straight line in any direction, the value of entropy decreases (or remains the same, but does not increase).

The properties of the entropy function expressed above give us that the calculated viewpoints in extracted regions will be the global maximum points where the object surface is perceived equally.

Viewpoint entropy [6] using Shannon Entropy is defined as

$$\mathbf{I}(\mathbf{S}, \mathbf{p}) = - \sum_{i=0}^{N_f} \frac{A_i}{A_t} \log_b \frac{A_i}{A_t} \quad (5.1)$$

where A_i is the projected area of face i over the sphere, A_t is the total area of the sphere and b is the base of logarithm which is taken as $b = 2$ in this case the result is bits/symbols. In other terms the formula shown above can be translated into where A_t can denote the number of pixels in the image, and A_i can represent the number of pixels that belongs to each face of the object. A_0 is a special case for the projected model or scene onto the screen. For the closed scenes A_0 is taken as 0 and for open scenes A_0 is considered as the number of pixels that belong to the background color. With the contribution of A_0 for open scenes we can have a viewpoint entropy definition that is consistent with Shannon's entropy where $\sum_{i=1}^n p_i = 1$.

The techniques to compute the viewpoint entropy using Graphics Processing Unit can be found in Castello et al. [67]. In this work we used *Frame Buffer Objects* for rendering and calculating optimal camera points. In Figure 5.1 an example model with unique colors assigned to each face is presented for calculating viewpoint

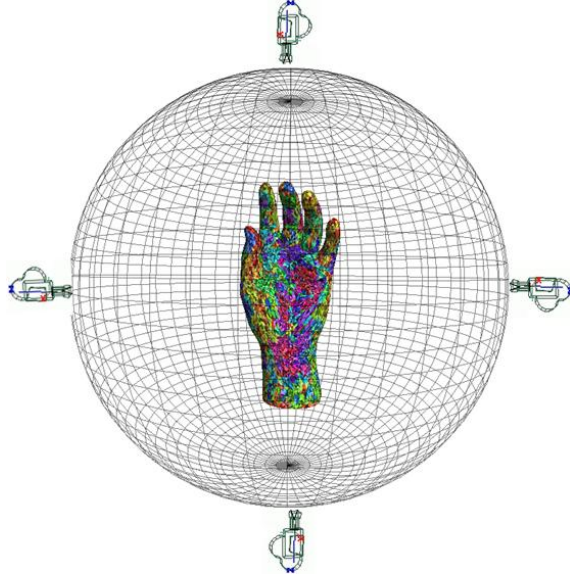


Figure 5.1: Hand model shown with unique colors for each face, used for viewpoint entropy calculations. Four of initial camera points are also presented.

entropies.

5.2 Mesh Saliency and Mean Curvature

Mesh Saliency is the concept of regional importance, which can be specified as a perception based metric for mesh processing and viewing. Center-surround operator is used to find regions that are unique relative to their surroundings [15]. Mesh saliency can be used to compute viewpoints for capturing representational images of 3D models, and mesh simplification. It depends on the surface curvature, which can be calculated by the surface normals. Our implementation uses Taubin’s mean curvature calculation approach that depends on the triangulated surfaces [68].

A triangulated surface is usually represented as a pair of lists $S = \{V; F\}$, a list of vertices $V = \{v_i : 1 \leq i \leq n_V\}$, and a list of faces $F = \{f_k : 1 \leq k \leq n_F\}$. Each face $f_k = (i_k^1, i_k^2, i_k^3)$ is a term of non-repeated indices of vertices, that represents itself a three dimensional triangle. The set of vertices that share a face with v_i is denoted as V^i . If the vertex v_j belongs to V^i , then v_j is a neighbor of v_i . The number of elements of the set V^i is denoted with $|V^i|$. The set of faces that contain vertex v_i is denoted with F^i . If the face f_k belongs to F^i , then f_k is incident to v_i .

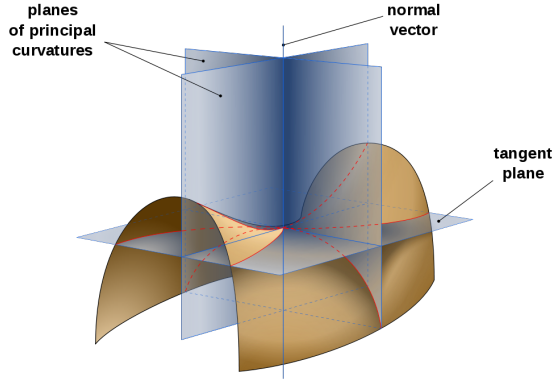


Figure 5.2: Surface normal, tangent plane and principal curvatures of the surface.

The number of elements of the set F^i will be denoted with $|F^i|$.

Mesh saliency depends on the principal curvatures, which also depends on the surface normals as shown in Figure 5.2. Principal curvatures and principal directions are obtained by computing in closed form the eigenvalues and eigenvectors of certain 3×3 symmetric matrices defined by integral formulas, and closely related to the matrix representation of the tensor of curvature using Taubin's formulations for the triangulated surfaces [68].

In order to calculate the principal curvatures, the first task is to calculate the normal vectors at the vertices of the surface. The faces of the surface are planar, and each face f_k has a well defined unit length normal vector N_{f_k} and normal vectors point to the same side of the surface. The normal vector at a vertex v_i is defined as the normalized weighted sum of the normals of the incident faces, with weights proportional to the surface areas of the faces as shown in equation(5.2).

$$N_{v_i} = \frac{\sum_{f_k \in F^i} |f_k| N_{f_k}}{\|\sum_{f_k \in F^i} |f_k| N_{f_k}\|} \quad (5.2)$$

The calculation of eigenvalues and eigenvectors is done with the approximation matrix shown in 5.3.

$$\widetilde{M}_{v_i} = \sum_{v_j \in V^i} w_{ij} \kappa_{ij} T_{ij} T_{ij}^t \quad (5.3)$$

For each neighbor v_j of v_i , T_{ij} is defined as the unit length normalized vector of $v_j - v_i$ onto the tangent plane $\langle N_{v_i} \rangle^\perp$

$$T_{ij} = \frac{(I - N_{v_i} N_{v_i}^t)(v_j - v_i)}{\| (I - N_{v_i} N_{v_i}^t)(v_j - v_i) \|} \quad (5.4)$$

The approximate directional curvature $\kappa_{ij}(T_{ij})$ is defined in equation(5.5).

$$\kappa_{ij} = \frac{2N_{v_i}^t(v_j - v_i)}{\|v_j - v_i\|^2} \quad (5.5)$$

The weight w_{ij} is chosen to be proportional to the sum of surface areas of all the triangles that are incident to both vertices v_i and v_j . The proportionality constant is set to make the sum of all weights in the neighborhood of vertex v_i equal to one.

$$\sum_{v_j \in V^i} w_{ij} = 1 \quad (5.6)$$

The normal vector N_{v_i} is eigenvector of matrix \widetilde{M}_{v_i} associated with eigenvalue 0. To compute two remaining eigenpairs the matrix \widetilde{M}_{v_i} is restricted to the tangent plane $\langle N_{v_i} \rangle^\perp$ using Householder transformation [69] denoted with the Householder matrix in equation(5.7)

$$Q_{v_i} = I - 2W_{v_i}W_{v_i}^t \quad (5.7)$$

In the equation(5.7) the W_{v_i} is the unit vector used to define reflection hyperplane that is orthogonal to that plane. Let $E_1 = (1, 0, 0)^t$ be the first coordinate vector then W_{v_i} is defined as in equation(5.8).

$$W_{v_i} = \frac{E_1 \pm N_{v_i}}{\|E_1 \pm N_{v_i}\|} \quad (5.8)$$

The calculation of the eigenpairs using Householder projection is done as shown in equation(5.9).

$$Q_{v_i}^t \tilde{M}_{v_i} Q_{v_i} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \tilde{m}_{v_i}^{11} & \tilde{m}_{v_i}^{12} \\ 0 & \tilde{m}_{v_i}^{21} & \tilde{m}_{v_i}^{22} \end{pmatrix} \quad (5.9)$$

The principal curvatures are obtained using the nonzero eigenvalues of M_p as shown in equation(5.10).

$$\begin{aligned} \kappa_p^1 &= 3\tilde{m}_p^{11} - \tilde{m}_p^{22} \\ \kappa_p^2 &= 3\tilde{m}_p^{22} - \tilde{m}_p^{11} \end{aligned} \quad (5.10)$$

The mean curvature is defined as the average of principal curvatures and shown in equation(5.11).

$$S_p = \frac{\kappa_p^1 + \kappa_p^2}{2} \quad (5.11)$$

5.3 Information Coverage

There are measures other than the ones mentioned, such as visibility ratio [70], curvature entropy [71], view-dependent measures as silhouette length, silhouette entropy or topological complexity [70]. We selected viewpoint entropy to cover the polygons of the 3D object and introduced *Mesh Saliency Entropy* to have salient points along with the face coverage. Viewpoint entropy and mesh saliency entropy expose the surface area as information to the viewer, which is suitable for many visualization tools. Here we provide only a comparative analysis of our approach to the work of Vazquez to solve the best view selection problem. For more details about Vazquez’s work we refer the reader to [6, 65, 66, 7]. In their approaches they predict the middle point entropy, add only the highest predicted entropy to the view set, and use spherical triangles for middle point calculations. In our approach we compute each entropy instead of estimating it, we use binary combination of points in view set for sampling, and we employ entropy-weighted midpoints. We also exploit the Earth Centered, Earth Fixed (*ECEF*) coordinate system for camera

sampling points over the bounding sphere. The points on $(0, 0)$, $(\pi/2, 0)$, $(-\pi/2, 0)$, $(0, \pi/2)$, $(0, -\pi/2)$, $(0, \pi)$ define the initial coverage set.

In order to project the calculated latitudes and longitudes to the local xyz coordinate system of the object we used the well known spherical formula shown in equation(5.12). This formula assumes the usage of OpenGL coordinate system.

$$\begin{aligned} \mathbf{x} &= r.\cos(\lambda).\sin(\theta) \\ \mathbf{y} &= r.\sin(\lambda) \\ \mathbf{z} &= r.\cos(\lambda).\cos(\theta) \end{aligned} \tag{5.12}$$

The differences stated above provided us with more view point samples on viewing-sphere, which outputs a viewpoint with higher polygon coverage.

5.3.1 Greedy N-Best View Selection

Best View Selection algorithm is modified for N-Best View Selection to take the previously covered faces as input and to return the currently covered faces as output. The viewpoint entropy computation is also changed not to include the pixels from already visited faces. It works as continuously calling the *Best View Selection* with supplying the face coverage set in each call. The output of the algorithm is the faces that are covered along with a selected viewpoint and entropy value for that iteration. In each call of the *Best View Selection* the returned faces are added to face coverage set. The algorithm terminates when it can not return any newly covered faces or predefined number of camera points are found. The algorithm steps are visualized in Figure 5.3.

Algorithm starts from initial sample points and navigates around the object on each best view selection call to find the best viewpoint, whereas Vazquez et al. [66] performs entropy re-computation only for already computed viewpoints. Our method resembles to finding the best viewpoint of non-visited faces for each iteration and therefore can be called Greedy N-Best View Selection.

Since in our greedy approach we do not include the visited faces with already computed viewpoints into entropy computation, we changed the contribution of the

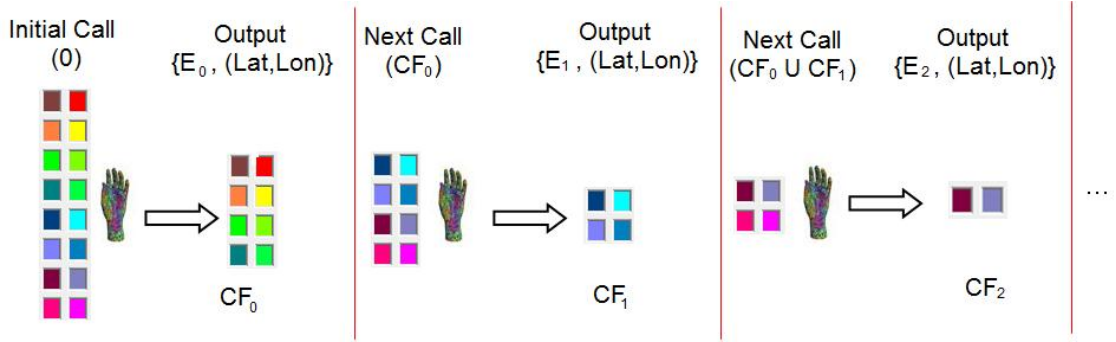


Figure 5.3: Greedy approach for best view selection. CF stands for covered faces, E for entropy and $Lat-Lon$ for latitude and longitudes over the sphere. Three dots show the continuous call of the algorithm till the termination. In the initial step algorithm is called with empty set, hence \emptyset . In the following steps CF includes all faces covered so far.

background pixels. For each viewpoint entropy computation, we set the background as the total number screen pixels hence $A_0 = A_t$, and remove the number of pixels for the faces that are involved in entropy calculation. The contribution of background into the entropy computation is included after all pixels are processed from the frame buffer object. This mechanism can provide us a greedy technique along with consistent entropy definition.

We compared the results of our method with the technique provided by Vazquez et al. in [6] and [7]. The method introduced by Vazquez is used to have maximal face or object coverage in a scene with the help of Viewpoint Entropy. It uses regularly placed viewpoints on the viewing sphere to calculate the viewpoint entropies and faces covered along with that viewpoint. The computed viewpoint entropies are sorted in decreasing order to select the viewpoints from the best to the worst. Algorithm proceeds with adding the viewpoints into a set and calculating the faces covered so far to have a terminal state. During the implementation of the technique stated in [6] and [7] we have used total 420 sampling viewpoints which regularly sampled viewing sphere with equal $\Delta\lambda$ and $\Delta\theta$.

In Figure 5.4 we compare the output of viewpoints for the teapot model. The computed five viewpoints on top row are from the approach presented in [6] and [7], and the bottom row presents the images from our greedy technique. The displayed

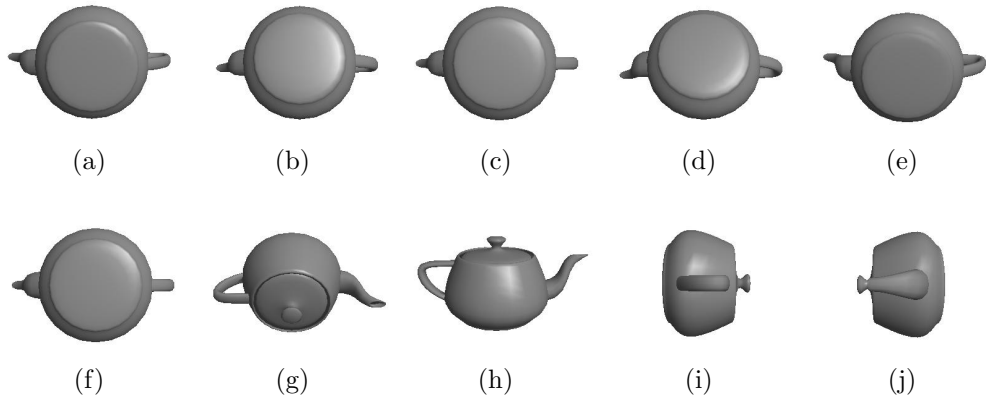


Figure 5.4: Teapot is displayed with five viewpoints using the approach from [6] and [7] compared to our greedy method. Images (a)-(e) cover 813 faces of total 2256 faces. However our method shown in (f)-(j) covers 2200 faces with provided views.

teapot is a fairly small 3D model with 2256 faces. However when we analyze the results we observe that face coverage on top row of Figure 5.4 is not that good, because what we can see is only the back of teapot with the provided images. Hence 813 faces are covered with that method, however with the greedy approach 2200 faces are covered by the provided viewpoints. In Figure 5.5 the outputs for Stanford Bunny model are compared. This model has total 69743 faces. Figure 5.5.(a)-(e) cover the 63748 faces of the model, however Figure 5.5.(f)-(j) can cover 68674 faces. We can say that the face or object coverage technique from [6] and [7] does fairly better than the teapot model. In Figure 5.6 and Figure 5.7 we compare the output of viewpoints for armadillo and dragon models. Armadillo model has total of 50000 and dragon model has total of 49755 faces. The top row from Figure 5.6 can cover only 20103 faces, the images displayed on the bottom row of Figure 5.6 cover 42009 faces. The results for dragon model are 36965 faces for the Figure 5.7 top row and 41911 faces for the bottom row.

In Figure 5.8 the outputs of the hand model are compared. We can observe from the images presented that our greedy technique performs better.

The results from both displayed images and face coverage analyses show that the brute force technique presented in [6] and [7] do not react stable to the input models. For instance in teapot or armadillo model it can only cover half of the faces

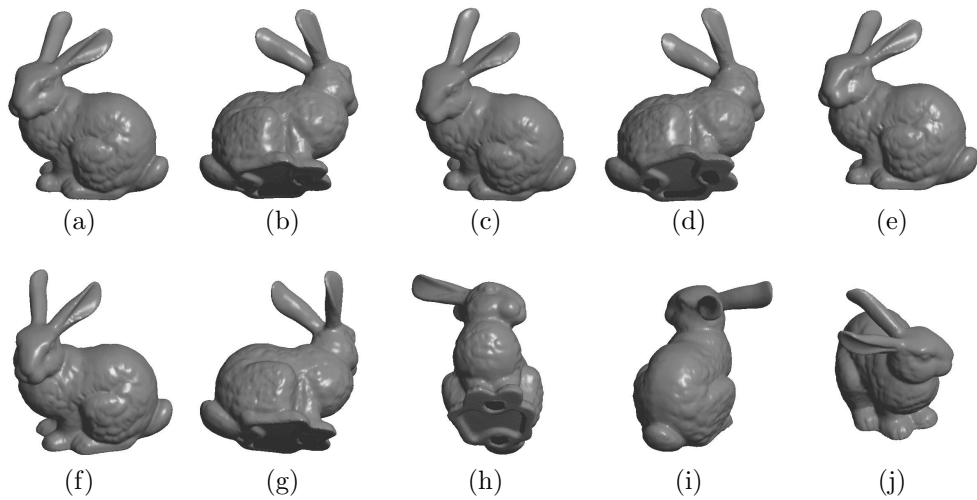


Figure 5.5: Stanford Bunny is displayed with five viewpoints using the approach from [6] and [7] compared to our greedy method. Images (a)-(e) cover 63748 faces of total 69743 faces. However our method shown in (f)-(j) covers 68674 faces with provided views.

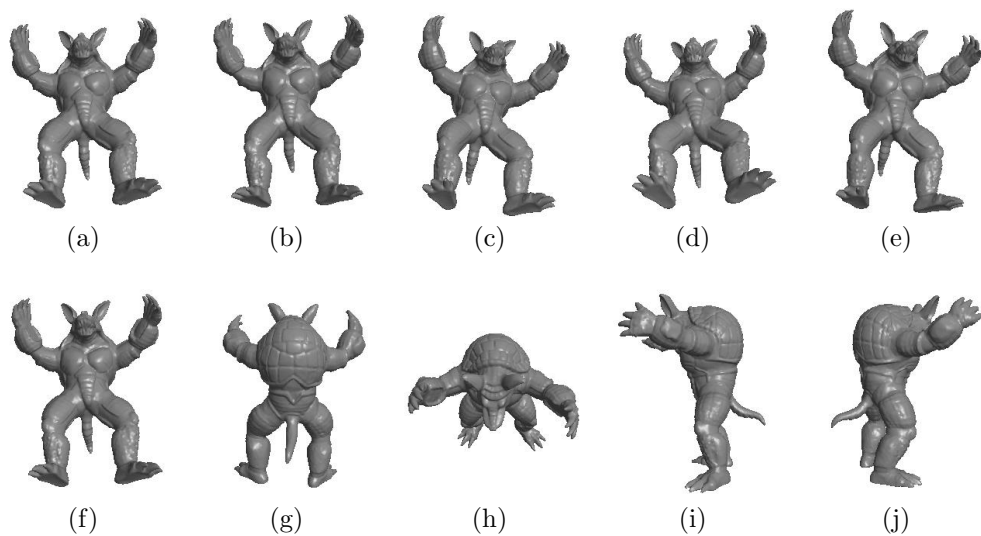


Figure 5.6: Armadillo is displayed with five viewpoints using the approach from [6] and [7] compared to our greedy method. Images (a)-(e) cover 20103 faces of total 50000 faces. However our method shown in (f)-(j) covers 42009 faces with provided views.

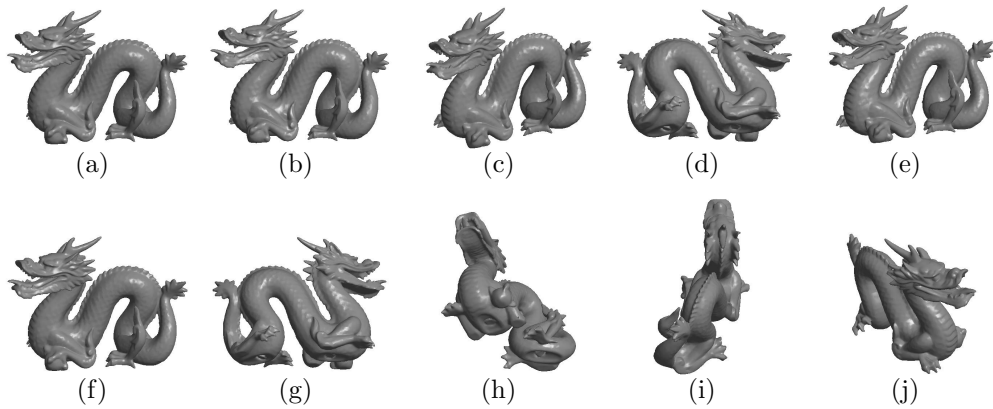


Figure 5.7: Dragon model is displayed with five viewpoints using the approach from [6] and [7] compared to our greedy method. Images (a)-(e) cover 36965 faces of total 49755 faces. However our method shown in (f)-(j) covers 41911 faces with provided views.

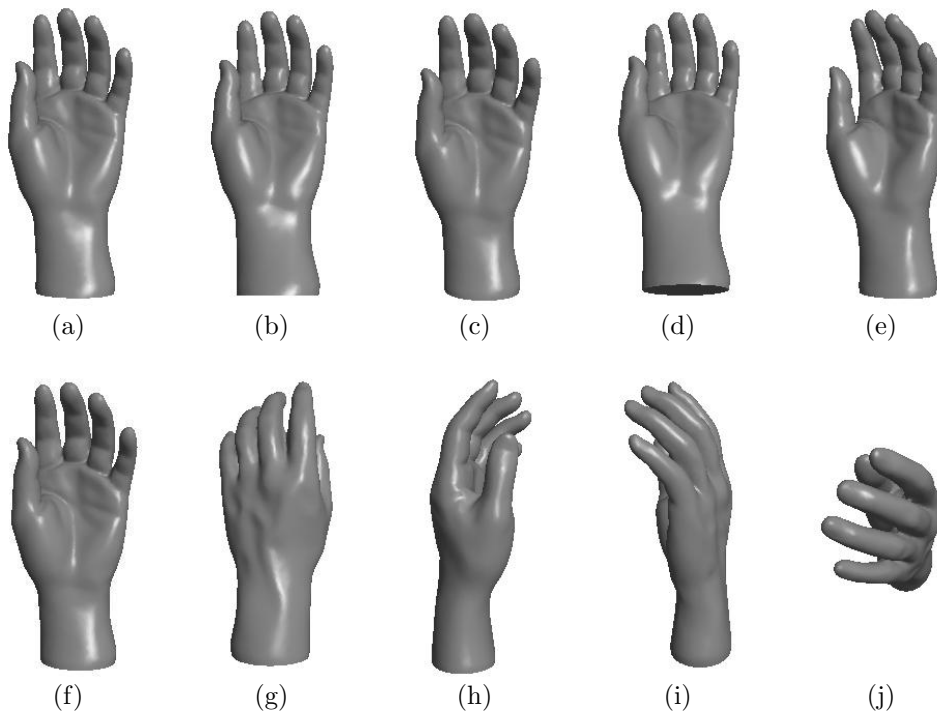


Figure 5.8: Hand model is displayed with five viewpoints using the approach from [6] and [7] compared to our greedy method. Images (a)-(e) cover 8976 faces of total 18905 faces. However our method shown in (f)-(j) covers 18406 faces with provided views.

with the provided five viewpoints, however for Stanford Bunny and dragon models, the viewpoints can cover fairly larger amount of faces. Furthermore the number of entropy calculations are fixed and depend on the sampling of the viewing sphere. It treats the teapot model with 2200 faces and Stanford Bunny model with 69743 faces equally. However our greedy technique uses 112 viewpoint entropy calculations for the teapot, and 272 viewpoint entropy calculations for the Stanford Bunny models.

5.3.2 Viewpoint Mesh Saliency Entropy

Mesh Saliency is the concept of regional importance, which can be specified as distinction in pixel colors, or luminance or geometric attributes. In our approach we borrow the techniques [42], and [15] to calculate curvature based mesh saliency. Curvature is one of the important features of a vertex which can point-out its distinctiveness among the other vertices. We use the Gaussian filtered mean curvatures of vertices proposed by [15] using Taubin’s procedure to calculate mean curvatures [68]. Meyer et al. [72] also provide a technique to calculate surface mean curvatures. Let $N(v, \sigma)$ be the set of points within a distance σ for vertex v therefore $N(v, \sigma) = \{x \mid \|x - v\| < \sigma, x \text{ is a meshpoint}\}$, and let $S(v)$ denote the surface mean curvature, hence Gaussian-weighted average of the surface mean curvature $G(S(v), \sigma)$ can be defined as;

$$G(S(v), \sigma) = \frac{\sum_{x \in N(v, 2\sigma)} S(x) \exp\left(-\frac{\|x-v\|^2}{2\sigma^2}\right)}{\sum_{x \in N(v, 2\sigma)} \exp\left(-\frac{\|x-v\|^2}{2\sigma^2}\right)} \quad (5.13)$$

In equation(5.13), a cut-off distance for the Gaussian filter is assumed to be 2σ . The saliency for vertex v is the absolute difference between coarse and fine scales, where the coarse scale standard deviation is twice of the fine scale. Then the saliency for vertex v for multiple scales is,

$$M_i(v) = |G(S(v), \sigma_i) - G(S(v), 2\sigma_i)| \quad (5.14)$$

where σ_i is the standard deviation of the Gaussian filter at scale i . We used five scales that are mentioned in [15] with this work. After the calculation of curvature saliency

for five different scales we linearly added those feature maps after the normalization method proposed by Itti et al. [42] hence denote $M(v)$. The calculated feature map for a hand model is shown in Figure 5.9(a), where the hot colors show the high salient points.

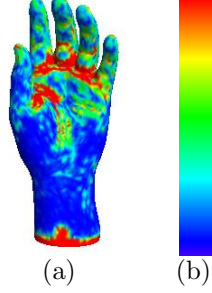


Figure 5.9: Mesh saliency for a hand model shown in (a). HSV color model shown in (b) is used to mark the saliency of the vertices. Hot colors (red) $Hue=0$ shows the highest saliency, and $Hue=240$ for the lowest. *Saturation* and *Value* are kept fixed in distribution.

Viewpoint Mesh Saliency entropy is defined as the entropy of the scene from a selected viewpoint using the saliency distribution as the probability mass function. Hence our calculations use the faces displayed to the viewer, we need to distribute saliency information $M(v)$ calculated for each vertex to the faces of the model. For this operation we use the vertex-face adjacency information. Let us denote the face (triangle in our case) saliency with $\mathcal{S}(F)$, number of adjacent faces to vertex v as $||Adj(v)||$, and vertices of a face as F^v , hence saliency of a face can be defined as,

$$\mathcal{S}(F) = \sum_{i=1}^{F^v} \frac{M(v_i)}{||Adj(v_i)||} \quad (5.15)$$

In Figure 5.10 an example triangulation of a surface is presented. This surface is constructed from four faces and five vertices. We demonstrate the distribution of vertex saliency quantity to the faces as,

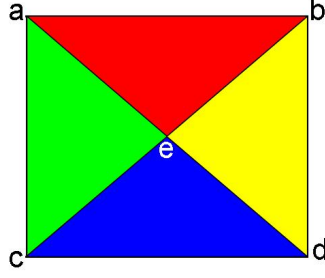


Figure 5.10: An example of triangulated surface for vertex to face saliency distribution.

$$\mathcal{S}(F_r) = \frac{M(v_e)}{4} + \frac{M(v_a)}{2} + \frac{M(v_b)}{2} \quad (5.16)$$

$$\mathcal{S}(F_g) = \frac{M(v_e)}{4} + \frac{M(v_a)}{2} + \frac{M(v_c)}{2} \quad (5.17)$$

$$\mathcal{S}(F_b) = \frac{M(v_e)}{4} + \frac{M(v_c)}{2} + \frac{M(v_d)}{2} \quad (5.18)$$

$$\mathcal{S}(F_y) = \frac{M(v_e)}{4} + \frac{M(v_b)}{2} + \frac{M(v_d)}{2} \quad (5.19)$$

The saliency entropy for viewpoint p is calculated from the visible faces of the model. The total saliency quantity from the viewpoint p is presented in equation(5.20) and saliency entropy for that viewpoint from the surface S is shown in equation(5.21).

$$\mathcal{S}_t = \sum_{i=1}^{N_f} \mathcal{S}(F_i) \quad (5.20)$$

$$\mathbf{I}(\mathbf{S}, \mathbf{p}) = - \sum_{i=1}^{N_f} \frac{\mathcal{S}(F_i)}{\mathcal{S}_t} \log_b \frac{\mathcal{S}(F_i)}{\mathcal{S}_t} \quad (5.21)$$

In equation (5.21), N_f stands for the number of faces, $\mathcal{S}(F_i)$ is the saliency of the face F_i , \mathcal{S}_t is total saliency of the visible faces from viewpoint p and b is the base of logarithm which is $b = 2$ in our case. Our definition of the mesh saliency entropy is consistent with Shannon's entropy where $\sum_{i=1}^n p_i = 1$

We modified Greedy N-Best View Selection to use either projected face area (Viewpoint Entropy), or saliency information (Viewpoint Mesh Saliency Entropy) or combination of both for viewpoint calculation and presentations. We will discuss about the combined approach in next subsection.

5.3.3 Combined Approach

Viewpoint Mesh Saliency Entropy is combined with Viewpoint Entropy in Greedy N-Best View Selection to cover both surface area and surface curvature information. The combined entropy for a given viewpoint p on the surface S can be specified as the product or summation of the two quantities. However, due to the differences in the magnitudes of the calculated information quantities we selected the product as the aggregation method. This aggregation approach favors the viewpoints that let viewers to perceive both projected surface area and saliency information provided by the model. The aggregation method for combined entropy is shown in equation(5.22).

$$\mathbf{Combined(S,p)} = I_f(S,p) \cdot I_s(S,p) \quad (5.22)$$

In equation(5.22), $I_f(S,p)$ denotes the face coverage information hence *Viewpoint Entropy*, and $I_s(S,p)$ denotes the saliency coverage information hence *Mesh Saliency Entropy*. Greedy N-Best View Selection can use either one of $I_f(S,p)$, or $I_s(S,p)$, or $Combined(S,p)$ quantities during the traversal on bounding sphere for optimal viewpoint search. These approaches are exercised and the outputs are presented in Section 5.4.

5.4 Results and Statistical Output

We have tried three different setups using our Greedy N-Best View Selection. The first one takes only projected surface area information i.e. viewpoint entropy into account, the second experiment uses the mesh saliency entropy and the final one employs the combination of both for view selection. We observed that the outcome

of the combined approach was tend to maximize the face coverage as well as salient points. The method we presented here for mesh saliency entropy uses the surface curvature, but it can be any other feature such as texture or luminance that belongs to an object.

The snapshots from the comparison of the three approaches are shown in Figure 5.11, in Figure 5.12, in Figure 5.13 and in Figure 5.14 respectively. In the displayed figures, the first row, hence (a)-(e) provides face area coverage maximization, the second row (f)-(j) provides saliency coverage maximization and the third row (k)-(o) displays the maximization of both face area and saliency. When we compare the figures row by row we can observe the differences of the calculated viewpoints. However the hand model emphasize the distinction less when compared to the other models.

The numerical results for face and saliency coverages are presented in Table 5.1, in Table 5.2 and in Table 5.3 respectively. Table 5.1 presents the results for the cumulative face coverage ratio using viewpoint entropy with our Greedy N-Best View Selection method, Table 5.2 also shows the cumulative face coverage ratio but in this case combined entropy is used in the greedy technique. We present the cumulative saliency coverage ratio in Table 5.3. The computation of saliency coverage is conducted with the viewpoints calculated in combined approach.

In the displayed figures we presented two highly self-occluded models on purpose, i.e. heart and brain. The reason for presenting those models is to give a legitimate explanation that we can not find a set of camera points that covers all the faces for all models. The face or saliency coverage depends on the visibility of the surfaces, where some parts of that surface can not be visible in some cases.

Here we discuss about the face coverage ratio contribution of the viewpoints. In Figure 5.15 a Stanford Bunny with 69743 faces is displayed using five selected viewpoints. As seen from Table 5.2 these views cover 98.85% of faces of the object. Figure 5.15.a covers 43.39% of the total faces. The contribution of Figure 5.15.b is 43.04%, Figure 5.15.c is 6.49%, Figure 5.15.d is 4.78% and Figure 5.15.e is 1.15%.

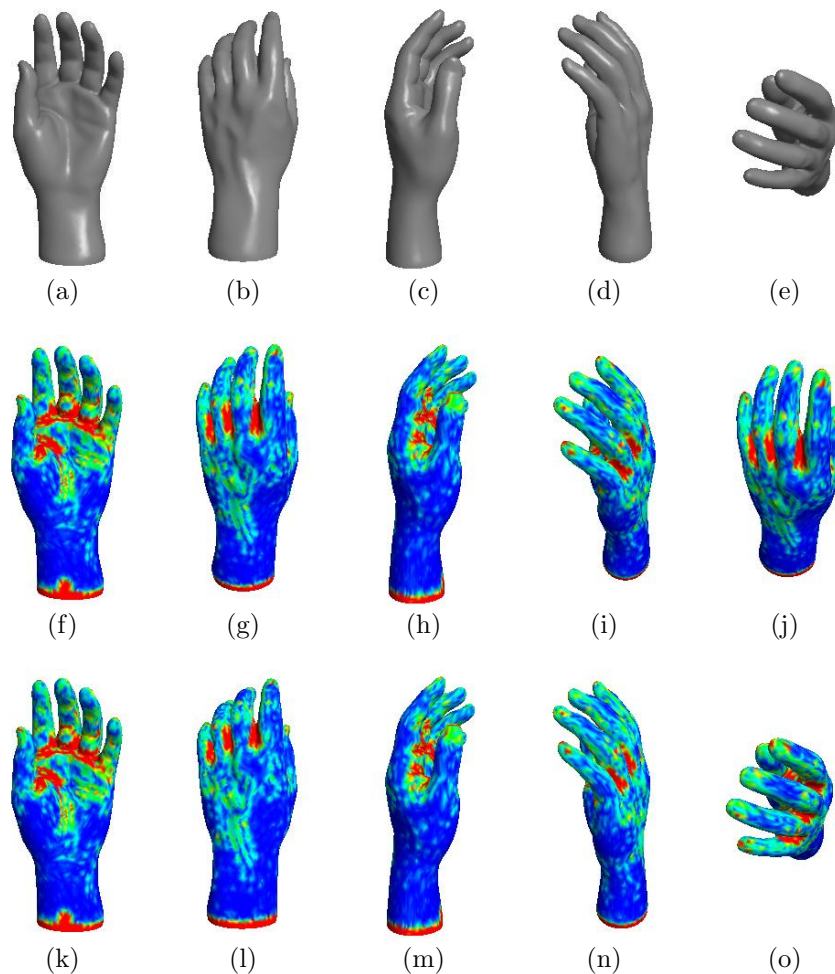


Figure 5.11: Hand shown from five viewpoints using face area maximization maximization (a)-(e), saliency coverage maximization (f)-(j) and combined approach (k)-(o). For each approach the figures are ordered from the most contribution to the least.

As we can notice the contribution of the first viewpoint starts with around 40% for non-self-occluded models and cumulative contribution speed decreases as each new view is added. Cumulative mesh saliency ratio for the models is also provided in Table 5.3. We can see that 99.09% of Stanford Bunny model saliency information is covered with the combined approach. In order to analyze how the face coverage is perturbed with the combined approach we can compare the results shown in Table 5.1 and Table 5.2 respectively. For instance the cumulative face coverage of Stanford Bunny model using surface area only entropy is 98.46% that is 98.85% in the combined method, which shows the area coverage is not perturbed.

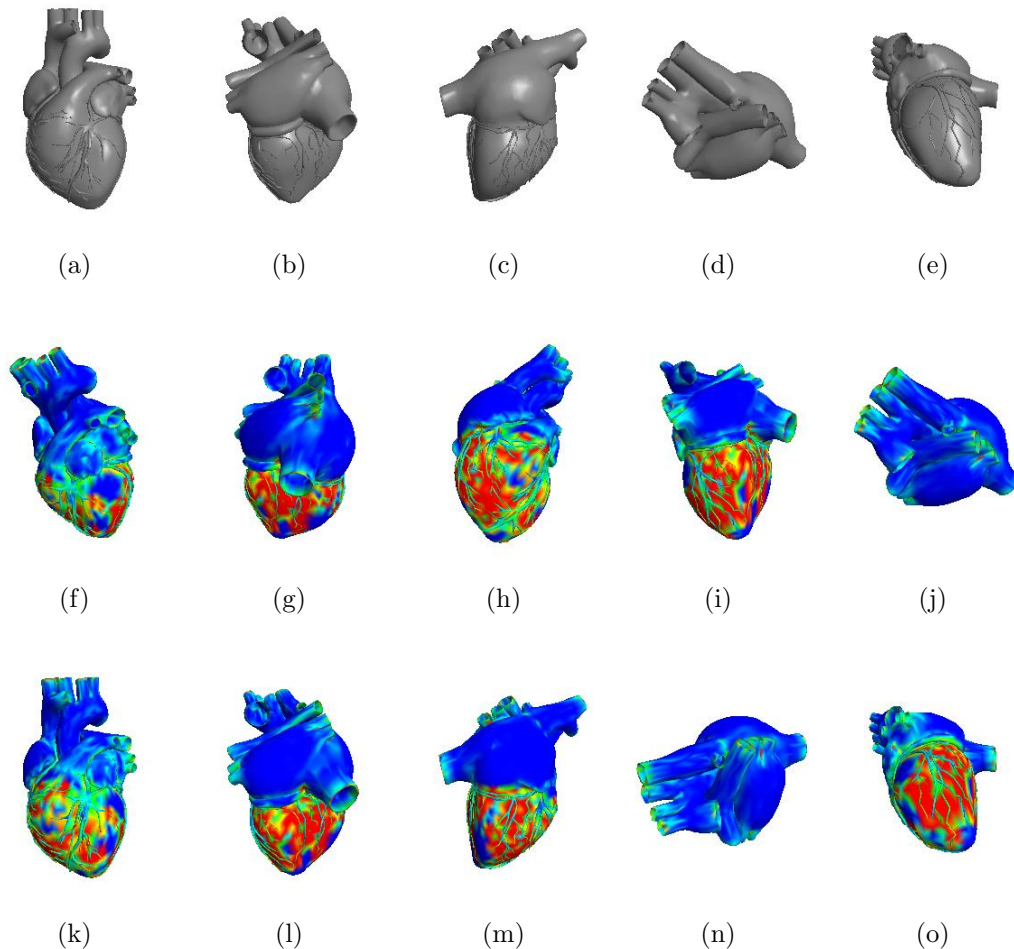


Figure 5.12: Heart shown from five viewpoints using face coverage maximization (a)-(e), saliency coverage maximization (f)-(j) and combined approach (k)-(o). For each approach the figures are ordered from the most contribution to the least.

Table 5.1: Cumulative face coverage contribution ratio of the viewpoints for different models using Greedy N-Best View Selection and taking surface area entropy into account.

Camera	Bunny	Hand	Dragon	Heart	Brain
1	43.37%	40.69%	33.47%	14.26%	8.80%
2	85.42%	80.23%	69.90%	25.77%	22.39%
3	93.81%	87.82%	76.40%	32.32%	31.78%
4	97.73%	95.09%	82.13%	35.09%	35.85%
5	98.46%	97.36%	84.23%	38.20%	39.92%

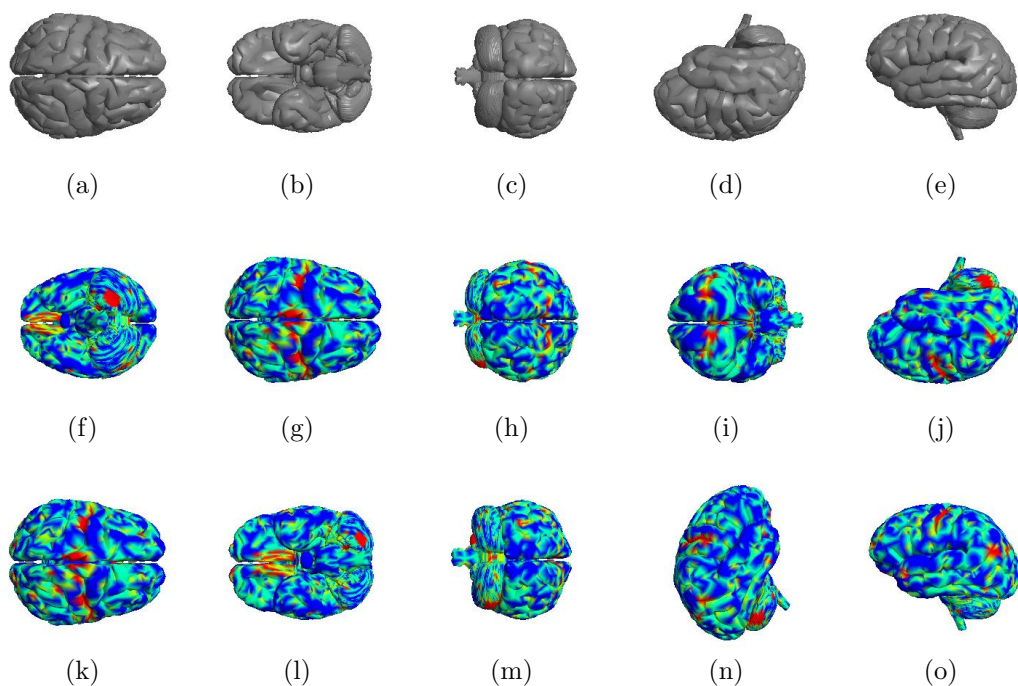


Figure 5.13: Brain shown from five viewpoints using face coverage maximization $(a)-(e)$, saliency coverage maximization $(f)-(j)$ and combined approach $(k)-(o)$. For each approach the figures are ordered from the most contribution to the least.

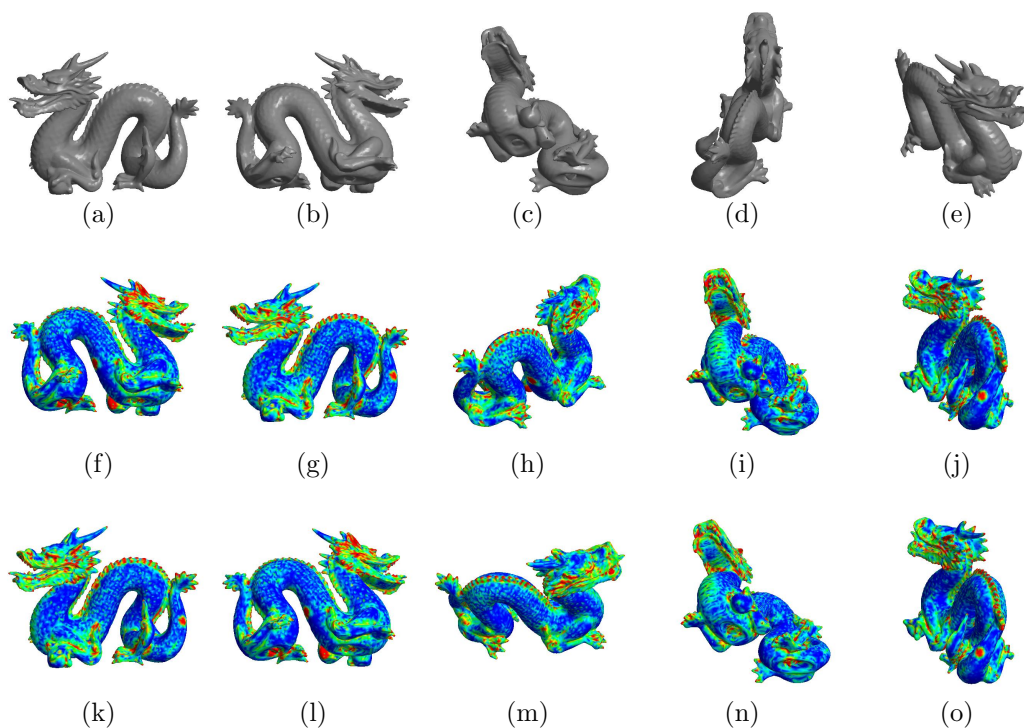


Figure 5.14: Dragon shown from five viewpoints using face coverage maximization $(a)-(e)$, saliency coverage maximization $(f)-(j)$ and combined approach $(k)-(o)$. For each approach the figures are ordered from the most contribution to the least.

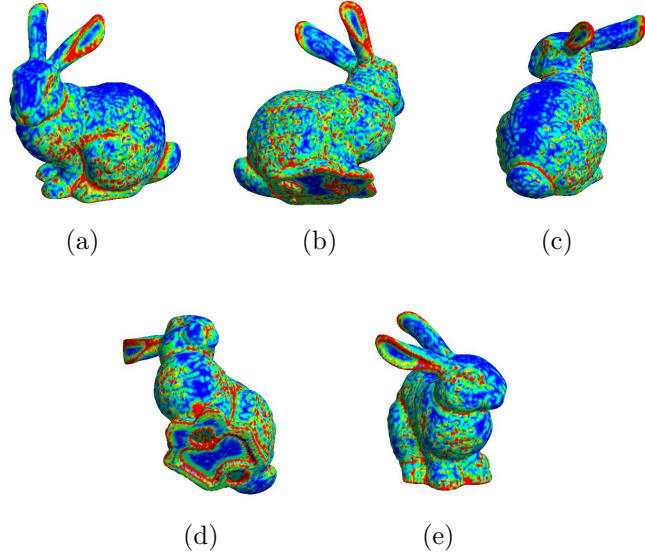


Figure 5.15: Stanford Bunny shown from five viewpoints using the combined approach i.e face and salient point coverage are maximized. The figures are ordered from the most (a) contribution to the least (e).

Table 5.2: Cumulative face coverage contribution ratio of the viewpoints for different models using Greedy N-Best View Selection and taking combined entropy into account.

Camera	Bunny	Hand	Dragon	Heart	Brain
1	43.39%	40.69%	33.52%	14.26%	8.80%
2	86.43%	80.23%	69.93%	25.88%	22.57%
3	92.92%	87.82%	76.20%	32.58%	32.11%
4	97.70%	95.35%	81.42%	35.40%	36.19%
5	98.85%	97.46%	84.62%	38.27%	40.28%

Table 5.3: Cumulative saliency coverage contribution ratio of the viewpoints for different models using Greedy N-Best View Selection.

Camera	Bunny	Hand	Dragon	Heart	Brain
1	42.31%	40.51%	33.71%	13.63%	8.84%
2	84.00%	80.23%	80.28%	24.82%	22.25%
3	91.72%	87.89%	76.53%	31.28%	31.67%
4	97.83%	95.49%	81.67%	33.92%	35.78%
5	99.09%	97.56%	84.81%	36.79%	39.89%

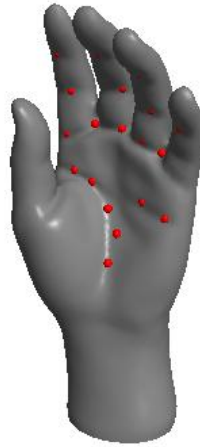


Figure 5.16: Hand model shown with red spheres used for visually queueing user selected points.

5.5 Usability Study

A simple usability study is conducted to measure the effectiveness of our technique and evaluate user tendencies for salient points interests. A group of 15 university students has participated to the study who has daily computer usage skills. The task to be completed by users was to place 20 points on the model where they were interested in most. The users were able to freely rotate/orient and zoom in/out the model shown to them. Hand model shown in Figure 5.16 is displayed with gray color, and users were visually queued by the small red spheres on the surface where they double-clicked. Each selected point is recorded, and analyzed at the end of task completion. We conducted two analyses on the user selected points. The first analysis was to find the coverage ratio of the user selected points by the viewpoints provided by our algorithm. When the results were analyzed, we observed that the face coverage of the user selected points was 100% for all participants using viewpoints from our technique. The second analysis was to find the average of the saliency for the selected points. The saliency mean of the points selected by each is user is shown in Figure 5.17. The surface saliency mean is denoted by the red circle and user values with blue. Those analysis provided us with preliminary feedback about the user interested points coverage by the viewpoints calculated



Figure 5.17: Mean saliency of the user selected points and surface saliency mean are shown. The surface saliency mean is denoted by the red circle and user values with blue. Note that the user selected points are higher than the surface mean which does not contradict with the knowledge in literature about user tendencies for salient points.

by our algorithm, and a primitive answer for the question that users are mostly interested in salient points on the model presented to them. We can present a reasoning that it does not contradict with the knowledge in literature because the mean saliency of the user selected points are greater than the surface mean saliency except one user. It can be summarized as user tendencies are somewhat towards the salient points.

5.6 Conclusion

In this thesis we presented a technique to inspect a 3D object in a scene with minimal loss of information where the information is modeled as faces and mesh saliency. We used the concept of the viewpoint entropy and introduced a greedy approach to solve N-Best View Problem. We also presented a novel view descriptor named *Mesh Saliency Entropy* to select the viewpoints in such manner to cover salient points along with the face coverage maximization.

We combined the viewpoint entropy and mesh saliency entropy in our *Greedy N-Best View Selection* algorithm to explore the object in 3D scene via minimal set of camera positions. We also conducted a usability study to evaluate the effectiveness of our approach and to measure user tendencies for salient points on a model. The results collected from the usability study showed us that the face coverage of the user selected points was 100% for all participants using the viewpoints calculated by our technique. For the knowledge about user tendencies towards salient points, we presented a reasoning that it did not contradict with the literature. Hence the mean saliency of the user selected points were greater than the surface mean saliency.

Our experiments and studies have shown that Shannon's entropy is a promising tool to solve viewpoint related problems by providing a measure to *quantify* the information on the communication channel between the user and the visual world in computer.

6 AUTOMATED TERRAIN NAVIGATION

Navigation in 3D terrain is considered to be a challenging task and requires virtual camera control skills such as zooming, panning and tilting. Novice users can easily get distracted and disoriented which may result with lost in space. Methods to overcome the virtual environment exploration problems are still being researched to assist users during their journey inside virtual environments. Assisted camera control techniques require viewpoint computation and path planning. This chapter introduces a novel approach to navigate over a 3D terrain with minimal loss of information.

We exploit the concept of the Viewpoint Entropy for the best view determination and use our Greedy N-Best View Selection for visibility calculations. We integrate road network data to extract regions for detailed visibility analysis in subsections of the terrain. In order to connect the calculated viewpoints an evolutionary programming approach for Traveling Salesman Problem(TSP) is used where the distance objective is minimized. The generated tour is presented using Google Earth framework for terrain exploration where we can get real data streams.

The computed and planned viewpoints reduces human effort when used as starting points for scene exploration or generating the representative images of the terrain dataset. The proposed framework can be integrated into 3D game engines or urban visualization systems to give quick glimpse or tour of the environment for the novice users without the help of prior planning.

The rest of the chapter is organized as follows: in Section 6.1 and in Section 6.2 we present the details of computations, in Section 6.3 we elaborate on how the calculated tour is exported to Google Earth for presentation and in Section 6.4 we discuss about the results and present the images generated, and finally we conclude

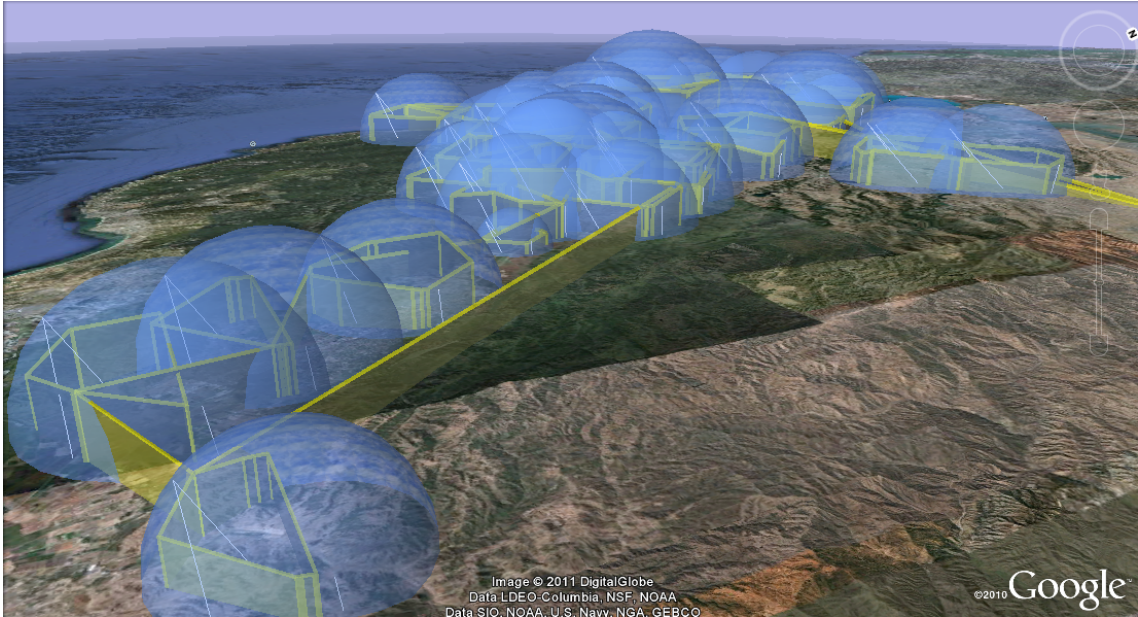


Figure 6.1: An automatically generated path by our algorithm for San Francisco shown in Google Earth framework.

our work with our remarks at the end.

6.1 Scene Analysis and Path Generation

Our method employs the divide and conquer metaphor for the scene analysis. It utilizes the help of the road network data to extract sub-regions, and calculates sub-optimal viewpoints for the regions and exploits the genetic TSP algorithm for connecting the calculated viewpoints.

6.1.1 Region Extraction

The purpose of region extraction is to provide meaningful information to the user by the help of analyzing the road intersection data. We believe that the intersection points give us a heuristic about residential areas which can be considered as significant salient features of a terrain. Although the details of our camera point generation and path construction algorithm will be discussed in subsequent sections, the salient points establishes the base of the analysis for sub-optimal viewpoint generation. Intersection points that form the bounding spheres are used as an enclosed space to decompose the surface to be investigated in detail.

The steps of our region extraction algorithm include the intersection point determination from road segments, intersection points grouping, creating a convex hull from the points in groups and bounding sphere generation. The generated bounding spheres are analyzed for mutual-inclusion, and the spheres that are enclosed by other spheres are removed programmatically. The road network data used for region extraction is shown in Figure 6.2 and region extraction steps are visualized in Figure 6.3. We used the line segment intersection algorithm for intersection points extraction that are considered as salient points, and Graham-Andrew Scan algorithm for convex hull determination [73].



Figure 6.2: Road network and terrain data

6.1.2 Terrain Rendering

In our application DTED Level-1 data is used for the terrain elevation. The data is preprocessed and converted to 2048 x 2048 grid Binary Terrain (BT) format where it is loaded into VTP [74] for rendering and viewpoint generation. The generated image depends on CLOD(Continuous Level of Detail) algorithm presented by Röttger et al. [75] which uses the dynamic triangulation of hierarchical quadtrees. When the viewpoint moves the triangulation changes continuously and results in a phenomenon called vertex popping. This dynamic behavior of the algorithm conflicts with Viewpoint Entropy when setting a metric to calculate. Projected face area is

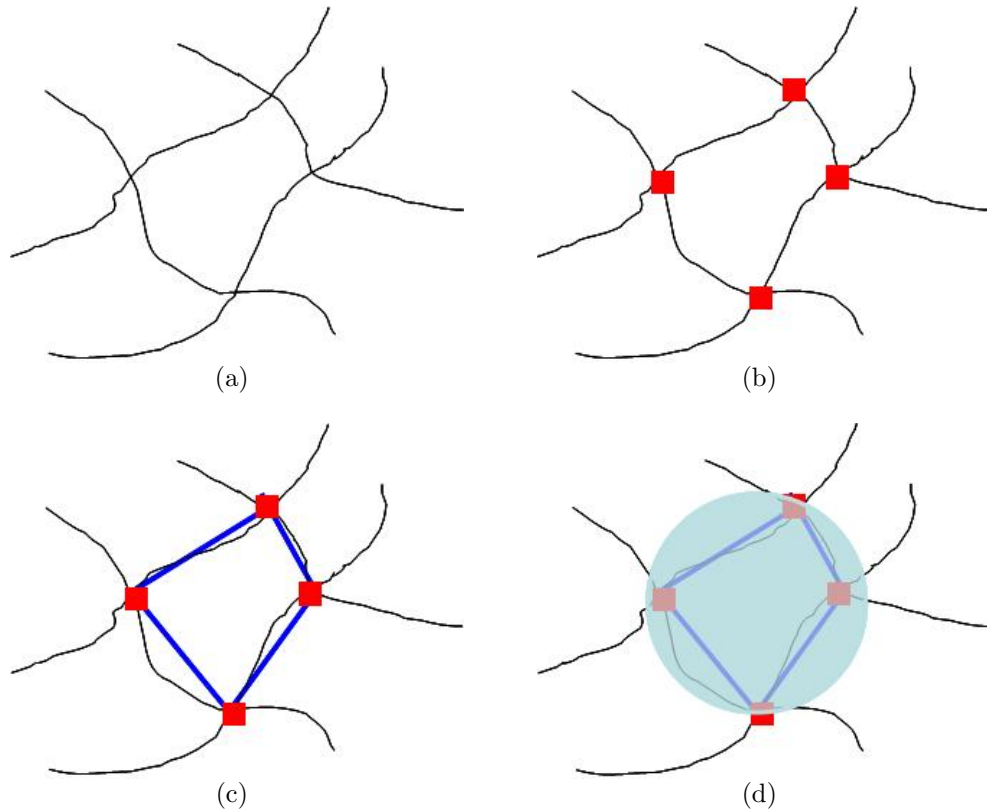


Figure 6.3: The region extraction algorithm steps are visualized. In (a) An example road network is shown, (b) Intersection points are marked with red square. In (c) the result of convex hull determination algorithm is presented. The extracted bounding circle is shown in (d)

used as probability mass function(pmf) in regular Viewpoint Entropy computation. In order to handle this problem we used texturing instead of colorization of triangles. Each texel is colored uniquely as shown in Figure 6.4 and mapped to the terrain grid. The projected texel colors are considered as the pmf during entropy computation and viewpoint generation. The sketch for vertex popping phenomenon is presented in Figure 6.5. Note that triangulation is sensitive to the current viewpoint position, when the camera shown with turquoise circle changes its position the algorithm re-generates the triangles used for rendering. In Figure 6.6.a the triangulated terrain data is shown using wireframe mode, and uniquely colored texturing to the terrain data is shown in Figure 6.6.b.

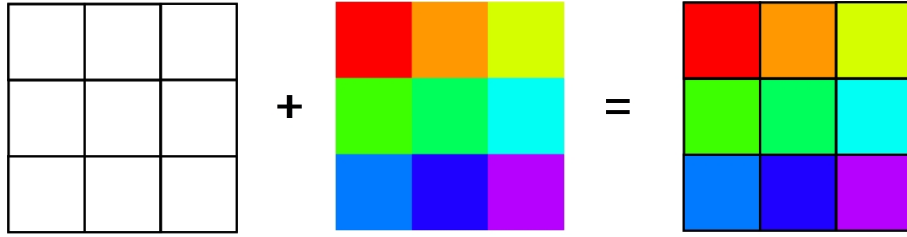


Figure 6.4: Sketch of uniquely colored texture mapping to a grid

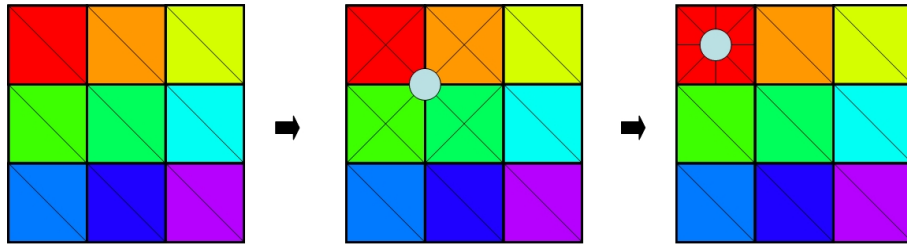


Figure 6.5: Sketch of CLOD algorithm on a grid. Camera is shown with a turquoise circle. Note that camera move changes the triangulation.

6.1.3 Best Viewpoints

The term “best” or “good” is highly subjective and difficult to quantify, and mostly depends on the application or context. Despite its subjectiveness, researchers may agree that some images created by the tessellation are more informative compared to the others using different criteria. The term informative is chosen on purpose. Because, the information amount is quantified by the term entropy. Although there are other measures such as visibility ratio quantified as the ratio of the visible 3D surface area to the total 3D surface area, curvature entropy quantified as the entropy of the Gaussian curvature distribution over the entire surface of the object, or view-dependent measures as silhouette length, silhouette entropy or topological complexity, we selected viewpoint entropy as our candidate to cover polygons of the 3D object by using a minimal set of camera points because it exposes surface area as information to the viewer.

We modified the Viewpoint Entropy calculation technique presented in [6] to utilize the usage of latitude and longitudes on spherical space. We calculated binary combination of each point in view set for midpoint calculation where they are entropy

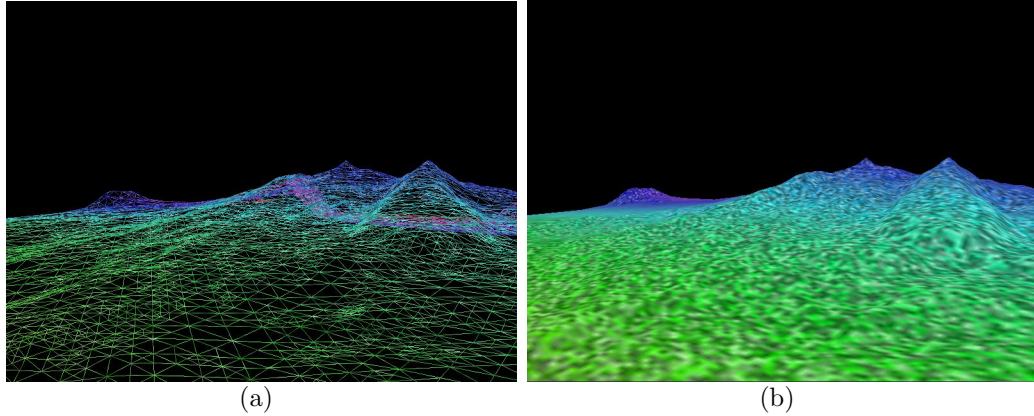


Figure 6.6: Wireframe mode for a region of terrain is shown in (a). When the camera gets closer vertex popping phenomenon occurs. In (b) the uniquely colored texturing is applied to the elevation data

weighted. Differences provided us with higher sample view points on sphere, which resulted in a viewpoint that covers as much polygon as possible. The algorithm shown below tries to find the best view of a extracted region from a single viewpoint.

(a) Cover the region by a viewing sphere, where each point on that sphere is defined by λ and θ .

(b) Place 5 sampling points on $(0, 0)$, $(\pi/2, 0)$, $(0, \pi/2)$, $(0, -\pi/2)$, $(0, \pi)$ λ and θ . This defines the initial coverage set.

(c) Compute the viewpoint entropy of the each initial point and store the maximum.

(d) Take binary combination of the coverage set, and find the weighted mid-points of them (using arc length) where weight is defined as $e_1 / (e_1 + e_2)$, and e is the viewpoint entropy.

(e) For each midpoint calculate its viewpoint entropy, if calculated entropy is higher than the current maximum add that point to the coverage set.

(f) Update the maximum entropy from the coverage set

(g) Go to (d) until no points can be added to the coverage set.

Although finding N-best view selection is known to be NP-hard, in this work we use our a greedy choice algorithm which tries to detect a sub-optimal N-best views

to perceive the information communicated by the object. The algorithm is modified to take the previously covered faces as input and to return the currently covered faces as output. The viewpoint entropy computation is also changed not to include the pixels from already visited faces.

- (a) Best view selection algorithm is called with empty polygon coverage set
- (b) Accumulate the visited faces into the set from previous best view selection algorithm
- (c) Call the best view selection algorithm with the new set
- (d) Go to (b) until all faces covered or best view selection algorithm can not output newly covered faces

The algorithm shown above starts from initial points and navigates around the object on each best view selection call. This method resembles to finding the best view of non-visited faces for each call.

6.2 Camera Path Planning

We treat the planning of a path from the calculated best viewpoints as a tour generation problem over the urban area to be visualized. The tour concept is tightly coupled with a well known NP-hard problem called Traveling Salesman Problem. Given a list of cities and their pairwise distances the task is to find a shortest possible tour that visits each city exactly once. In our urban visualization problem the cities are the calculated viewpoints for the extracted sub-regions of the terrain and the tour is a problem stated quick urban exploration. In this work we tried to present a plausible solution by optimizing the the total distance traveled.

6.2.1 Traveling Salesman Problem

The traveling salesman problem(TSP) is an NP-hard problem of combinatorial optimization studied in operations research and computer science. Given a list of cities and their pairwise distances the task is to find a shortest possible tour that visits each city exactly once [76].

Euclidian and Spherical TSP

In our framework we use two versions of TSP problem, hence Euclidian space TSP, and spherical TSP. Euclidian space TSP is used to enumerate the sequence of the extracted regions to be traveled on the texture surface. The calculated tour will have N extracted regions with M computed best viewpoints for that region. We can formulate the concept of a tour,

$$\mathbf{T} = \{R_1, R_2, \dots, R_n : n \in Z\} \quad (6.1)$$

$$R_i = \{c_1, c_2, \dots, c_m : m \in Z\} \text{ such that } R_i \in T \quad (6.2)$$

where T denotes a tour of N different regions and R_i denotes the region i on the surface of terrain.

The spherical TSP is used to enumerate the sequence of the calculated camera points in region R_i shown in equation(6.2). The difference between Euclidian space TSP and spherical TSP is the distance function used to determine length between two points. In Euclidian space the geodesic distance between two 3D points is a straight line and calculated as in equation(6.3);

$$\mathbf{dij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2} \quad (6.3)$$

However the shortest distance between two points (p_0, p_1) on a spherical surface is the arc length of the points along the *Great Circle*. So it is the angle of *alpha*(α) between two vectors \vec{v}_0 and \vec{v}_1 from the origin of sphere to $p_0(\lambda, \theta)$ and $p_1(\lambda, \theta)$ on the surface respectively with and can be calculate directly using *Haversine formula* [77]. The shortest distance on a sphere between two points is shown in equation(6.4) where R is the radius of the sphere.

$$\begin{aligned} \Delta\lambda &= \lambda_0 - \lambda_1 \\ \Delta\theta &= \theta_0 - \theta_1 \\ \mathbf{a} &= \sin(\Delta\lambda/2)^2 + \cos(\lambda_0) \cdot \cos(\lambda_1) \cdot \sin(\Delta\theta/2)^2 \\ \mathbf{c} &= 2 \cdot \arctan 2(\sqrt{\mathbf{a}}, \sqrt{1 - \mathbf{a}}) \\ \mathbf{d} &= R \cdot \mathbf{c} \end{aligned} \quad (6.4)$$

The provided distance functions are used during the execution of genetic TSP for the purposes stated above.

Genetic Approach for TSP

Genetic algorithms are one of the computational intelligence methods which are used to find approximate or sub-optimal solutions to the NP-hard combinatorial optimization problems. It is generally inspired from the biological facts and evolution. Genetic algorithms employ the concept of population, gene, crossover and mutation. Population is a set of genes in the current iteration of the algorithm, and a gene is an enumeration of a valid solution to the problem being solved. The crossover concept is inspired from inheritance of two parents, where a child carry the combination of two parent genes. The mutation can be expressed as the effect of the environmental factors over a gene. Evolution concept is applied by terminating the genes that are progressing poorly and creating new genes from a random group of successful genes where the newly created genes will do better eventually.

6.2.2 Path Planning for Intra-Regions

Best viewpoints for the extracted sub-regions are calculated by the help of our Greedy N-Best View Selection algorithm which uses modified Viewpoint Entropy technique. In this algorithm the model or the region to be explored is bounded with a sphere where the region and bounding sphere centers are aligned. Our objective is to find best viewpoints on this bounding sphere where the camera position is denoted by (λ, θ) and the up-vector is perpendicular to the viewing direction along North-pole(+Y). Due to the shortest distance between two points (p_0, p_1) on a spherical surface is the arc length of the points along the *Great Circle*, we exploited the spherical genetic approach for Traveling Salesman Problem to enumerate the tour in this region.

A gene is encoded with a valid tour that contains all the id's of the calculated camera positions. A random population of 10,000 genes are created and simulation

is run 100,000 generations where the mutation ratio is set to be 3%. Evolution concept is applied by terminating the worst two genes and creating two new genes from a random group of successful genes. An example output of the spherical genetic algorithm is shown below where two valid genes A and B are presented which show a tour over a sphere with five points.

$$\begin{aligned}
p_0 &= (0, 0) \\
p_1 &= \left(\frac{\pi}{4}, \frac{\pi}{4}\right) \\
p_2 &= \left(\frac{-\pi}{4}, \frac{-\pi}{4}\right) \\
p_3 &= \left(\frac{-\pi}{4}, \frac{\pi}{4}\right) \\
p_4 &= \left(\frac{\pi}{4}, \frac{-\pi}{4}\right) \\
A &= (p_0, p_1, p_2, p_3, p_4, p_0) = 7.33 \\
B &= (p_0, p_3, p_2, p_4, p_1, p_0) = 5.75
\end{aligned} \tag{6.5}$$

The the cost of the tour A is 7.33 on unit sphere where the the cost of tour B is 5.75. The tour B is the output of the spherical genetic TSP algorithm. In the case of not using unit sphere, the difference in the cost will increase proportionally with respect to the radius of sphere to be calculated, which complies with the need of finding a sub-optimal solution for camera enumeration. This sub-optimal enumeration of the viewpoints presents that the total traveled distance is minimized in our framework.

In order to project the calculated latitudes and longitudes to the local xyz coordinate system of the calculated region we used the well known spherical formula shown in equation(5.12).

After the calculation of enumeration and positional values of the camera points, the next task to handle for path planning is to choose a technique to travel along the curves. The spherical linear interpolation(Slerp) [78] is used which refers to constant speed of motion along a unit radius of great circle. Since our computations are done on spherical space this technique suits well for our problem design. Its constant speed of motion is natural and produces smooth animation curves which does not distract the users perception.

6.2.3 Path Planning for Inter-Regions

The path between the extracted regions are arranged using Euclidian TSP algorithm with evolutionary programming approach. The algorithm enumerates the sequence of the regions to be traveled by using region centers as points to be visited in a tour. Similar to genetic approach used for intra-region, a valid gene set called population is constructed. Each gene encodes all the regions to be traveled via a sequence number or region id.

The created population is run for 100,000 generations where the mutation ratio is set to be 3%. Evolution concept is also applied by terminating the worst two genes and creating two new genes from a random group of successful genes. When the simulation is done, the enumerated region centers are used to construct the Bezier curve for the camera trajectory in inter-region movement.

6.2.4 Final Camera Trajectory

The final camera path is constructed by combining the paths generated for intra and inter regions. The tour can be started from a region selected to be initial or any region that the user is interested in. The camera follows the constructed intra-region path and continues onto the next region. When the camera trajectory enters the next region it starts to follow the intra-path constructed for that region. The camera visits all the enumerated region in the same approach.

With the techniques provided with this work, we tried to present a plausible solution for an automatic camera trajectory. Best views calculated from the extracted salient points optimized the user's surface perception, and the genetic TSP algorithm enabled us to construct a path that creates an optimal tour for the terrain exploration.

6.3 Tour Presentation in Google Earth

We present automatically generated tours using Google Earth [79] framework. Even though it is possible to create tours with VTP API, Google Earth provides a better way to demonstrate a tour in a realistic and detailed 3D environment. Google Earth also enables us to define tours through geo-spatial data with the ability of smooth flight pass locations and specific flight durations between those points. The tour is mainly defined using KML file format, Google Earth's XML notation for expressing geographic annotation and visualization. With the aid of the tour generated by our algorithm, we automatically export our best viewpoints and their fly-over order into the KML document for touring actions in Google Earth. Later on, Google Earth's plugin can be used to play tours authored in a KML file.

6.3.1 Camera in KML

In order to provide a smooth fly-through over terrain dataset, camera parameters should be set properly in KML file. The format of a camera object and its parameters are stated in [80]. *Camera* and *LookAt* XML elements specify the viewpoint of observer and associated view parameters. Since both elements define the placement and orientation of a virtual camera viewing the Earth, either of them can be chosen for a path visualization. The difference is that *LookAt* specifies the view in terms of the point of interest that is being viewed. *Camera*, in contrast, specifies the view in terms of the viewer's position and orientation. We used *LookAt* object in our KML documents which is more suitable for extracted region exploration.

The necessary camera parameters are exported to KML elements as we do in OpenGL. These are; viewpoint, altitude, heading, tilt, range and altitude type. We avoid dealing with the altitude value of any point and altitude type since our camera location is not on the ground. The range defines the distance of viewpoint from the point of interest which is in our case will be the radius of an extracted region. Centers of the intra-regions will be the points of interests that is being viewed. These values

are global latitude and longitude points that are calculated by our region extraction algorithm. On the other hand, all camera viewpoints are local coordinates which means they are calculated by taking region centers as the origin of local spheres. Important part with the camera operations in KML lies in setting heading and tilt values of the virtual camera. Using coordinates of camera locations with respect to intra region spheres, heading and tilt can be calculated by the formula shown in 6.6. In this formula θ states the local longitude and λ states for the local latitude of the viewpoint calculated for a region.

$$\begin{aligned} \mathbf{Heading} &= 90 - \lambda \\ \mathbf{Tilt} &= 90 - \theta \end{aligned} \tag{6.6}$$

6.3.2 Tour Generation in KML

After camera parameters are exported for each camera location, they are combined to form a tour. Tours in KML can contain any number of *FlyTo* elements in which information about each local best viewpoint that tour flies to is stored. Other touring-related elements in KML are the timing and behavior of the tour. Timing and velocity between points is controlled by the inclusion of a *Duration* element, which defines the time that the browser takes to travel from the current point to the next defined point. In other words, once that time has elapsed tour starts flying to the next point. Since our goal is to explore the terrain efficiently, speed of the tour is an important factor for user's perception of the environment. In order to fly at a constant speed, durations between each viewpoints are calculated using the distance between two coordinates. Additionally, behavior of the flying mode is specified by *FlyToMode* which tells how to approach the point while tour is playing. We selected *smooth mode* which is suitable for this kind of controlled flight in a 3D space. An unbroken flight is made up of a series of *FlyTo*'s with smooth *FlyToModes*. The Google Earth browser interpolates the velocity and the curved path between points so that each placemark is reached exactly at the time specified in the KML.

The velocity of the first *FlyTo* contained within a playlist needs to be selected

appropriately depending on the position of the viewer in Google Earth when the tour is started. Initial position is outside of the extracted regions. A duration of five seconds is added into this *FlyTo* element with mode *bounce* for giving a feedback to the user about the start of the tour.

6.3.3 Importing the Tour Using Google Earth API

The created tours can be run in two ways using the exported KML file. The first one is to start KML manually on the local computer. The other and automated way is to fetch KML document by Google Earth API and pass it to an *GETourPlayer* object in Google Earth plugin. After the tour object is loaded into the plugin, we can set it as a currently active tour and tour control appears on the screen. Tour can be started, paused or reset with the method calls supported by Google Earth API.

6.4 Results

In this framework San Francisco Bay Area DTED data and major highways road network data is used for automatic path computation(Figure 6.2). The DTED data is a 2048x2048 grid and road network data is a set of 12084 linestrings which can be considered as real world data.

We extracted 35 regions using the extraction algorithm presented in this work and generated a complete tour with the methods presented in previous sections. In Figure 6.7 we present the extracted regions and generated path. Our technique is completely automatic and needs no user intervention.

Sketch for the generated path is shown in Figure 6.8. In this figure the circles demonstrate the path followed for intra-region viewpoints and lines show the path followed by the camera on the way from one region to the other. The radius of the sphere depends on the intersection point locations extracted from the road network data. The generated intra-region camera path resembles a circle on the sphere that bounds the region, which is consistent with the expectation from our best viewpoint

computation and the spherical TSP. The complete set of the extracted regions using Google Earth framework are shown in Figure 6.9.

Inter-region tour is shown with connecting lines in Figure 6.10. Region centers are represented with the placemarks. The length of the generated path is sub-optimal due to the usage of TSP algorithm. The complete tour starts from the first region and follows the camera points generated for that region and moves to the next region. The tour is terminated when all the viewpoints for the final region are visited.

Elevations of viewpoints vary due to the radii of the extracted regions from the network data where it can be observed from Figure 6.11.

The timing for our non-optimized application is shown in Table 6.1 for the machine with Intel Core 2 Duo T9600 2.80 GHz cpu, 4GB memory and nVidia GT 240M gpu. Viewpoint calculation denotes the total timing of camera point calculations for the 35 regions where per region average is about 30.230 secs. The value for inner region tsp shows the total duration for all regions, where the average per region is about 0.809 sec. The total duration is the elapsed time starting from loading terrain and road network data to the final KML output.

Step	Duration(sec.)
Region Extraction	1.803
Viewpoint Calculation	1058.051
Inner Region Tsp	28.333
Inter Region Tsp	2.850
KML Export	0.450
Total Duration	1091.446

Table 6.1: Timing for non-optimized application. Note that all the values are total duration of the corresponding steps.

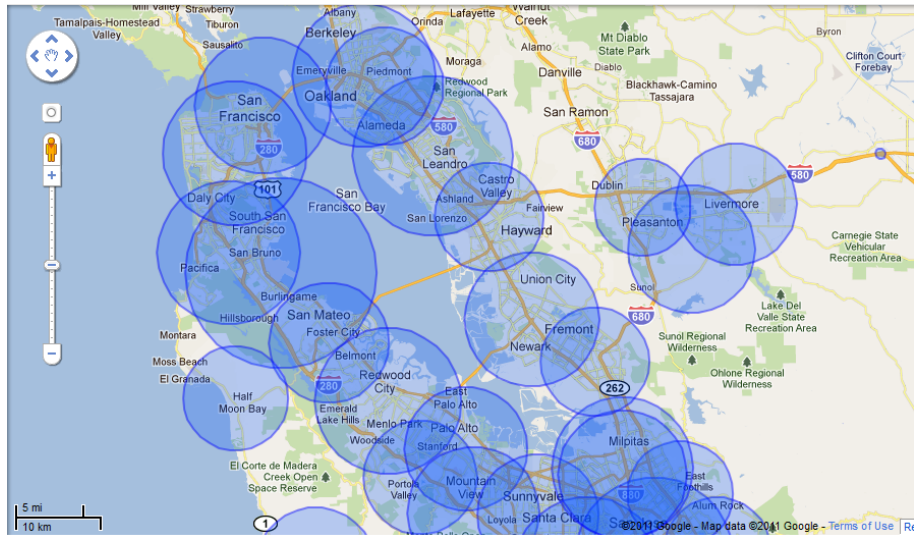
6.5 Conclusion

We present an entropy assisted solution to explore the terrain dataset effectively. Our technique can provide a quick glimpse or tour of the environment for the novice

users and can improve user perception. The computed and planned viewpoints reduces human effort when used as starting points for scene tour or generating the representative images of the terrain dataset. The proposed framework can be integrated into 3D game engines or urban visualization systems to introduce the virtual environment for the novice users without the help of prior path planning.

We tested our method using real terrain and road network dataset and exported the generated tour to visualize it with Google Earth framework.

The generated tour visualization has shown that Shannon's entropy model is a promising tool to solve viewpoint related problems.



(a)



(b)

Figure 6.7: In (a), extracted regions in San Francisco are shown by circles using Google Maps. With the aid of these regions, path is generated on the terrain(b)

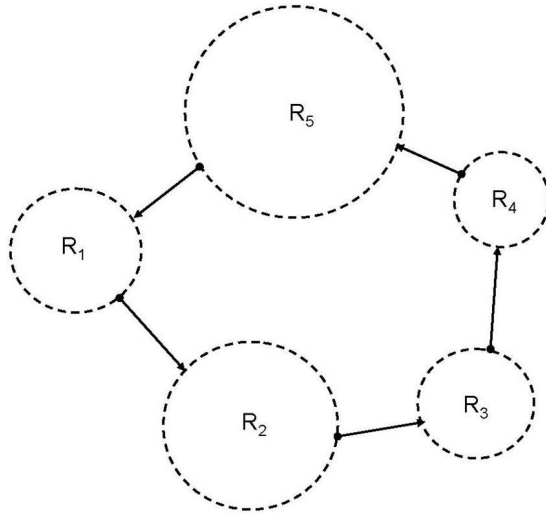


Figure 6.8: Sketch for the generated path. Note that intra-region camera path resembles circles however not exact, they are on sphere and the connection between them is an arc. Straight lines show the path for inter-regions, however the start and finish points may not be on the same plane.



Figure 6.9: Extracted regions are presented by the spheres using Google Earth framework.

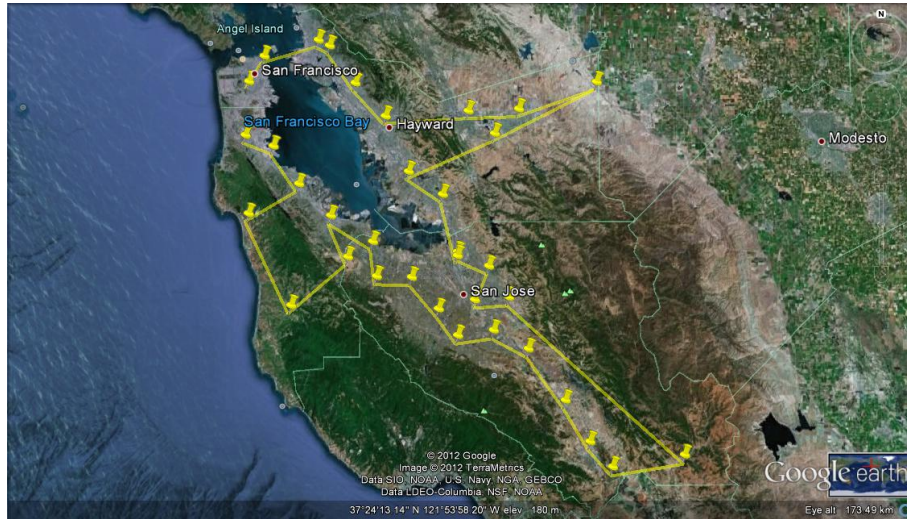


Figure 6.10: Inter-region tour shown with connecting lines using Google Maps framework. Placemarks represent the region centers.

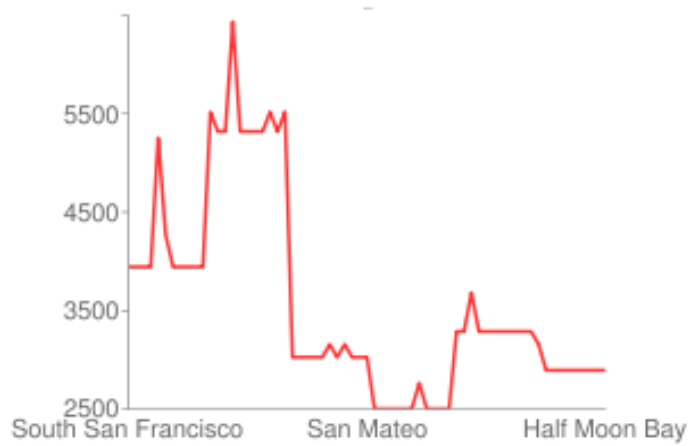


Figure 6.11: Heights(m) of viewpoints in first 3 regions for the path generated on San Francisco.

7 CONCLUSION

Information Visualization is a wide research area with the purpose to convey useful and helpful information to the users where it can ease the tasks that users try to accomplish.

In this research we developed metrics and techniques to improve the computer generated visualizations where the established metrics are used to form the basis for color, size of objects visualized on the screen and as well as finding good camera positions to enhance the user perception.

We experimented the usage of Shannon's entropy to improve the visualization of a social network with the help of network centralities such as degree, betweenness and closeness. We constructed centrality entropies, and conducted sensitivity analysis to display large scale social networks in a useful manner. We exploited coloring, sizing and filtering mechanisms. These techniques helped us to enable users to quickly understand actors and their importance in large scale networks. The importance is varied from the degree centrality, to the sensitivity analysis of the total system change. Experiments are preformed using different datasets varying from hand generated to collaboration data extracted from various sources.

We also exercised the usage of entropy to find optimum camera positions for object exploration. We employed Viewpoint Entropy and introduced *Viewpoint Mesh Saliency Entropy* as a novel view descriptor. We introduced a greedy approach to solve N-Best View Problem and combined the viewpoint entropy and mesh saliency entropy into an aggregate quantity to explore the object in 3D scene via minimal set of camera positions.

We conducted a usability study to evaluate the effectiveness of our approach and to measure user tendencies for salient points on a model. The results collected from the usability study showed us that the face coverage of the user selected points was

100% for all participants using the viewpoints calculated by our technique. For the knowledge about user tendencies towards salient points, we presented a reasoning that it did not contradict with the literature. Hence the mean saliency of the user selected points were greater than the surface mean saliency.

The usage of entropy in automatic path generation for large scale terrains is also studied. Viewpoint Entropy is used to find optimal camera positions in regions extracted from road network data. Evolutionary programming approach to connect the camera points to establish a tour over 3D terrain is exercised. Our technique provides a quick glimpse or tour of the environment for the novice users and can improve user perception. In this work we presented,

- Region extraction from real road network data,
- Conducting visibility analysis in regions and finding optimal camera points,
- Employment of evolutionary programming approach for camera path generation, and large terrain exploration.

We tested our method using real terrain and road network dataset. We presented a technique to export the generated tour into Google Earth framework for visualization.

Our work and studies during this research have shown that Shannon's entropy is a promising concept to solve visualization related problems by providing a measure to *quantify* the information on the communication channel between the user and visual world in computer.

7.1 Future Work

Although Information Visualization domain is not a new research area, it is still being widely investigated to find proper and better techniques to visualize the data or information. We believe that Information Theory is one of the right places to

investigate such metrics or techniques to enhance the current visualization systems.

In future we would like to further investigate,

- The mutual information (transinformation) which is quantity that measures the mutual dependence of the two random variables,
- The Kullback-Leibler divergence which is a non-symmetric measure of the difference between two probability distributions.

and their usability and application to visualization systems.

Bibliography

- [1] R. Mazza, *Introduction to Information Visualization*. Springer, 2009.
- [2] “A PERIODIC TABLE OF VISUALIZATION METHODS,” <http://www.visual-literacy.org> December 2011”.
- [3] C. Chen, *Information visualization - beyond the horizon (2. ed. paperback)*. Springer, 2006.
- [4] “PREFUSE: INFORMATION VISUALIZATION TOOLKIT,” <http://prefuse.org> December 2011”.
- [5] D. Phan, L. Xiao, R. B. Yeh, P. Hanrahan, and T. Winograd, “Flow map layout,” in *INFOVIS*, J. T. Stasko and M. O. Ward, Eds. IEEE Computer Society, 2005, p. 29.
- [6] P.-P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich, “Viewpoint selection using viewpoint entropy,” in *Proceedings of the Vision Modeling and Visualization Conference 2001*. Aka GmbH, 2001, pp. 273–280.
- [7] P.-P. Vázquez, M. Feixas, M. Sbert, and A. Llobet, “Realtime automatic selection of good molecular views,” *Computers & Graphics*, vol. 30, no. 1, pp. 98–110, 2006.
- [8] C. E. Shannon, “A mathematical theory of communication,” *The Bell System Technical Journal*, vol. 27, pp. 379–423, 1948.
- [9] “THE DBLP COMPUTER SCIENCE BIBLIOGRAPHY,” <http://dblp.uni-trier.de/> August 2011.
- [10] D. M. A. Hussain and D. O. Arroyo, “Locating key actors in social networks using bayes’ posterior probability framework,” in *EuroISI*, ser. Lecture Notes in Computer Science, D. O. Arroyo, H. L. Larsen, D. D. Zeng, D. L. Hicks, and G. Wagner, Eds., vol. 5376. Springer, 2008, pp. 27–38.

- [11] A. Bulbul, C. Koca, T. Capin, and U. Gdkbay, “Saliency for animated meshes with material properties,” in *Proceedings of the 7th Symposium on Applied Perception in Graphics and Visualization*, ser. APGV ’10. Los Angeles, California: ACM, New York, NY, USA, 2010, pp. 81–88. [Online]. Available: <http://doi.acm.org/10.1145/1836248.1836263>
- [12] K. Mhler, M. Neugebauer, C. Tietjen, and B. Preim, “Viewpoint selection for intervention planning,” in *Proceedings of EuroVis07: Joint Eurographics - IEEE VGTC Symposium on Visualization, Norrkping, Sweden, 23-25 May 2007*. Eurographics Association, 2007, pp. 267–274.
- [13] G. Ji and H.-W. Shen, “Dynamic view selection for time-varying volumes,” *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 5, pp. 1109–1116, 2006.
- [14] M. Klomann and J.-T. Milde, “Semi autonomous camera control in dynamic virtual environments,” in *HCI (14)*, 2011, pp. 362–369.
- [15] C. H. Lee, A. Varshney, and D. W. Jacobs, “Mesh saliency,” *ACM Trans. Graph.*, vol. 24, no. 3, pp. 659–666, 2005.
- [16] M. Christie, P. Olivier, and J.-M. Normand, “Camera control in computer graphics,” *Comput. Graph. Forum*, vol. 27, no. 8, pp. 2197–2218, 2008.
- [17] M. Friendly, “Milestones in the history of thematic cartography, statistical graphics, and data visualization,” 2009, <http://datavis.ca/milestones/>. [Online]. Available: <http://www.math.yorku.ca/SCS/Gallery/milestone/>
- [18] N. Shedroff, “Information Interaction Design: A Unified Field Theory of Design,” Vivid studios, Tech. Rep., 1994.
- [19] E. R. Tufte, *Envisioning Information*. Graphics Press, 1990.
- [20] —, *The Visual Display of Quantitative Information*, 2nd ed. Graphics Press, 2001.

- [21] B. Shneiderman and B. B. Bederson, *The Craft of Information Visualization: Readings and Reflections*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003.
- [22] C.-h. Chen, W. Hrdle, A. Unwin, C.-h. Chen, W. Hrdle, and A. Unwin, *Handbook of Data Visualization (Springer Handbooks of Computational Statistics)*, 1st ed. Santa Clara, CA, USA: Springer-Verlag TELOS, 2008.
- [23] A. Badsey-Ellis, *London's lost tube schemes*. Capital Transport, 2005. [Online]. Available: <http://books.google.com/books?id=zTUeAQAAMAAJ>
- [24] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis, "Algorithms for drawing graphs: an annotated bibliography," *Computational Geometry: Theory and Applications*, vol. 4, no. 5, pp. 235–282, 1994. [Online]. Available: <http://portal.acm.org/citation.cfm?id=195598>
- [25] D. Fisher, "Hotmap: Looking at geographic attention," *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 6, pp. 1184–1191, 2007.
- [26] C. Chen, "Top 10 unsolved information visualization problems," *IEEE Comput. Graph. Appl.*, vol. 25, pp. 12–16, July 2005. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1079828.1079847>
- [27] D. A. Keim, F. Mansmann, J. Schneidewind, J. Thomas, and H. Ziegler, "Visual analytics: Scope and challenges," in *Visual Data Mining*, ser. Lecture Notes in Computer Science, S. J. Simoff, M. H. Böhlen, and A. Mazeika, Eds. Springer, 2008, vol. 4404, pp. 76–90.
- [28] M. E. J. Newman, "The structure and function of complex networks," *SIAM REVIEW*, vol. 45, pp. 167–256, 2003.
- [29] S. Kaza, D. Hu, and H. Chen, "Dynamic social network analysis of a dark network: Identifying significant facilitators," in *ISI*. IEEE, 2007, pp. 40–46.

- [30] M. E. J. Newman, “The structure of scientific collaboration networks,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 98, no. 2, pp. 404–409, January 2001. [Online]. Available: <http://dx.doi.org/10.1073/pnas.021544898>
- [31] L. C. Freeman, “Centrality in social networks: Conceptual clarification,” *Social Networks*, vol. 1, no. 3, pp. 215–239, 1979. [Online]. Available: [http://dx.doi.org/10.1016/0378-8733\(78\)90021-7](http://dx.doi.org/10.1016/0378-8733(78)90021-7)
- [32] S. White and P. Smyth, “Algorithms for estimating relative importance in networks,” in *KDD*, L. Getoor, T. E. Senator, P. Domingos, and C. Faloutsos, Eds. ACM, 2003, pp. 266–275.
- [33] T. Dwyer, S.-H. Hong, D. Koschützki, F. Schreiber, and K. Xu, “Visual analysis of network centralities,” in *APVIS*, ser. CRPIT, K. Misue, K. Sugiyama, and J. Tanaka, Eds., vol. 60. Australian Computer Society, 2006, pp. 189–197.
- [34] M. K. Sparrow, “The application of network analysis to criminal intelligence: An assessment of the prospects,” *Social Networks*, vol. 13, no. 3, pp. 251–274, 1991. [Online]. Available: <http://www.sciencedirect.com/science/article/B6VD1-466DRKY-14/2/4785fa0af47facaa260a47d231af3f98>
- [35] T. Crnovrsanin, C. D. Correa, and K.-L. Ma, “Social network discovery based on sensitivity analysis,” in *ASONAM*, N. Memon and R. Alhaji, Eds. IEEE Computer Society, 2009, pp. 107–112.
- [36] T. Falkowski, J. Bartelheimer, and M. Spiliopoulou, “Mining and visualizing the evolution of subgroups in social networks,” in *Web Intelligence*. IEEE Computer Society, 2006, pp. 52–58.
- [37] T. Kamada and S. Kawai, “A simple method for computing general position in displaying three-dimensional objects,” *Computer Vision, Graphics, and Image Processing*, vol. 41, no. 1, pp. 43–56, 1988.

- [38] P. Barral, G. Dorme, and D. Plemenos, “Scene understanding techniques using a virtual camera,” in *Proceedings of Eurographics 2000, Computer Graphics Forum*, 2000.
- [39] P.-P. Vázquez, “Automatic view selection through depth-based view stability analysis,” *The Visual Computer*, vol. 25, no. 5-7, pp. 441–449, 2009.
- [40] D. Sokolov and D. Plemenos, “High level methods for scene exploration,” *Journal of Virtual Reality and Broadcasting*, vol. 3, no. 12, 2006.
- [41] C. Koch and S. Ullman, “Shifts in selective visual attention: towards the underlying neural circuitry,” *Human Neurobiology*, vol. 4, no. 4, 1985.
- [42] L. Itti, C. Koch, and E. Niebur, “A model of saliency-based visual attention for rapid scene analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, 1998.
- [43] S. Takahashi, I. Fujishiro, Y. Takeshima, and T. Nishita, “A feature-driven approach to locating optimal viewpoints for volume visualization,” in *Proceedings of IEEE Visualization Conference (VIS 2005), 23-28 October 2005, Minneapolis, MN, USA*. IEEE Computer Society, 2005, p. 63.
- [44] U. Bordoloi and H.-W. Shen, “View selection for volume rendering,” in *Proceedings of IEEE Visualization Conference (VIS 2005), 23-28 October 2005, Minneapolis, MN, USA*, 2005, p. 62.
- [45] Y.-S. Liu, M. Liu, D. Kihara, and K. Ramani, “Salient critical points for meshes,” in *Proceedings of ACM Solid and Physical Modeling Symposium (ACM SPM 2007), 04-06 June 2007, Beijing, China*, 2007, pp. 277–282.
- [46] C. Ware and S. Osborne, “Exploration and virtual camera control in virtual three dimensional environments,” *SIGGRAPH Comput. Graph.*, vol. 24, pp. 175–183, February 1990. [Online]. Available: <http://doi.acm.org/10.1145/91394.91442>

- [47] K. Shoemake, “Arcball: a user interface for specifying three-dimensional orientation using a mouse,” in *Proceedings of the conference on Graphics interface '92*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1992, pp. 151–156. [Online]. Available: <http://portal.acm.org/citation.cfm?id=155294.155312>
- [48] A. J. Hanson and E. A. Wernert, “Constrained 3d navigation with 2d controllers,” in *IEEE Visualization*, 1997, pp. 175–182.
- [49] R. Turner, F. Balaguer, E. Gobbetti, and D. Thalmann, “Scientific visualization of physical phenomena,” N. M. Patrikalakis, Ed. New York, NY, USA: Springer-Verlag New York, Inc., 1991, ch. Physically-based interactive camera motion control using 3D input devices, pp. 135–145. [Online]. Available: <http://portal.acm.org/citation.cfm?id=139834.139852>
- [50] D. Xiao and R. J. Hubbard, “Navigation guided by artificial force fields,” in *CHI*, 1998, pp. 179–186.
- [51] M. Gleicher and A. P. Witkin, “Through-the-lens camera control,” in *SIGGRAPH*, 1992, pp. 331–340.
- [52] N. Elmqvist, M. E. Tudoreanu, and P. Tsigas, “Tour generation for exploration of 3d virtual environments,” *VRST*, vol. 27, pp. 207–210, 2007.
- [53] C. Andjar, P. Vázquez, and M. Fairn, “Way-finder: guided tours through complex walkthrough models,” *Computer Graphics Forum*, vol. 23, no. 3, pp. 499–508, 2004. [Online]. Available: <http://dx.doi.org/10.1111/j.1467-8659.2004.00781.x>
- [54] B. Espiau, F. Chaumette, and P. Rives, “A new approach to visual servoing in robotics,” in *Geometric Reasoning for Perception and Action*, 1991, pp. 106–136.

- [55] W. H. Bares, S. McDermott, C. Boudreaux, and S. Thainimit, “Virtual 3d camera composition from frame constraints,” in *ACM Multimedia*, 2000, pp. 177–186.
- [56] J. Shetty and J. Adibi, “Discovering important nodes through graph entropy the case of enron email database,” in *Proceedings of the 3rd international workshop on Link discovery*, ser. LinkKDD '05. New York, NY, USA: ACM, 2005, pp. 74–81. [Online]. Available: <http://doi.acm.org/10.1145/1134271.1134282>
- [57] “THE GRAPHML FILE FORMAT,” ”<http://graphml.graphdrawing.org> December 2011”.
- [58] L. C. Freeman, “A Set of Measures of Centrality Based on Betweenness,” *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.
- [59] U. Brandes, “A faster algorithm for betweenness centrality,” *Journal of Mathematical Sociology*, vol. 25, pp. 163–177, 2001.
- [60] M. Safar, N. El-Sayed, K. Mahdi, and D. Taniar, “Entropy optimization of social networks using an evolutionary algorithm,” *J. UCS*, vol. 16, no. 6, pp. 983–1003, 2010.
- [61] R. Albert, H. Jeong, and A.-L. Barabasi, “Error and attack tolerance of complex networks,” *NATURE*, vol. 406, p. 378, 2000. [Online]. Available: [doi:10.1038/35019019](https://doi.org/10.1038/35019019)
- [62] I. Herman, G. Melançon, and M. S. Marshall, “Graph visualization and navigation in information visualization: A survey,” *IEEE Trans. Vis. Comput. Graph.*, vol. 6, no. 1, pp. 24–43, 2000.
- [63] N. Henry, J.-D. Fekete, and M. J. McGuffin, “Nodetrix: a hybrid visualization of social networks,” *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 6, pp. 1302–1309, 2007.

- [64] A. Noack, “Energy models for graph clustering,” *J. Graph Algorithms Appl.*, vol. 11, no. 2, pp. 453–480, 2007.
- [65] P.-P. Vázquez and M. Sbert, “Fast adaptive selection of best views,” in *Proceedings of Computational Science and Its Applications - ICCSA 2003, International Conference, Montreal, Canada, May 18-21*, ser. Lecture Notes in Computer Science. Springer, 2003, pp. 295–305.
- [66] P.-P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich, “Automatic view selection using viewpoint entropy and its applications to image-based modelling,” *Comput. Graph. Forum*, vol. 22, no. 4, pp. 689–700, 2003.
- [67] P. Castelló, M. Sbert, M. Chover, and M. Feixas, “Techniques for computing viewpoint entropy of a 3d scene,” in *Proceedings of International Conference on Computational Science (ICCS 2006), 28-31 May 2006, Reading, UK*, 2006, pp. 263–270.
- [68] G. Taubin, “Estimating the tensor of curvature of a surface from a polyhedral approximation,” in *Proceedings of International Conference on Computer Vision (ICCV 95), 20-23 June 1995, Cambridge, Massachusetts, USA*, 1995, pp. 902–907.
- [69] G. Golub and C. Van Loan, *Matrix Computations*, 2nd ed. Baltimore: Johns Hopkins University Press, 1989.
- [70] O. Polonsky, G. Patanè, S. Biasotti, C. Gotsman, and M. Spagnuolo, “What’s in an image?” *The Visual Computer*, vol. 21, no. 8-10, pp. 840–847, 2005.
- [71] D. L. Page, A. Koschan, S. R. Sukumar, B. Roui-Abidi, and M. A. Abidi, “Shape analysis algorithm based on information theory,” in *Proceedings of the International Conference on Image Processing (ICIP 2003), 14-18 September 2003, Barcelona, Catalonia, Spain*, 2003, pp. 229–232.

- [72] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, “Discrete differential-geometry operators for triangulated 2-manifolds,” in *Visualization and Mathematics III*, H.-C. Hege and K. Polthier, Eds. Heidelberg: Springer-Verlag, 2003, pp. 35–57.
- [73] “CGAL, Computational Geometry Algorithms Library,” <http://www.cgal.org>.
- [74] VTP. (2011, July) Virtual terrain project <http://www.vterrain.org>. [Online]. Available: <http://www.vterrain.org>
- [75] S. Röttger, W. Heidrich, P. Slusallek, H. peter Seidel, and G. D. (immd, “Real-time generation of continuous levels of detail for height fields,” 1998.
- [76] G. Reinelt, *The traveling salesman: computational solutions for TSP applications*. Berlin, Heidelberg: Springer-Verlag, 1994.
- [77] R. W. Sinnott, “Virtues of the haversine,” *Sky and Telescope*, vol. 68, no. 2, p. 159, 1984.
- [78] K. Shoemake, “Animating rotation with quaternion curves,” *SIGGRAPH Comput. Graph.*, vol. 19, pp. 245–254, July 1985. [Online]. Available: <http://doi.acm.org/10.1145/325165.325242>
- [79] “GOOGLE EARTH,” <http://earth.google.com> August 2011.
- [80] “GOOGLE EARTH API,” <http://code.google.com/apis/earth> August 2011.