

**The 20<sup>th</sup> International Conference in Central Europe on Computer  
Graphics, Visualization and Computer Vision**

in co-operation with

**EUROGRAPHICS**

**W S C G ' 2012**

**Poster Papers Proceedings**

Plzen

Czech Republic

June 26 - 28, 2012

*Co-Chairs*

**Enhua Wu, University of Macau & Chinese Academy of Sciences, China  
Vaclav Skala, University of West Bohemia Czech Republic**

*Edited by*  
**Vaclav Skala**

Vaclav Skala – Union Agency

***WSCG'2012 Poster Papers Proceedings***

Editor-in-Chief: Vaclav Skala  
c/o University of West Bohemia, Univerzitni 8  
CZ 306 14 Plzen  
Czech Republic  
*skala@kiv.zcu.cz*

Managing Editor: Vaclav Skala

Published and printed by:  
Vaclav Skala – Union Agency  
Na Mazinách 9  
CZ 322 00 Plzen  
Czech Republic

Hardcopy: *ISBN 978-80-86943-80-0*

# WSCG 2012

## International Program Committee

Adzhiev,V. (United Kingdom)

Benes,B. (United States)

Bengtsson,E. (Sweden)

Benoit,C. (France)

Bilbao,J. (Spain)

Biri,V. (France)

Bittner,J. (Czech Republic)

Bouatouch,K. (France)

Bourke,P. (Australia)

Coquillart,S. (France)

Daniel,M. (France)

de Geus,K. (Brazil)

Debelov,V. (Russia)

Feito,F. (Spain)

Ferguson,S. (United Kingdom)

Flaquer,J. (Spain)

Gavrilova,M. (Canada)

Gudukbay,U. (Turkey)

Havran,V. (Czech Republic)

Hege,H. (Germany)

Chmielewski,L. (Poland)

Chover,M. (Spain)

Chrysanthou,Y. (Cyprus)

Jansen,F. (Netherlands)

Klosowski,J. (United States)

Magnor,M. (Germany)

Max,N. (United States)

Molla Vaya,R. (Spain)

Muller,H. (Germany)

Murtagh,F. (Ireland)

Myszkowski,K. (Germany)

Pan,R. (China)

Pasko,A. (United Kingdom)

Pedrini,H. (Brazil)

Platis,N. (Greece)

Rojas-Sola,J. (Spain)

Rokita,P. (Poland)

Rudomin,I. (Mexico)

Sakas,G. (Germany)

Santos,L. (Portugal)

Skala,V. (Czech Republic)

Slavik,P. (Czech Republic)

Sochor,J. (Czech Republic)

Sramek,M. (Austria)

Stadt,O. (Germany)

Stroud,I. (Switzerland)

Teschner,M. (Germany)

Tokuta,A. (United States)

Triantafyllidis,G. (Greece)

Vergeest,J. (Netherlands)

Vitulano,D. (Italy)

Weiss,G. (Germany)

Wu,E. (China)

Wuethrich,C. (Germany)

Zara,J. (Czech Republic)

Zemcik,P. (Czech Republic)

Zitova,B. (Czech Republic)





# WSCG 2012

## Board of Reviewers

Abad,F. (Spain)	Durikovic,R. (Slovakia)	Chrysanthou,Y. (Cyprus)
Adzhiev,V. (United Kingdom)	Eisemann,M. (Germany)	Ihrke,I. (Germany)
Ariu,D. (Italy)	Erbacher,R. (United States)	Jansen,F. (Netherlands)
Assarsson,U. (Sweden)	Erleben,K. (Denmark)	Jeschke,S. (Austria)
Aveneau,L. (France)	Essert,C. (France)	Jones,M. (United Kingdom)
Barthe,L. (France)	Faudot,D. (France)	Juettler,B. (Austria)
Battiato,S. (Italy)	Feito,F. (Spain)	Kanai,T. (Japan)
Benes,B. (United States)	Ferguson,S. (United Kingdom)	Kim,H. (Korea)
Benger,W. (United States)	Fernandes,A. (Portugal)	Klosowski,J. (United States)
Bengtsson,E. (Sweden)	Flaquer,J. (Spain)	Kohout,J. (Czech Republic)
Benoit,C. (France)	Flerackers,E. (Belgium)	Krivanek,J. (Czech Republic)
Beyer,J. (Saudi Arabia)	Fuenfzig,C. (Germany)	Kurillo,G. (United States)
Biasotti,S. (Italy)	Galo,M. (Brazil)	Kurt,M. (Turkey)
Bilbao,J. (Spain)	Garcia Hernandez,R. (Spain)	Lay Herrera,T. (Germany)
Biri,V. (France)	Garcia-Alonso,A. (Spain)	Lien,J. (United States)
Bittner,J. (Czech Republic)	Gavrilova,M. (Canada)	Liu,S. (China)
Bosch,C. (Spain)	Giannini,F. (Italy)	Liu,D. (Taiwan)
Bouatouch,K. (France)	Gobron,S. (Switzerland)	Loscos,C. (France)
Bourdin,J. (France)	Gonzalez,P. (Spain)	Lucas,L. (France)
Bourke,P. (Australia)	Gudukbay,U. (Turkey)	Lutteroth,C. (New Zealand)
Bruckner,S. (Austria)	Guérin,E. (France)	Maciel,A. (Brazil)
Bruder,G. (Germany)	Hall,P. (United Kingdom)	Madeiras Pereira,J. (Portugal)
Bruni,V. (Italy)	Hansford,D. (United States)	Magnor,M. (Germany)
Buriol,T. (Brazil)	Haro,A. (United States)	Manak,M. (Czech Republic)
Cakmak,H. (Germany)	Hasler,N. (Germany)	Manzke,M. (Ireland)
Capek,M. (Czech Republic)	Hast,A. (Sweden)	Mas,A. (Spain)
Cline,D. (United States)	Havran,V. (Czech Republic)	Masia,B. (Spain)
Coquillart,S. (France)	Hege,H. (Germany)	Masood,S. (United States)
Corcoran,A. (Ireland)	Hernandez,B. (Mexico)	Matey,L. (Spain)
Cosker,D. (United Kingdom)	Herout,A. (Czech Republic)	Matkovic,K. (Austria)
Daniel,M. (France)	Hicks,Y. (United Kingdom)	Max,N. (United States)
Daniels,K. (United States)	Horain,P. (France)	McDonnell,R. (Ireland)
de Geus,K. (Brazil)	House,D. (United States)	McKisic,K. (United States)
De Paolis,L. (Italy)	Chaine,R. (France)	Mestre,D. (France)
Debelov,V. (Russia)	Chaudhuri,D. (India)	Molina Masso,J. (Spain)
Dingliana,J. (Ireland)	Chmielewski,L. (Poland)	Molla Vaya,R. (Spain)
Dokken,T. (Norway)	Choi,S. (Korea)	Montrucchio,B. (Italy)
Drechsler,K. (Germany)	Chover,M. (Spain)	Muller,H. (Germany)

Murtagh,F. (Ireland)	Sadlo,F. (Germany)	Tian,F. (United Kingdom)
Myszkowski,K. (Germany)	Sakas,G. (Germany)	Tokuta,A. (United States)
Niemann,H. (Germany)	Salveti,O. (Italy)	Torrens,F. (Spain)
Okabe,M. (Japan)	Sanna,A. (Italy)	Triantafyllidis,G. (Greece)
Oliveira Junior,P. (Brazil)	Santos,L. (Portugal)	TYTKOWSKI,K. (Poland)
Oyarzun Laura,C. (Germany)	Sapidis,N. (Greece)	Umlauf,G. (Germany)
Pala,P. (Italy)	Savchenko,V. (Japan)	Vavilin,A. (Korea)
Pan,R. (China)	Sellent,A. (Germany)	Vazquez,P. (Spain)
Papaioannou,G. (Greece)	Sheng,B. (China)	Vergeest,J. (Netherlands)
Paquette,E. (Canada)	Sherstyuk,A. (United States)	Vitulano,D. (Italy)
Pasko,A. (United Kingdom)	Shesh,A. (United States)	Vosinakis,S. (Greece)
Pasko,G. (United Kingdom)	Schultz,T. (Germany)	Walczak,K. (Poland)
Pastor,L. (Spain)	Sirakov,N. (United States)	WAN,L. (China)
Patane,G. (Italy)	Skala,V. (Czech Republic)	Wang,C. (Hong Kong SAR)
Patow,G. (Spain)	Slavik,P. (Czech Republic)	Weber,A. (Germany)
Pedrini,H. (Brazil)	Sochor,J. (Czech Republic)	Weiss,G. (Germany)
Peters,J. (United States)	Solis,A. (Mexico)	Wu,E. (China)
Peytavie,A. (France)	Sourin,A. (Singapore)	Wuensche,B. (New Zealand)
Pina,J. (Spain)	Sousa,A. (Portugal)	Wuethrich,C. (Germany)
Platis,N. (Greece)	Sramek,M. (Austria)	Xin,S. (Singapore)
Plemenos,D. (France)	Stadt,O. ( )	Xu,D. (United States)
Poulin,P. (Canada)	Stroud,I. (Switzerland)	Yang,X. (China)
Puig,A. (Spain)	Subsol,G. (France)	Yoshizawa,S. (Japan)
Reisner-Kollmann,I. (Austria)	Sunar,M. (Malaysia)	YU,Q. (United Kingdom)
Renaud,c. (France)	Sundstedt,V. (Sweden)	Yue,Y. (Japan)
Reshetov,A. (United States)	Svoboda,T. (Czech Republic)	Zara,J. (Czech Republic)
Richardson,J. (United States)	Szecs,L. (Hungary)	Zemcik,P. (Czech Republic)
Rojas-Sola,J. (Spain)	Takala,T. (Finland)	Zhang,X. (Korea)
Rokita,P. (Poland)	Tang,M. (China)	Zhang,X. (China)
Rudomin,I. (Mexico)	Tavares,J. (Portugal)	Zillich,M. (Austria)
Runde,C. (Germany)	Teschner,M. (Germany)	Zitova,B. (Czech Republic)
Sacco,M. (Italy)	Theussl,T. (Saudi Arabia)	Zwettler,G. (Austria)

# **WSCG 2012**

## **Poster papers Proceedings**

### **Contents**

Vergeest,J.S.M.: Self-speed and Headway Measurement in Highway Traffic from Onboard Video Footage	1
Kheyfets,A.L.: 3D as a Method of Geometrical Simulation (on the example of combination of a quadric and conic)	5
Yang,K.-C., Hung,C.-J, Yu,C.-H., Wang,J.-S.: Robust Affinity Propagation using Preference Estimation	11
Kovalčík,V., Chmelík,J., Bezdíka,M., Sochor,J.: Virtual Reality System as a Tool for Education	15
Simac-Lejeune,A.: Use the Palmprint for Identification and Authentication of Persons: A New Real-time Method of Data Treatment Based on Graph Construction of Spatial Interest Points	19
Zafar Asoodeh,A., Rostamabadi,F., Ahmadi,A.: Map Point-Labeling with Rotation in Slider Model Using an Efficient Evolutionary Algorithm	23
Bay,T., Raffin,R., Daniel,M: Geometric modeling of pelvic organs with a discrete offset approach	27
Fares,C.: Convex Envelope Generation Using a Mix of Gift Wrap and QuickHull Algorithms	31
Kolomazník,J., Horáček,J., Krajíček,V., Pelikán,J.: Implementing Interactive 3D Segmentation on CUDA Using Graph-Cuts and Watershed Transformation	35
Seeman,M., Zemčík,P.: Visual Acuity and Comfortable Distance from a Display	39
Li,H., Wu,Z., Huang,D.: Geodesic Stripes Based Hierarchical Evaluation for 3D Facial Similarity	43



# Self-speed and Headway Measurement in Highway Traffic from Onboard Video Footage

Joris Vergeest  
Delft University of Technology  
Landbergstraat 15  
NL-2628 CE Delft, The  
Netherlands  
j.s.m.vergeest@tudelft.nl

## ABSTRACT

We present a method to automatically collect data about distance and speed of vehicles ahead of a car participating in highway traffic. The method should support the registration of traffic in front of a driving car over long periods of time in order to be able to detect over-persistent congestion (OPC) situations. The method requires no specialized devices; it is based on a simple video stream from any (low-cost) camera mounted near the front window of a car. The data is used to fine-tune traffic flow models in order to determine the conditions for the emergence and the dissipation of traffic jams more accurately. The paper deals with the performance of the vehicle tracking method itself and presents a criterion for over-persistent traffic jams applied to the data. The outlook towards a driver's advisory system to reduce traffic congestion (and thus time loss and environmental load) is discussed as well.

## Keywords

Collecting traffic statistics, car detection, speed measurement, reduction of traffic congestion.

## 1. INTRODUCTION

In densely populated regions (such as in the western part of the Netherlands), infrastructural expansion of motorized traffic is almost no longer an option. Instead, several types of regulations are being considered to flatten rush-hour peaks and to create financial penalties on car drivers using certain roads at certain times. To optimize the traffic flow, information provision systems through road panels or radio, telephone and internet channels can advise the driver and/or update the car navigation device. Research is rapidly progressing toward intelligent vehicles, based on onboard observation equipment and automatic information exchange among nearby cars [Kim 2007, Guenther 2008, Doman 2010, Sun 2006]. We are still far away from automatic driving, and much of the efficiency of the road capacity usage depends heavily on the individual car drivers and their conduct in various traffic situations. Individual

driving styles can be modeled and evaluated using traffic flow models [Fellendorf 2000, Hoogendoorn 2010]. It has been reported that traffic congestions are reduced or avoided when car drivers receive particular guidance or when adaptive cruise control is implemented [Driel 2010]. To obtain better understanding of car driving behavior and to reveal critical situations we propose to collect and analyze speed and distance data taken from a car participating in high-way traffic over long periods of time. The acquired information will be different from and complementary to what is measured customary at counting points stationary to the road, and thus have the potential to validate and enrich traffic flow models.

## 2. OVER-PERSISTENT CONGESTIONS

One factor that affects the life time of traffic congestions significantly is the outflow of vehicles at the downstream front of jams. In [Kesting 2008] it has been demonstrated in simulation runs that an increase of acceleration combined with a decrease of headway time is positively correlated to the maximum capacity for freeways.

To model the car driver's behavior we have made use of the intelligent-driver model or IDM [Treiber 2006]. The IDM is a microscopic traffic flow model

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

that can accommodate simple human driving schemes as well as ACC algorithms and other driving assistance principles by setting a number of parameters.

Using the IDM, we have simulated a number of single-lane traffic situations, representing jam forming and dissipation and studied the evolution of traffic for differing settings of (*e.g.*)  $T$  and  $a_{\max}$ . In the scenario we have initialized two clusters of cars. The first consists of 200 cars driving in a platoon fashion at low speed,  $v_i = 10\text{Km/h}$  at  $t = 0$ . The length of the cluster is approximately 3.1Km. In the simulation, the foremost car is assigned a constant acceleration of  $1.3\text{m/s}^2$  beginning at  $t > 300\text{s}$ , until car 1 reaches  $v = 120\text{Km/h}$ , which takes about 24s. The remaining 199 cars are followers according to the IDM.

The distance between the clusters is shrinking until  $t \approx 750\text{s}$ , when car 201 gets close to the tail of cluster 1 and needs to reduce its speed from 120 to approximately 65Km/h. At that point in time car 200 has  $v = 50\text{Km/h}$  and appears to quickly accelerate towards  $v^*$  as did all cars in cluster 1 before. For  $t > 770\text{s}$  all 400 cars have merged into a single cluster and have reached an equilibrium state at a speed near  $v^*$ . The duration of merging process is about 40s.

An over-persistent congestion (OPC) has been produced in the simulation by modifying the parameters of the IDM. Two effects are then obvious. Car 200 has reached a speed of 105Km/h at  $t = 861\text{s}$ , which is 95s later than it did in condition B. This delay triggers another effect on cluster 2. Car 201, leading cluster 2, reaches car 200 while the latter is still at its low speed of 10Km/h. After about 1 minute car 200 starts to accelerate, dragging cluster 2 behind it. A strip of about 60 low-speed cars propagates to the tail of cluster 2 in about 300s. It can be concluded that (unnecessary) slow and small acceleration contributes to the emergence and lifetime of traffic jams and causes delay for traffic behind. Similar observations were made by [Treiber 2003].

To be able to predict (and then to conquer) OPCs reliably, it is necessary to improve the car following model. To do so we need empirical statistics about the acceleration function  $a(s, v, \Delta v)$ . Collecting data  $a, v$  and  $s$  as a function of  $t$  over long durations would serve that purpose.

### 3. COLLECTING LONG-DURATION DATA FROM VIDEO FOOTAGE

The most common way to register traffic flow is by using stationary measurement points such as induction loops. However, the resulting data would

not yield sufficient information about the behavior of individual car drivers. Video data from a helicopter as obtained in [Hoogendoorn 2006] could provide some clues. High-statistics information from onboard cameras are very suited to correctly model the behavior of car drivers when they leave traffic jams, either by adapting the parameters of equation (1) or by extending equation (1) with empirically-based terms.

Instead of video data one could use alternative ways to collect position and distance data. Position (and thus speed) information could be obtained from GPS data and distance to the car ahead could be obtained from radar sensing. However, we are aiming at a method to obtain the information with minimal intervention using a simple device which can easily be mounted and demounted from a car. The idea is that information should be gathered from a number of cars over longer durations, *e.g.* 1 to 4 weeks, where the operation of the device should be either absent or minimal. Also, there should not be the necessity of a technical installation involving the connection to existing devices in the car (such as its navigation system or ACC). The device should be simply mountable into any car and require not more from the driver than its switching on or off. A video camera with built-in hard drive is one option. Another option would be a webcam (perhaps attached to the front window) connected to a data storage system, provided that the devices do not cause any disturbance to the driver.

The method that we developed starts from pixel data coming from any simple webcam or video camera fixed inside the car near the lower edge of the wind shield, where the camera is aimed nearly horizontally into the forward driving direction. The video data is analyzed frame by frame. As a first step we search, in the frame, for brightness drops on the road in the positive  $z$ -direction, which is the driving direction of our car, corresponding to the positive  $y$ -direction in the pixel plane. If in addition the brightness of the darker pixel is below a certain limit we have possibly detected the lower part or shadow from a car ahead. The threshold for brightness drop was defined as a fraction of the average brightness of the road surface, which was calculated for each frame individually. We did not make use from color information. Treating dark regions as hints of vehicles was recently applied successfully in the framework presented by [Nieto 2011]. The method works in daylight conditions but also during night conditions with street light illumination. Depending on the threshold settings of the brightness drop, the darkness criterion etc, candidate regions are detected

on and outside of the road.

#### 4. IMPLEMENTATION DETAILS

The input to the software are video recordings made by a low-cost camera with a resolution of 720 times 576 at 25 frames per second. Video recordings with practically unlimited duration without interruption can be obtained. The only non-standard provision is electric power supply from the car's 12V battery. We used a simple 12V DC to 230V AC converter to power the camera over long time periods as to be independent from the camera's battery pack.



**Fig. 1. Result of single-frame analysis with different settings of darkness criterion and brightness drop-down. Candidate cars have been marked red; white lane markers marked blue; the assumed horizon is represented by the green line (bottom).**

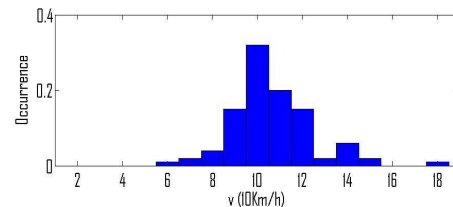
The speed and distance extraction method has been implemented in C++ on the Visual Studio platform [Microsoft 2012] using the OpenCV library of real-time computer vision algorithms [OpenCV 2012]. OpenCV supports the import of video films and accessing and editing of individual video frames, as well as outputting of (annotated and/or modified) video files.

Criteria have been introduced to define whether a dark region is positioned on or beyond the road surface and by requiring a minimal size of the region in horizontal direction. When bright lane markings are visible the algorithm can be focused on the area bounded by them. However, the most important criterion for a dark region to be associated with a driving car ahead is temporal coherence.

#### 5. RESULTS

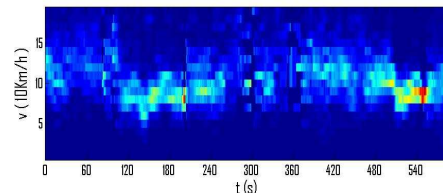
We have collected video footage from a dozen of journeys, ranging in duration from 1 to 13 hours. The algorithm extracts the right speed and distance under various daylight conditions. However, strong against-light or a snowy road surface prohibit the detections.

As explained above,  $v$  is computed from the maximally populated bin occurring in the past 250 consecutive velocity histograms, representing 10 seconds. The fraction of speed values that actually occurred in such bins is plotted as well and floats around 20%. An example of the self-speed histogram at a particular point in time is shown in Fig. 4.



**Fig. 2. Distribution of measurements of self-velocity at one point in time.**

When this fraction gets low (*e.g.* smaller than 5%) then the speed measurement is no longer reliable. This occurs *e.g.* at  $t = 270s$ . The speed  $v$  is not strongly influenced by short interruptions of information from white markings, as the averaging over 250 frames acts as a low-pass filter for  $v$ . The evolution of the velocity probability histogram over time is shown in Figure 3.



**Fig. 3. Velocity probability as a function of time. Bright and red color signify high probability, blue/dark signifies no or few occurring velocities .**

Based on our desktop implementation in C++ we could achieve a throughput of the data with more 25fps, that is the program would run in real time. Since much of the computation can be handled in separate independent pieces, GPU-enhanced processing could certainly be considered.

## 6. CONCLUSIONS

An initial evaluation revealed that the method works satisfactory under good and reasonable light conditions. Short interruptions of data can be compensated by low-pass filtering, but against-light, snow or darkness prevent the algorithm from working correctly. Several ways of improvement are suggested in the literature. One we are working on is to detect optical flow from any road surface. Two or three frames taken some 5ms apart in time at reasonable resolution would be sufficient to compute self-speed. It is not necessary to take frames every 5ms, but to have three frames once very second. A common high-speed camera would probably be overkill and counter effective when it would be difficult to mount and to operate.

The speed and distance values as a function of time are suited to study the driving behavior in various conditions. The data provides information about the occurrence of situations potentially compatible with OCPs.

## 7. REFERENCES

- [Doman 2010] K. Doman, D. Deguchi, T. Takahashi, Y. Mekada, I. Ide, H. Murase, Y. Tamatsu (2010), "Estimation of traffic sign visibility toward smart driver assistance". Proc. 2010 IEEE Intelligent Vehicles Symposium, pp 45-50.
- [Driel 2010] C.J.G. van Driel and B. van Arem (2010), "The impact of a congestion assistant on traffic flow efficiency and safety in congested traffic caused by a lane drop". Journal of Intelligent Transportation Systems, 14(4), pp 197-208.
- [Fellendorf 2000] M. Fellendorf, P. Vortisch (2000), "Integrated modeling of transport demand, route choice, traffic flow and traffic emissions". Proc. 79th Annual Meeting of the Transportation Research Board.
- [Guenther 2008] W. Guenther, L. Ginzinger, H. Ulbrich (2008), "Gaze control for driver assistance systems". Journal of Vibration and Control, 16, 4, pp 527-538.
- [Hoogendoorn 1998] S.P. Hoogendoorn and P.H.L. Bovy, A new estimation technique for vehicle-type specific headway distributions. Transportation Research Record 1646, 18-28, 1998.
- [Hoogendoorn 2006] S.P. Hoogendoorn, S. Ossen, K. Schreuder, Empirics of multi-anticipative car-following behavior. Transportation Research Record: Journal of the Transportation Research Board, Vol 1965, 2006.
- [Hoogendoorn 2010] S. Hoogendoorn, R. Hoogendoorn (2010), "Calibration of microscopic traffic-flow models using multiple data sources". Phil. Trans. R. Soc. A (2010) 368, 4497-4517.
- [Kesting 2010] A. Kesting, M. Treiber, D. Helbing, Enhanced Intelligent Driver Model to Assess the Impact of Driving Strategies on Traffic Capacity. Phil. Trans. R. Soc. A 13 October 2010 vol. 368 no. 1928.
- [Kim 2007] S-Y Kim, J-K Kang, S-Y Oh, Y-W Ryu, K. Kim, S-C and Machine Intelligence. 28, 5, pp 694-711.
- [Microsoft 2012] [www.microsoft.com](http://www.microsoft.com).
- [Nieto 2010] M. Nieto, J. Arróspide Laborda, L. Salgado (2010), "Road environment modeling using robust perspective analysis and recursive Bayesian segmentation". Machine Vision and Applications, 7 August 2010.
- [OpenCV 2012] [www.opencv.willowgarage](http://www.opencv.willowgarage).
- [Sun 2006] Z. Sun, G. Bebis (2006), "On-road vehicle detection: A review". IEEE Transactions on Pattern Analysis and Machine Intelligence. 28, 5, pp 694-711.
- [Treiber 2003] M. Treiber and D. Helbing, Memory effects in microscopic traffic models and wide scattering in flow-density data. Physical Review E 68, 046119 (2003).
- [Treiber 2006] M. Treiber, A. Kesting, D. Helbing, Delays, inaccuracies and anticipation in microscopic traffic models. Physica A: Statistical Mechanics and its Applications Volume 360, Issue 1, Pages 71-88, 2006.
- [Vergeest 2011] J.S.M. Vergeest, A. Kooijman, Y. Song, Collecting long-duration car driving data using an onboard camera. Proceedings of Int. Conf on Innovative Technologies, In-Tech 2011, Kudlacek, pp 566-569, 2011.



# 3D as a method of geometrical simulation (on the example of combination of a quadric and conic)

Alexander L. Kheyfets

South Ural State University, Chelyabinsk, Russia

heifets@yandex.ru

## ABSTRACT

It is shown that on the example of the mentioned problem 3d-computer geometrical models allow in full to solve and investigate applied problems of geometrical simulation.

## Keywords

Geometrical simulation, 3d-computer technologies, AutoCAD.

## 1. INTRODUCTION

A peculiar feature of most applied problems of geometric simulation is that we can't obtain an explicit and theoretically exact projection or analytical solution to these problems. However, if the solution is found, as a rule it is connected with a number of assumptions, it is complicated and cumbersome, has a low clearness and therefore, doesn't allow us to carry out the necessary analysis of the model without further graphic explanation.

The existence of graphical editors, which allow us to create virtual realistic 3D geometric models, highly expands the possibilities of geometric simulation. On this basis a new method of 3D geometric simulation has appeared, combining with computer and software it allows us to study the model and obtain visual target characteristics without analytical and structural (projection) design.

The object of this article is to show the application of 3D method and its possibilities on the example of combination of a quadric and conic. The choice of this problem as an example is due to the availability of its solutions in particular cases [1-3].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

The author has complicated the problem by considering the combination of all types of conic and selectable quadric [4, 5]. A variant of this problem is given in this article: *let us combine the given hyperbola with the given one-sheet elliptic hyperboloid (OH) if the hyperbola must pass through a selectable and given point on the surface of the hyperboloid.*

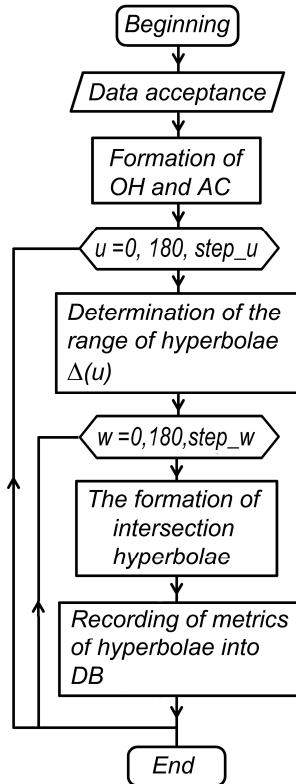
Defining the point as a parameter of the problem hasn't allowed us to obtain accurate geometric or analytical solution. Being a theoretical one, a considered problem demonstrates the possibility of new 3D methods to study and solve similar complicated applied problems.

## 2. METHOD OF STUDY

"Black box" method which is peculiar to experimental research is used to study the objects, internal structure of which is unknown. Algorithm of the solution (fig. 1) is to get a range of hyperbolae, located on the surface of  $OH$  and passing through the given point and to find hyperbola with the required metrics among them.

We shall build the model of  $OH$  (fig. 2) on the framework of two hyperbolae in the planes of symmetry and 10...20 cross-cut ellipsis. We will get hyperbolae to the framework as sections of elliptical cone [4]. We shall set out check point  $B$  on the surface of  $OH$ .

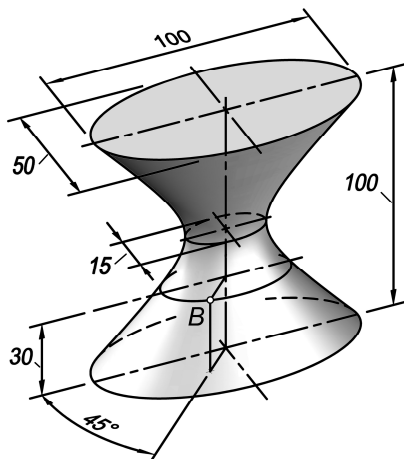
To create a range of hyperbolae we shall introduce a cutting plane  $\psi$  (fig. 3), performing a rotation around two axes passing through the given point  $B$ . The first rotation is around axis  $iI$ , which is parallel to axis  $i$  of



**Fig. 1. Block-diagram of an algorithm**

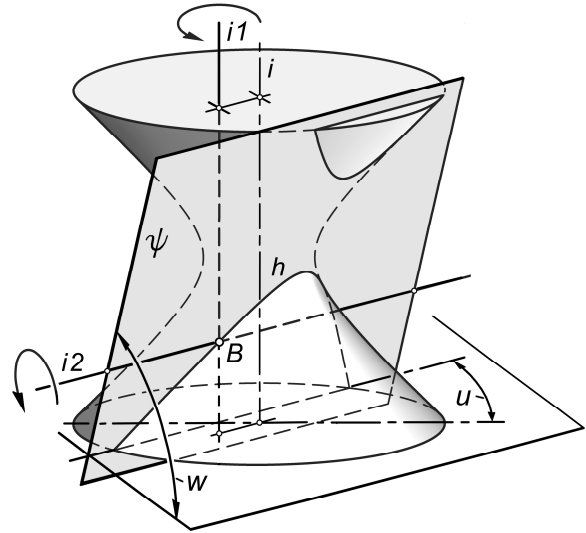
*OH*. The second rotation is around contour curve *i2*, which belongs to the plane  $\psi$  and rotating round *i1* with it. The position of plane  $\psi$  is set with the angle *u* and angle *w*.

Taking into consideration the symmetry of the model, we shall set the interval of each rotation at  $0 \dots 180^\circ$ .



**Fig. 2. Parameters of a model**

The rotary steps which are *step\_u* (at position *u*) and *step\_w* (at position *w*) are taken as  $0.5 \dots 1^\circ$  considering the required accuracy of the solution. In this case the range of conics is  $20 \dots 50$  thousand. This



**Fig. 3. A diagram of formation of a range of hyperbolae**

range is successfully proceeded on a personal computer.

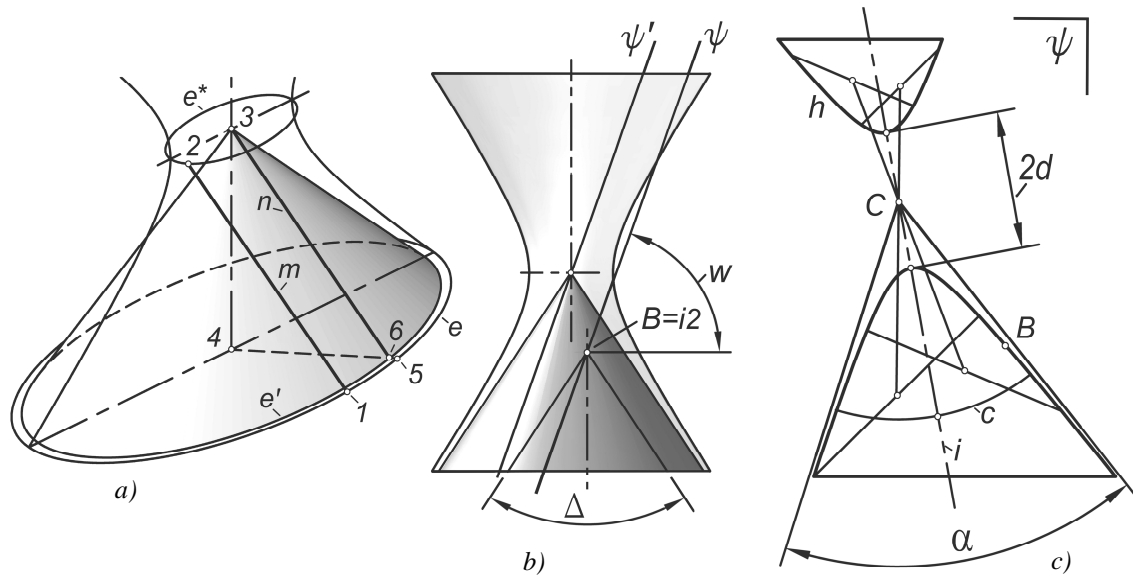
The range of hyperbolae shall be defined at asymptotic cone (AC). To build (fig. 4, *a*) it we shall draw tangent *m* to ellipsis *e* of the neck of *OH* from arbitrary point *l* of ellipsis *e* of *OH*. Tangent point 2 is determined by the object snap. Let us place *m* to the center 3 of ellipsis *e*. From the center 4 through point 6 we shall draw the segment till its intersection with ellipsis *e* at point 5. Let us scale ellipsis *e* so that it passes through point 6, thus we obtain ellipsis *e'*. We shall build AC at its base *e'* and apex 3.

For each value of *u*-coordinate we shall project AC to the plane (fig. 4, *b*) which is perpendicular to the current position of axis *i2*, and find sector  $\Delta$ . For hyperbolae cutting plane  $\psi$  shall pass within sector  $\Delta$ .

Hyperbola of the section *h* of plane  $\psi$  obtained as spline curve is characterised by metrics *d*,  $\alpha$  (fig. 4, *c*). We shall define center *C* by the method of chords. Let us draw axis *i* through the middle point *c*, then we shall define apex of hyperbola, defining metric *d*. Angle  $\alpha$  between asymptotes of hyperbola is determined by section AC with plane  $\Psi' \parallel \Psi$ .

Metrics of every hyperbolae  $\alpha$ , *d*, as well as their angle coordinates *u*, *w*, angles of inclination of the planes of hyperbolae to the planes of symmetry of *OH*, and other parameters which are necessary for the study of the model, are placed into database (DB).

Further study of the model is in the extraction of hyperbolae with the given values of metrics  $\alpha$ , *d* from DB. We shall introduce the allowance of the search (error)  $del = 0.5 (del1 + del2)$ , where *del1* is the error to metric *d*, *del2* is the error to metric  $\alpha$ . Originally having determined  $del = 5 \dots 10\%$ , we shall extract a



**Fig. 4. Elements of solution algorithm: a – asymptotic cone; b - determination of the range of hyperbolae; c – determination of metrics of hyperbolae**

group of hyperbolae from *DB*. The number of groups is the number of solutions. In each group we shall find the hyperbola with a minimum value *del*, which is one of the variant of solution.

AutoCAD is used to perform simulation, algorithms and their metrics, analysis of database is performed by means of list-processing of AutoLisp.

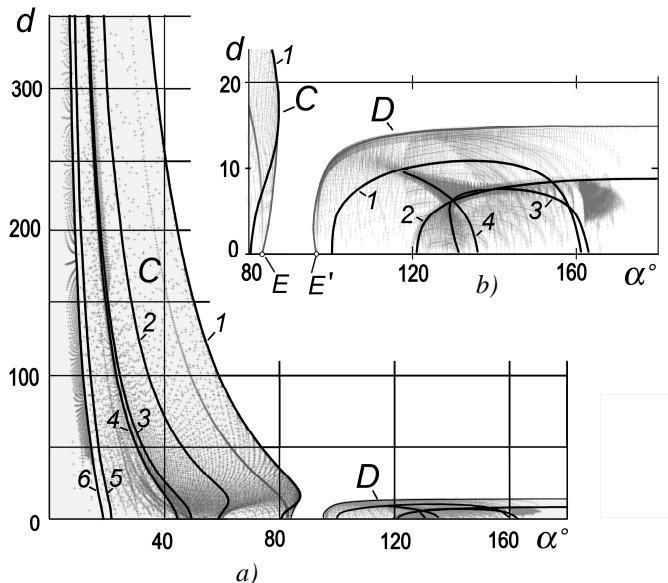
### 3. THE RANGE OF POSSIBLE SOLUTIONS

Let us display a database in the coordinates  $\alpha$ ,  $d$  (fig. 5, *a*). Each hyperbola is marked with the point. About  $\approx 36000$  points are displayed all in all. We have two areas *C* and *D*. Tests show that area *C* are the hyperbolae with the exterior center (see fig. 8, *b*). Area *D* (fig. 5, *b*) is the hyperbolae with the inner center (see fig. 9, *b*).

### 4. HYPERBOLAE OF A PARTICULAR CASE

Defining the angles of inclination of the plane of hyperbola to the planes of symmetry of *OH*, we shall get the curves 1...6. The curves 1, 5 are front-projecting hyperbolae (their planes are perpendicular to the frontal plane of symmetry of *OH* including minor axis of ellipsis); hyperbolae 1 and 5 differ from each other by the sign of the angle of inclination to the frontal plane of symmetry of *OH*. Hyperbola 2 and 4 are equally inclined to the planes of symmetry of *OH* in a number of ways. The curves 3 and 6 are front-projecting hyperbolae with different angles of inclination to the frontal plane of symmetry.

Provided  $\alpha \rightarrow 0$  asymptotes of hyperbolae are combined, distance  $2d$  between the apexes increases



**Fig. 5. The range of possible solutions**

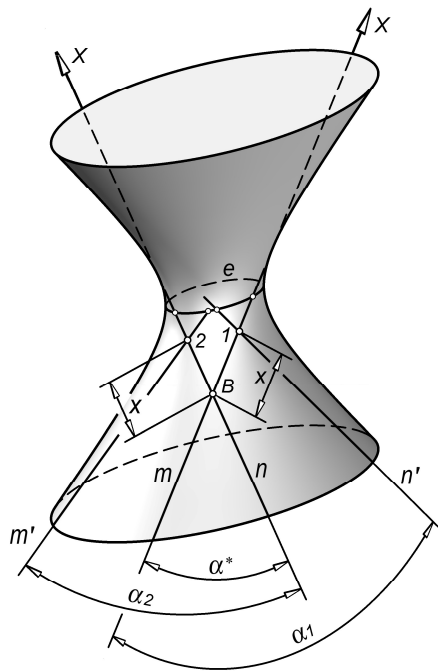
without restrictions, hyperbola degenerates into the straight line. Provided  $\alpha = 180^\circ$ , hyperbola is degenerated into two parallel lines tangent to the neck. One of the lines of each pair passes through point *B*. Provided  $d = 0$  (points of axis  $\alpha$ ), the apexes of hyperbolae coincide with the center, hyperbolae degenerate into their asymptotes, that is into two intersection lines.

## 5. THE NUMBER OF POSSIBLE SOLUTIONS

We shall build lines  $m$  and  $n$  passing through point  $B$ , which are tangent to the ellipsis  $e$  of the neck of  $OH$  (fig. 6). We shall take point  $1$  to the tangent  $m$ , and build tangent  $n'$  to the ellipsis  $e$  from this point. We will obtain lines  $m \cap n'$  which can be considered as a degenerated hyperbola with angle  $\alpha_1$  between asymptotes and  $d = 0$ . The move of point  $1$  leads to the change of angle  $\alpha_1$  according to the experimentally defined curve 1 (fig. 7). Curve 2 reflects the same dependence while moving point 2 along tangent  $n$ .

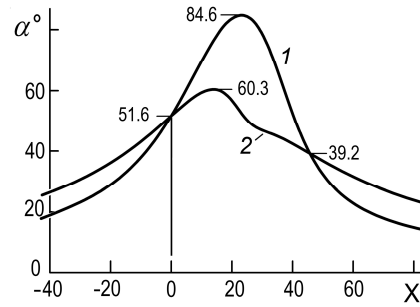
Let us assume that for a group of hyperbolae with the same value of the angle between asymptotes the number of possible solutions is equal to the number of degenerated hyperbolae of this group and doesn't depend on the distance  $d$  between the apex and center. This assumption is proved by the experimental testing (see below).

If we put the horizontal straight line on the level  $\alpha$  (see fig. 7), we shall define the number of solutions of this group with the given value of the angle  $\alpha$ . If  $\alpha > 84.6^\circ$ , there is no solution. Value  $\alpha = 84.6^\circ$  refers to the point  $E$  (see fig. 5,  $b$ ), in which there is the only



**Fig. 6. The tangent planes to the surface of  $OH$  passing through point  $B$**

solution. If  $60.3^\circ < \alpha < 84.6^\circ$  there are two solutions. If  $\alpha < 60.3^\circ$  there are four solutions. And among them there are coinciding solutions. If  $\alpha = 51.6^\circ$  and  $60.3^\circ$  two planes coincide among them and that is why among four solutions we see only three solutions. If  $\alpha$



**Fig. 7. Dependence of the angle between the tangents on the inclination of the point of tangency**

$= 39.2^\circ$  there are four solutions, two of which have equal angles between the asymptotes of hyperbolae (but their planes do not coincide).

For the circular  $OH$  the curves 1 and 2 coincide, that is the reason of existence of maximum two solutions.

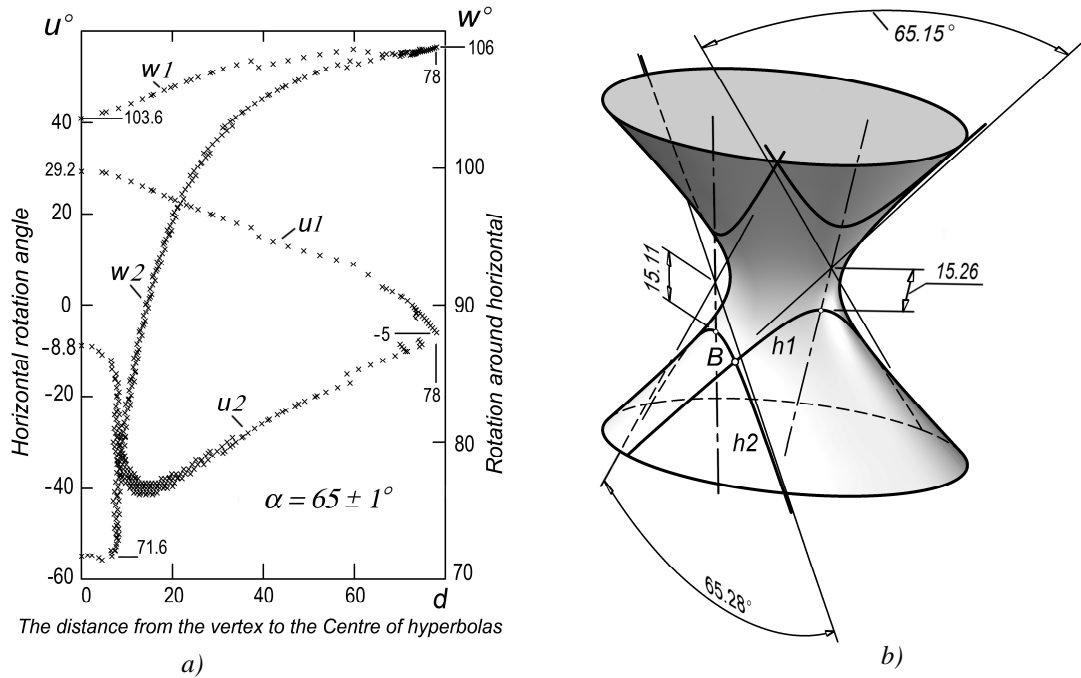
## 6. GROUPS OF HYPERBOLAE

Let us consider the formation of hyperbolae with the given angle  $\alpha$  between the asymptotes. Let  $\alpha = 65^\circ$ . Let us draw the sample of hyperbolae with  $\alpha = 65 \pm 1^\circ$  from database. We shall extract angular coordinates  $u, w$  of the planes of hyperbola and build dependency of their changes on metric  $d$  (fig. 8,  $a$ ).

We see that if  $d = 0$  there are two degenerate hyperbolae. This corresponds to fig. 7, according to which if  $\alpha = 65^\circ$  there are two solutions. To the first degenerate hyperbola there are coordinates  $u_1 = 29.2^\circ, w_1 = 103.6^\circ$  (see fig. 8,  $a$ ), to the second hyperbola the coordinates are  $u_2 = -8.8^\circ, w_2 = 71.6^\circ$ . Having planes with the given coordinates  $u, v$ , in sections we will get two pairs of concurrent straight lines, the angle between the straight lines of which is  $65^\circ$ . The increase of metric  $d$  requires the rotation of sectional plane  $\psi$  toward axes  $u$  and  $v$  in accordance with the dependence  $u_1(d), w_1(d)$  to the first hyperbola. To the second hyperbola the dependence is  $u_2(d), w_2(d)$ . In the interval  $0 < d < d_{max} \approx 78$  the original number of hyperbolae is kept. The value of  $d_{max}$  corresponds to the upper value of the area  $C$  (see fig. 5,  $a$ ). At  $d = d_{max}$  hyperbolae coincide and their coordinates take the values  $u \approx -5^\circ, w \approx 106^\circ$ .

Metrics  $\alpha = 65^\circ$  and  $d = 15$  are set as an example. Two hyperbolae  $h_1, h_2$  are found in database (fig. 8,  $b$ ). The values of  $u, v$  coordinates of these hyperbolae correspond to fig. 8,  $a$ . For  $h_1$  the error of  $del$  is  $(0.26:15 + 0.15:65) \cdot 50 = 0.98\%$ . For  $h_2$  the value of  $del = 0.58\%$ .

Obtuse angle between the asymptotes leads to the hyperbolae with inner center (see fig. 5, area  $D$ ). As the tangent planes for the angles  $\alpha$  and  $(180 - \alpha)$  are



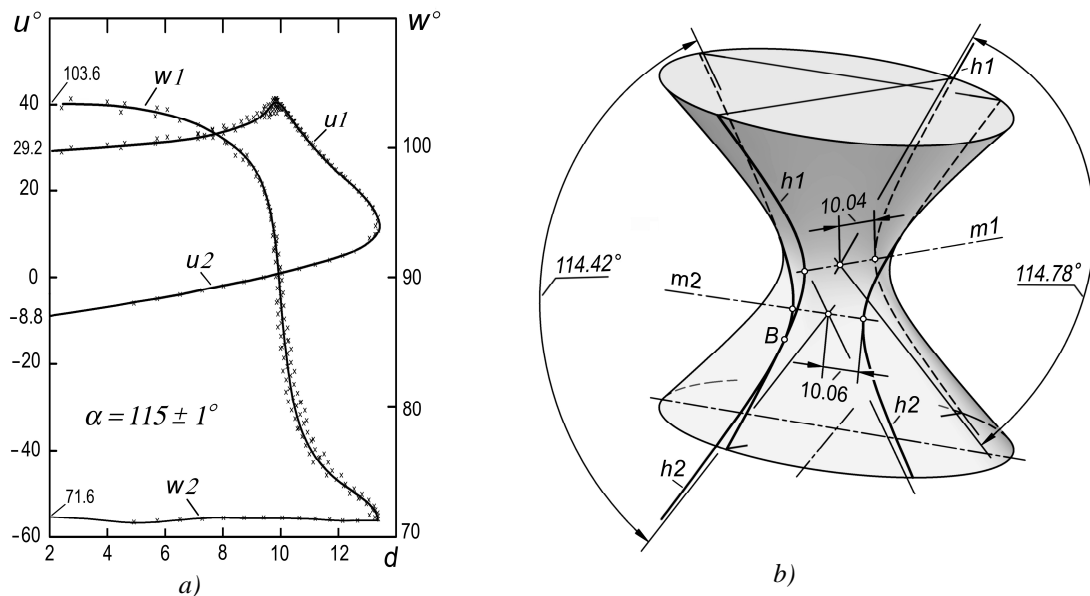
**Fig. 8. Hyperbolae with exterior center for  $\alpha = 65^\circ$ : a – is the change of angular coordinates of a sectional plane; b – hyperbolae for  $d = 15$  (two solutions)**

equal, than the number of solutions for obtuse angle can be determined in accordance with fig. 7 for additional aris. For example, the number of solutions for  $\alpha = 115^\circ$  and  $\alpha = 65^\circ$  is the same and equals two. Limiting point  $E$  has a responsive point  $E'$  (see fig. 5).

Let us draw a sample to  $\alpha = 115 \pm 1^\circ$  (fig. 9). The number and coordinates of degenerated hyperbolae

are the same (see the values of  $u$ ,  $v$  if  $d = 0$ ) as for the angle  $\alpha = 65^\circ$  (see fig. 8, a above). However the cinematics of movement of a sectional plane is considerably different (fig. 9, a). Throughout the whole interval  $0 \leq d < d_{max} \approx 13.5$  two solutions are kept. At  $d_{max}$  two combined hyperbolae are obtained.

As an example of hyperbola with inner center from database we shall get hyperbolae  $h1$ ,  $h2$  if  $d = 10$  (fig.

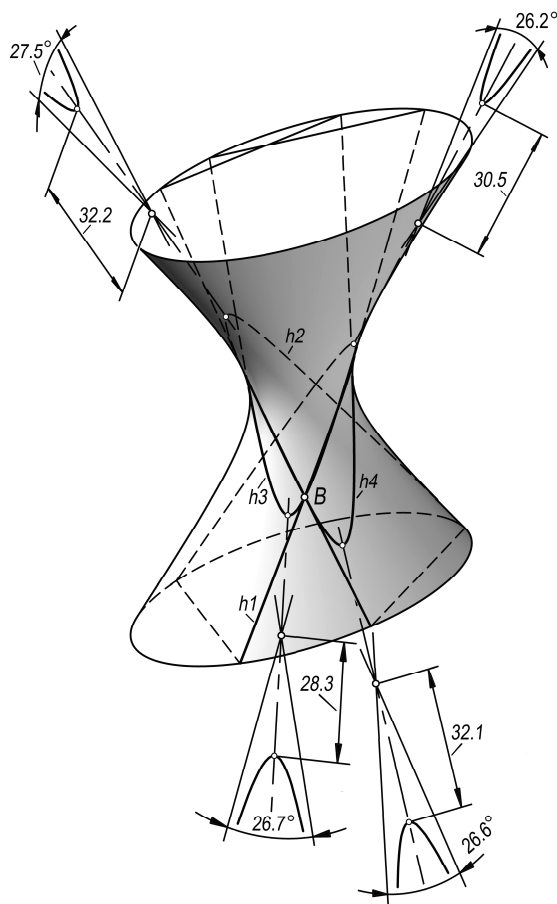


**Fig. 9. Hyperbolae with inner center: a – angular coordinates of a sectional plane; b – hyperbolae  $\alpha = 115^\circ$  and  $d = 10$  (two solutions)**

9, b). We see that real axes of these hyperbolae  $m1$ ,  $m2$  are perpendicular to the axis of  $OH$ . The errors  $del$  of hyperbolae  $h1$ ,  $h2$  are 0.55% and 0.30% correspondingly. Hyperbola  $h2$  is front-projecting (for it  $u2 \approx 0$ , see fig. 9, a).

Having taken metrics  $\alpha = 27^\circ$ ,  $d = 30$  we will get the solution with four hyperbolae with inner center (fig. 10). This corresponds to fig. 7, where the horizontal straight line  $\alpha = 27^\circ$  crosses the curves 1, 2 summarily in 4 points. Only one half-hyperbola is on the bay of  $OH$  of each hyperbolae  $h1 \dots h4$ . The second half-hyperbola is built on the basis of five points of the first half-hyperbola [4]. Because of insufficient density of database in this area the found hyperbolae have serious errors which are having  $del = 2.3$ , 4.6, 3.4 and 4.2% for hyperbolae  $h1 \dots h4$  correspondingly.

Improved accuracy of solution to any desired value is achieved by creating the  $DB$  higher density, the use of interpolation or the transition to straight line skeleton from  $OG$  [6].



**Fig. 10. Four solutions of hyperbola with  $\alpha = 27^\circ$ ,  $d = 30$**

## SUMMARY

1. 3D method of computer geometric simulation allows us to find the solution of the problem under consideration with the required accuracy and studying its mechanism including the sphere and number of solutions.

2. 3D method in a combination with the model of a “black box” and programming can be recommended for practical problems of geometric simulation, in which the construction of analytical and geometric models is complicated or unreasonable due to their difficulty.

3. 3D method doesn't require projection and analytical constructions and can be considered as an independent method of geometric simulation.

## REFERENCES

- [1] Даниленко, Я.М. Определение секущей плоскости по заданному коническому сечению / Прикладная геометрия и инженерная графика / Я.М. Даниленко / Киев: Будівельник, 1967. – Вып. 5. – С. 146 – 152.
- [2] Иванова, Г.Г. Построение сечений заранее заданной формы / Прикладная геометрия и инженерная графика / Г.Г. Иванова / Киев: Будівельник, 1967. – Вып. 5. – С. 150 – 152.
- [3] Пеклич, В.А. Мнимая начертательная геометрия: учебное пособие / В.А. Пеклич – М.: Издательство АСВ, 2007. – 104 с.
- [4] Хейфец, А.Л. Компьютерные 3d алгоритмы в курсе геометрического моделирования (на примере задачи совмещения коник с квадраками). / А.Л. Хейфец // Труды 18-той международной научно-технической конференции “Информационные средства и технологии. Москва 19-21 октября 2010”. – М.: Издательский дом МЭИ, 2010. – Т.3. – С. 110 – 117.
- [5] Хейфец, А.Л. Алгоритмы 3D компьютерного геометрического моделирования / Вестник Южно-Уральского государственного университета. Серия “Компьютерные технологии, управление, радиоэлектроника”. – Челябинск: Изд-во ЮУрГУ. 2012. – Вып. 15. №3(262), с. 57–62.
- [6] Хейфец А.Л. 3D-модели линейчатых поверхностей с тремя прямолинейными направляющими / А.Л. Хейфец, А.Н. Логиновский // Вестник Южно-Уральского государственного университета. Серия “Строительство и архитектура”. – Челябинск: Изд-во ЮУрГУ. 2008. – Вып. 7. №25(125), с. 51–56.

# Robust Affinity Propagation using Preference Estimation

Kai-Chao Yang  
National Tsing Hua University  
30013, Hsinchu, Taiwan  
kcyang@vc.cs.nthu.edu.tw

Chang-Hsin Yu, Chen-Jung Hung  
National Tsing Hua University  
30013, Hsinchu, Taiwan  
chyu@vc.cs.nthu.edu.tw

Jia-Shung Wang  
National Tsing Hua University  
30013, Hsinchu, Taiwan  
jswang@cs.nthu.edu.tw

## ABSTRACT

Affinity propagation is a novel unsupervised learning algorithm for exemplar-based clustering without the priori knowledge of the number of clusters ( $NC$ ). In this article, the influence of the “preference” on the accuracy of AP output is addressed. We present a robust AP clustering method, which estimates what preference value could possibly yield an optimal clustering result. To demonstrate the performance promotion, we apply the robust AP on picture clustering, using local SIFT, global MPEG-7 CLD, and the proposed preference as the input of AP. The experimental results show that over 40% enhancement of ARI accuracy for several image datasets.

## Keywords

Affinity propagation; classification algorithms; clustering method; image classification

## 1. INTRODUCTION

Traditionally, clustering problems are solved by learning a set of centers to minimize the sum of squared errors between data points and their nearest centers. These centers might be selected from actual data points called “exemplars” or virtual ones. A representative method is the K-centers clustering algorithm. Different from the traditional clustering algorithms, Frey and Dueck proposed an unsupervised clustering algorithm called Affinity propagation (AP) [Jou01a], which considers every data point as a potential exemplar and iteratively exchanges messages between data points to determine the most suitable exemplars. In [Jou01a], AP has shown its outstanding performance in several areas, such as face detection, gene and exon finding, representative sentence detecting, and air-travel routing. Since the proposal of AP, there have been a lot of relative discussions about it. Some researches focus on variations of AP [Con01a] or combination of K-centers algorithm and AP [Con02a].

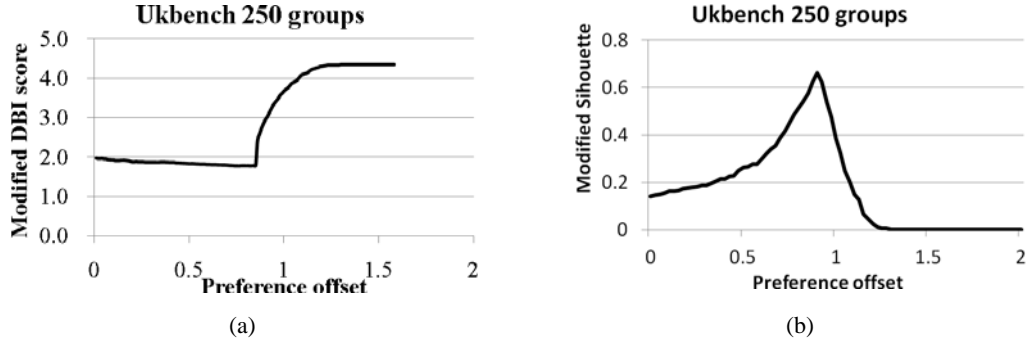
Despite the topics extended from AP, this article focuses on improving the accuracy of AP output. The input of AP consists of two parts, the similarity matrix and the preference. Because AP does not need to pre-assign the number of clusters ( $NC$ ), the similarity and the preference are the keys of the

clustering results of AP. There are rare discussions about the preference. In [Jou01a], the preference value is suggested to be the median of the similarity matrix. However, according to the observation, the decision of the preference value has a significant impact on clustering results. A median preference value may not lead to the optimal clustering result. In [Jou02a], a method to scan preference for finding the optimal clustering solution was proposed. Preference changes adaptively in the process of AP to find more reliable convergence, but progressive scanning takes a lot of time. Besides, the definition of “optimal solution” was not clearly defined in the paper.

In this article, we use some famous criteria that have been frequently used as the clustering accuracy indices of the AP output, such as the Adjusted Rand Index (ARI) [Jou03a], the Davies-Bouldin Index (DBI) [Jou04a], and the Silhouette index 0 [Jou05a] to measure the accuracy of the AP output, and discuss the relationship between the preference and clustering results. We further extend the observations about the preference in [Url01a] to a preference estimation algorithm, which could possibly find out the most suitable preference value. The estimated preference can significantly improve the accuracy of AP. Besides, it can be used in all related researches mentioned above as long as they apply AP for clustering. For demonstration, we choose some image databases as the source material, and then classify images using AP with the proposed preference estimation.

The remaining part of the article is organized as follows. In Section 2, we first address the importance of the preference to AP. Then, according to our observation, the robust AP using preference

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.



**Figure 1. (a) Modified DBI scores and (b) Modified Silhouette Index scores of AP output using different preference values. The selected preference value is reference preference + offset, where the reference preference is set as  $\text{med}(\text{Sim}) + \min(\text{med}(\text{Sim}))$ .**

estimation is proposed. In Section 3, we apply the proposed robust AP to the image clustering to demonstrate its performance. Some experimental results are presented here. Finally, the discussions and future works are given in Section 4.

## 2. ROBUST AFFINITY PROPAGATION

### Preference of Affinity Propagation

Affinity Propagation (AP) views each data as a potential exemplar, and recursively passes messages between points until corresponding clusters emerge. In AP, there are two types of messages communicated between data points. The “responsibility”  $r(i, k)$ , sent from data point  $i$  to point  $k$  reflects how well point  $i$  favors the point  $k$  as its exemplar over other candidate exemplars. The “availability”  $a(i, k)$  sent from point  $k$  to point  $i$  reflects how well point  $k$  favors itself as an exemplar of point  $i$  over other candidate exemplars. For  $k = i$ , the  $r(k, k)$  represents the *preference* of data point  $k$  to be chosen as an exemplar.

The preference of AP plays an important role. When the preference of a candidate exemplar  $k$  is larger, the responsibility  $r(k, k)$  and the availabilities  $a(i, k)$  for all  $i$  are stronger, so that it is likely that the point  $k$  becomes an exemplar eventually. This means that the number of identified clusters is increasing with the preference correspondingly. According to the observation, when the preference is small enough, all data points will be classified into the same cluster. On the contrary, every data point will form a cluster itself when the preference is increasing to near zero. Note that the number of output clusters ( $NC$ ) is a non-decreasing function of preference. In [Jou01a], the preference is suggested to be the median of the similarity matrix. This leads to a moderate number of clusters. However, this selection may not lead to the optimal clustering result because it does not consider dataset content and may produce too many or too few clusters.

### Optimality of the Preference and Unimodal

In this sub-section, we discuss how to evaluate the accuracy of clustering results and how the preference affects the clustering results.

The clustering result is usually compared with the ground-truth to evaluate the accuracy. The adjusted rand index (ARI) [Jou03a] is a popular similarity measure of agreement between two partitions. However, there is no ground-truth for comparisons in many cases. In this situation, some other criteria can be used instead of ARI, such as the Davies-Bouldin Index (DBI) [Jou04a] and Silhouette Index [Jou05a].

Let  $a(x)$  be the ARI score at preference value  $x$ . Then we observe that  $a(x)$  tends to be a unimodal function. Thus there exists a point  $p$  such that  $a(x)$  is increasing for  $x \leq p$  and decreasing for  $x \geq p$ . Therefore, we can clearly define that, the optimal solution occurs at preference =  $p$ .

As for the DBI, we assume  $d(x)$  is the DBI score at preference value  $x$ . Note that lower DBI scores represent better clustering results. We find that  $d(x)$  monotonically decreases with  $x$ . This trend results from two facts:

- 1)  $NC$  is a non-decreasing function of the preference value  $x$ .
- 2)  $d(x)$  is a non-increasing function of  $NC$ .

DBI takes all data points into consideration, even though some data points only contain itself as singleton or twin. This situation results in the best score of  $d(x)$  for these extreme small clusters. As the preference value increases, these trivial clusters contribute much and results in the best DBI score. This is the main reason that ARI and DBI induce different results. In fact, the Silhouette Index that gives the optimal score as long as the preference is larger than a certain value also has the same problem. However, these extreme small clusters caused by large preference values are less informative comparing to those of large size, so it is not



---

```

top = 0;
bottom = med(Sim) + min(med(Sim));
cut = (top + bottom) / 2;
slope = (DBImodified(top) - DBImodified(cut)) / (top - cut);
FOR i = 1 TO round_number
  IF slope < 0
    top = (top - bottom) × 2 + bottom;
    bottom = cut;
    cut = (top - bottom) / 2 + bottom;
    slope = (DBImodified(top) - DBImodified(cut)) / (top - cut);
  ELSE
    top = cut;
    cut = (top - bottom) / 2 + bottom;
    slope = (DBImodified(top) - DBImodified(cut)) / (top - cut);
  END
END

```

---

**Table 1. Adjusted binary search.**

reasonable to give them better scores. Therefore, we modify the evaluating criteria in order to reflect the useful clustering information. The rule is: if the number of data points in a cluster is smaller than a threshold  $T$ , all points within this cluster are considered isolated and excluded from the evaluation.

Figure 1(a) and (b) show the modified DBI scores and modified Silhouette Index scores, respectively. The dataset is UKbench 250 groups [Url02a], and the preference value is selected as (the referenced preference + offset), where the offset (the x-axis) ranges from 0 to 2. Here the referenced preference is defined as  $\text{med}(\text{Sim}) + \min(\text{med}(\text{Sim}))$ , the sum of the median vector of the similarity matrix and the minimum value of the median vector of the similarity matrix. Here “Sim” denotes the similarity matrix, and “med” represents median. From Figure 1, it can be seen that the curve trend becomes a unimodal function after the removal of trivial clusters.

### Preference Estimation

According to the above simulations in Fig. 1, we found that it is highly likely that the above accuracy indices can be transformed to unimodal functions of the preference value. Thus we can estimate the preference that generates the optimal solution. No matter what accuracy index is used, the relationship between the preference and the clustering result of AP can be concluded as follows: Initially, when extremely small preference value is set, every data point is unlikely being the exemplar of other points. When the preference value increases, some data points start being accumulated as different clusters. Then more and more clusters are produced, and the attractive force of every exemplar is getting stronger as well. However, when the preference is larger than a certain value, too many exemplars will be generated. This scatters the attractive force of every exemplar again.

The next step is to search for the preference that generates the optimal clustering result. Table 1 shows the proposed adjusted binary search algorithm that has time complexity of  $O(\log n)$ . The proposed

algorithm takes advantage of the feature that a unimodal function has exact one extreme value. Since most online image databases have no ground-truth, it is unlikely to evaluate ARI scores for these images. Thus we put emphasis on the DBI, but any other index that can be transformed to a unimodal function can apply the proposed algorithm.

The search starts by calculating the modified DBI scores of two extreme points, preference =  $\text{med}(\text{Sim}) + \min(\text{med}(\text{Sim}))$  and 0. Then the point with higher scores is iteratively replaced by the median point between the prior two search points. The number of iterations decides the searching time and accuracy. After enough number of iterations, a near optimal preference can be obtained.

### Estimated Preference Adjustment

The preference estimation process can generate the best score of the accuracy index. However, for the same dataset we observe that different accuracy indices may imply different optimal preference values. For example, when we use Ukbench 250 image dataset as the input data, the estimated preference offset in the ARI is about 0.88. In Figure 1(b), the same result is observed in the modified Silhouette Index. However, in Figure 1(a), the estimated preference offset becomes 0.76 in the modified DBI. This situation is mainly because different criteria are used by different accuracy indices. Among these accuracy indices, the estimated preference in the ARI should be treated as the optimal preference because ARI scores are evaluated using the ground-truth.

From the simulation using different datasets, we observe that the quotient of the estimated preference offsets between the ARI and modified DBI is roughly fixed. Therefore, at beginning we can train a small number of data points to get the ARI score and DBI score. Then the quotient between them can be calculated. When taking all data points into account, the optimal preference offset can be approximated by the estimated preference offset in the modified DBI multiplying the quotient. Then AP is able to use this preference and the similarity matrix as its input.

### Sub-cluster Combination

Although the clustering result of AP using the proposed preference can get nearly optimal ARI scores, we still observe that there are some small clusters containing only one or two members. We call them sub-clusters. To solve this problem, we use AP again to generate new clusters with only exemplars of sub-clusters as input. If two exemplars are classified as the same cluster, all members of these two exemplars are combined as one big cluster. Because the influence of non-sub-clusters has been removed, AP can generate more accurate results than that taking all pictures as input.

Image dataset	ARI scores using median preference	ARI scores using preference estimation and sub-cluster combination		Enhancement (%)	
		9 runs	11 runs	9 runs	11 runs
Ukbench 250	0.633	0.717	0.810	13.21	27.97
Ukbench 350	0.545	0.723	0.743	32.74	36.40
Ukbench 450	0.542	0.746	0.761	37.69	40.59
Ukbench 550	0.536	0.726	0.739	35.42	37.85
Ukbench 650	0.494	0.689	0.713	39.63	44.42
Ukbench 750	0.483	0.570	0.707	18.06	46.50
Average	-	-	-	29.46	39.00

**Table 2. ARI score enhancement using preference estimation and sub-cluster combination.**

### 3. CASE STUDY

The proposed algorithm in this article can be used in all areas that are suitable for applying AP or other variations of AP. In this section, we applied AP in the image clustering to demonstrate the performance of the proposed algorithm.

We use SIFT and MPEG-7 to extract features from each image, and then calculate the similarity between images. The input of AP includes the similarity matrix and preference. After the similarity matrix has been obtained, the next step is to determine the preference. Initially, the modified DBI scores are calculated using the preference value equaling  $\text{med}(\text{Sim}) + \min(\text{med}(\text{Sim}))$  and 0, respectively. Then the adjusted binary searching algorithm is applied to estimate the optimal preference offset in the modified DBI. Next, the estimated preference offset multiplies 1.11 to approximate the ARI result. After that, AP is applied to classify images using this preference. Finally, the proposed sub-cluster combination is applied to combine sub-clusters, and produce the final clustering result.

Six different image datasets from Ukbench database [Url02a] are applied to demonstrate the proposed algorithm. UK benchmark database was proposed by Nister and Stewenius. In this database, we pick out 250, 350, 450, 550, 650, and 750 groups as our experimental datasets. We resize the shorter edge of every picture to 120 pixels, but keep the ratio of height and width unchanged.

The final preference we obtained is used to calculate its ARI score, so that our clustering result can be compared with the ground-truth. In Table 2, we show the ARI scores of AP output using the estimated preference and the ARI scores using preference

selection in [Jou01a], respectively. It can be seen that after 11 iterations, the ARI score improves on average 39%.

### 4. CONCLUSION

In this article, we proposed a robust AP method. Our contribution is to suggest an estimation procedure of what preference value can yield an optimal solution of AP output. First, we transform the accuracy index to a unimodal function. Then, the adjusted binary search algorithm is used to estimate the optimal preference. Furthermore, the sub-cluster combination is proposed to refine the clustering result. The case study of image clustering shows that 39% performance improvement can be achieved.

### 5. REFERENCES

- [Jou01a] Frey, B. J., and Dureck, D., Clustering by passing messages between data points. *Science*, Vol. 315, No. 5814, pp. 972-976, 2007.
- [Con01a] Zhang, X., Gao, J., Lu, P., and Yan, Y., A novel speaker clustering algorithm via supervised affinity propagation, *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4369-4372, 2008.
- [Con02a] Zhang, X., Wang, W., Nørvag, K., and Sebag, M., K-AP: generating specified K clusters by efficient affinity propagation, *IEEE International Conference on Data Mining*, pp. 1187-1192, 2010.
- [Jou02a] Wang, K., Zhang, J., Li, D., Zhang, X., and Guo, T., Adaptive affinity propagation clustering, *Acta Automatica Sinica*, vol. 33, no. 12, pp. 1242-1246, 2007.
- [Jou03a] Hubert, L., and Arabie, P., Comparing partitions, *Journal of Classification*, vol. 2, no. 1, pp.193-218, 1985.
- [Jou04a] Davies, D.L., and Bouldin, D.W., A cluster separation measure, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, no. 2, pp. 224-227, 1979.
- [Jou05a] Rousseeuw, P., Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, *Journal of Computational and Applied Mathematics*, vol. 20, no. 1, pp. 53-65, 1987.
- [Url01a] Preference and the Number of Clusters. Available: <http://www.psi.toronto.edu/affinitypropagation/faq.html#clusters/>
- [Url02a] UK benchmark database. Available: <http://vis.uky.edu/~stewe/ukbench/>

# Virtual Reality System as a Tool for Education

V. Kovalčík, J. Chmelík, M. Bezděka and J. Sochor

Masaryk University

Botanická 68a

602 00, Brno, Czech Republic

vit.kovalcik@mail.muni.cz; jchmelik@mail.muni.cz; mbezdeka@mail.muni.cz; sochor@mail.muni.cz

## ABSTRACT

Education in the field of computer graphics is attractive, but the necessity of strong theoretical background and the amount of work needed before achieving a non-trivial output could be discouraging for many students. To cope with inherent complexity of computer graphics applications, we designed a system intended to reduce implementation burden and help the students to achieve interesting results in less time. In our labs we use an in-house virtual reality system, which has been gradually improved by the work of our students and staff for a period of more than eight years. The foundation of the system is a modular architecture that allows rapid assembly of prototyped applications as well as easy implementation of extensions. Computer graphics students with even a fundamental level of knowledge are encouraged to participate in the lab projects. Working on system that allows rapid prototyping and shortens the path to achieving interesting visible results, builds up the students' motivation and it is a key element of a continuous improvement during the study.

### Keywords:

Virtual Reality, Computer Graphics, Education.

## 1 MOTIVATION

Computer graphics is regarded as one of the most attractive parts of computer science. However, at the beginning students often do not realize the amount of work that is necessary to create a new non-trivial graphics application. When they realize this fact, it might decrease their motivation and diminish the enjoyable part of the learning. However, much of the work has been already done in the past. For their lab assignments, students can use existing graphical libraries, which implement the basic and repetitive parts of the applications and leave the student to do the creative part of work. Unfortunately, the higher level interface is often missing, which would allow the user to connect the modules into a loosely coupled systems and to support assembly of complex applications from partial solutions.

To achieve better cooperation and allow rapid prototyping of virtual reality applications we designed a modular system VRECKO. Students can extend this system and build upon the work of their colleagues from previous years. This is similar to the educational concept of fixing and extending application used by Costantini et



Figure 1: 3D painting in a virtual environment controlled with Wiimotes™, with position of user's hands being tracked by the OptiTrack™ system. The application was developed using the VRECKO system.

al. [CMC09]. The products of students' work are usually rich visual applications. This is also motivating – it corresponds to the findings of Guzdial et al. [GS02], stating that creating multimedia application instead of working on abstract algorithms builds up the motivation.

Similar long-term projects were created in previous years. ShadowLight [Lee05] is a modular system for rapid prototyping in virtual reality created at University of Illinois. Pettersen et al. described a Virtual Laboratory [PJS03], a 3D network-distributed virtual environment for education of computer graphics programming. Similarly, a system VRUT (developed at the

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Czech Technical University in Prague) was designed as an experimental VR system and it is used for education as well as in industry. For the augmented reality applications, the STUDIERSTUBE [FLSG98] software framework gained much attraction in the past decade.

## 2 HISTORY AND CONCEPT OF THE SYSTEM

Virtual reality system VRECKO<sup>1</sup>, was designed to support the research in Human-Computer Interaction laboratory at Masaryk University. The system was originally developed as a part of two dissertation theses. The first one modeled the concept of “object with changeable roles” [Mar04] and second one used VRECKO framework for development of applications for prototyping various interaction tasks in virtual reality [Fla05].

From educational point of view, common graphic applications are usually designed for a specified purpose and difficult to extend beyond the original scope, while graphic engines and graphic libraries are too generic for rapid implementation. We tried to strike the balance and develop a system that fulfills the basic tasks in interactive applications and offers a high degree of flexibility.

The system allows the programmer to create scenes composed of objects and to define the behavior of each object by adding one or more roles, called *Abilities*. The potential of *Abilities* range from changing object color or adding a grass onto a surface to the navigation and movement of a car in a virtual city.

The user can designate connections between the *Abilities* which are then able to communicate using messages, originating usually from some user’s action. Alternatively, an *Ability* can be scheduled to execute in periodic time intervals without the need of any external message.

Recently, the system is used as a basis for several research projects and in many undergraduate projects.

## 3 USAGE ADVANTAGES

The usage of our system is advantageous in many ways. First of all, it supports various kinds of input/output devices. The output devices include stereo projection on a large screen or head-mounted displays. The input devices include PHANTOM<sup>®</sup> haptic devices, OptiTrack<sup>™</sup> camera tracking system, Nintendo<sup>®</sup> Wii<sup>™</sup> or Microsoft<sup>®</sup> Kinect<sup>™</sup> among many others. A support for new devices can be quickly added when necessary.

All devices are regarded as special objects that are able to communicate via messages. Using this approach, the same application can be controlled by the OptiTrack<sup>™</sup> system and displayed on the stereo projection wall, as well as with a mouse and on a common computer screen. The concept of replaceable device objects allows students to work either in the laboratory or on their computers at home and also to present the final work outside the laboratory.

Another advantage is standardization of the environment which is easily understandable and the knowledge is transferable. If a student decides to do their work in the laboratory room, they can discuss the issues with their colleagues, which strengthens the spirit of the teamwork and speeds up the process of passing experience and learning new things.

## 4 LEARNING TOPICS

At the beginning of a semester, the students are offered with a list of possible research topics. Alternatively they can also come up with their own suggestions. Some projects may span over more than one semester, and may even converge to bachelor or master theses.

Throughout the years several main branches of the system utilization had evolved, with the needs for specific extensions. In the following parts we will describe the main directions of the development and give examples of student’s contributions.

### 4.1 Editor

The oldest branch in development is the Editor application for creating and editing 3D scenes [KFS07]. The Editor itself uses a set of specifically designed tools implemented as *Abilities*. Students can easily implement their own tools extending the editor capabilities.

The common tools for interaction in VE are already implemented as a core functionality of the Editor. Such tools include: addition of objects to the scene; object movement; path definition for movement and animation of individual objects; camera setup and movement; removal of objects.

In the previous years, a number of high quality specialized tools have been developed by the students as a practical part of their bachelor and diploma theses. The most notable examples are:

- *Room Edit* for editing 3D room schemes, i.e. editing walls, creating and moving windows and doors and coloring the objects.
- *Constrained Movement* for fast placement of objects with pre-edited snapping capabilities.
- *FFD Editor* for free-form deformations of objects (see Figure 2).
- *Connect Editor* for seamless automatic aligning and connection of objects geometries.

<sup>1</sup> <http://decibel.fi.muni.cz/wiki/index.php/VRECKO>

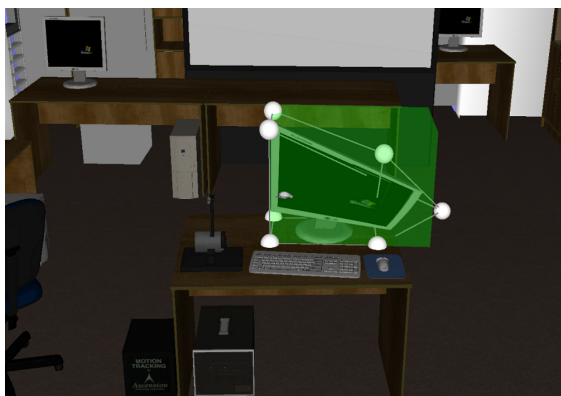


Figure 2: The Editor applied to the model of the HCI laboratory. The virtual LCD monitor can be deformed using a free-form deformation tool.

Students participating in this branch of development learn to implement and improve low-level algorithms for manipulation with objects, such as positioning, shaping, lighting and viewing, and another similar operations typical for 3D editors. By solving the elementary tasks students get acquainted with practical usage of quaternions, matrix transformations and other mathematics background frequently used in computer graphics.

## 4.2 Artistic applications

Another optional track for students' work is development of extensions for artistic applications (see Figure 1). Similarly to the *Editor* branch, there is also a layer providing common functionality. Therefore, students do not have to re-implement the code common to many artistic applications like basic GUI, objects manipulation, color selection, etc.

Currently implemented modules include:

- brush strokes based 3D painting,
- dance visualization,
- design of mathematical sculptures (see Figure 3),
- abstract performance art based on Perlin noise,
- exploration of chaotic attractors and fractals.

By implementing new art techniques, students can learn various topics of computer graphics. Availability of various input devices (like OptiTrack™, 3D mouse or Wiimotes™) also forces students to think about user interfaces and usability aspects in virtual environment.

## 4.3 Artificial World

Third direction of development is a set of applications for creating and displaying artificial worlds (see Figure 4).

Technically, the virtual world is based on a set of XML files, which can be arranged in a tree. Each of the files

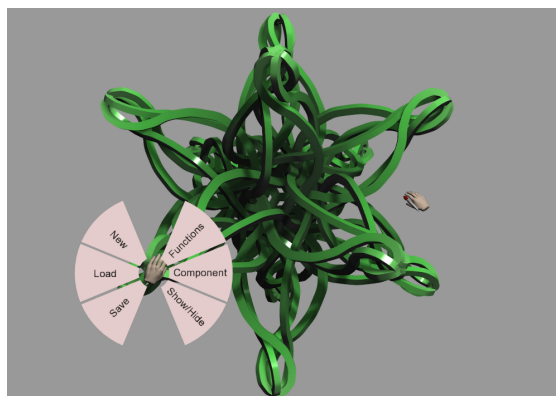


Figure 3: Designing a mathematical sculpture based on a dodecahedron.

contain one level of a world and references to the files containing the details.

The XML files hierarchy is created by a system of independent generators. Because of the independence of the generators, students have freedom to choose their preferred programming language and programming environment.

The currently implemented generators are:

- Planet generator allowing the user to generate the whole planet's surface by approximating the natural laws. Written in C.
- City generator, able to create a layout of a city on a given height-map and decide where to put roads and parcels and determine their usage. Written in Python.
- Building generator that can take parcels as the input and generate appropriate buildings or building blocks. Written in Delphi.

The freedom to choose the programming language has demonstrated the diversity of students' preferences. From a maintainer's perspective, this is not a perfect solution, but the generators are written in much less time, which is beneficial for the educational aspects, and each of the generators can always be easily replaced by a different one.

The generated world can be visualized in the VRECKO engine, but also enhanced by a number of plug-ins. Currently, we are using plug-ins for volumetric clouds, flocks of birds and car traffic simulations, that bring the city to life. We have also a parametric stochastic L-system able to generate high quality trees. All of the mentioned plug-ins were created by students. In this track the students learn mainly the processing of huge amounts of data stored in complex data structures.

## 4.4 Haptic interaction

Another set of applications use the VRECKO engine to perform fast multi-contact collision detection and



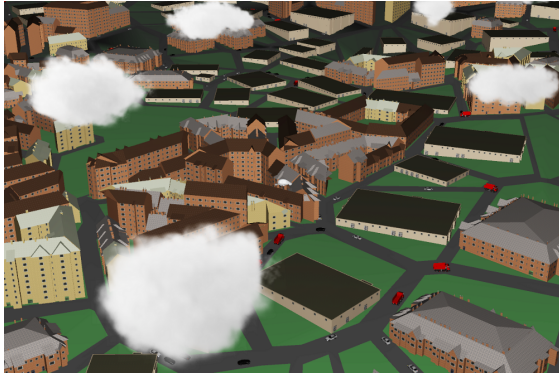


Figure 4: The Artificial World application, incorporating long-term work of many students.

to simulate haptic feedback, see [KKS11]. Projects cover the issues of different data preprocessing and different data structures for collision tasks evaluated in the scene. The collision detection algorithms are designed to be fast enough for the use with the SensAble® PHANTOM®.

While developing haptic algorithms, students learn how to create effective and optimized code which does not depend on the high performance of latest computer hardware and particularly the graphics cards.

## 5 EXPERIENCES AND RESULTS

The virtual reality and computer graphics in general is an attractive field for students. The chance to create interesting output alongside with the possibility to work with rare equipment presents a motivation to do better work. Aside from semester-long lab assignments, there were already more than 25 successfully defended bachelor and master theses that were based on the VRECKO engine.

The source codes of the best modules are added into the main distribution of the VRECKO engine and newcomer students can use them and learn from their code. The best applications are also presented to the public during numerous excursions, among others to high-school students interested in the study of Informatics. The fact of their work being actually used also manifest as a motivation boost for participating students.

The results proved to be attractive and the demonstrations of student's works draws attention of many prospective students to our laboratory and to our school. We do not collect any data confirming the shift in interest, but we already encountered students for which the excursion was a strong impulse to enroll the computer graphics program or other study programs at our school.

## 6 CONCLUSION AND FUTURE WORK

We have presented a virtual reality system, which is used for research and for education of students of the

computer graphics program. Its modular architecture and number of main directions of development allows students with almost any programming skill to participate in the development either by writing a module or assembling an application based on existing modules.

The core of the system proved to be well designed and it is essentially the same as it was in its very beginning, more than eight years ago. It is stable and the most of the older modules are still compatible with current versions and can be utilized by present students. In our opinion, the effort put into building an universal VR engine has paid off and now we can use it for research and education as well as for excursions to attract new students.

Nowadays, we are continuously extending the system and we plan to prepare a set of generic task assignments to cover the educational topics more evenly and for various levels of education. During design and evaluation of new art modules, we are also collaborating with students of fine arts.

## 7 ACKNOWLEDGEMENT

This work has been supported by the Student Project Grant at MU under the research program MUNI/A/0922/2009

## 8 REFERENCES

- [CMC09] G. Costantini, G. Maggiore, and A. Cortesi. Learning by fixing and extending games. In *Proceedings of Eurographics 2009*, 2009.
- [Fla05] Jan Flasar. *Interaction Techniques in Non-Immersive Virtual Environments*. PhD thesis, Faculty of Informatics, Masaryk University, Brno, 2005.
- [FLSG98] A. Fuhrmann, H. Löffelmann, D. Schmalstieg, and M. Gervautz. Collaborative visualization in augmented reality. *Computer Graphics and Applications, IEEE*, 18(4):54–59, jul/aug 1998.
- [GS02] Mark Guzdial and Elliot Soloway. Teaching the nintendo generation to program. *Commun. ACM*, 45:17–21, April 2002.
- [KFS07] Vít Kovalčík, Jan Flasar, and Jiří Sochor. Extensible approach to the virtual worlds editing. In *Computer graphics, virtual reality, visualisation and interaction in Africa / AFRIGRAPH'07*, pages 31–37, Grahamstown, South Africa, 2007. ACM.
- [KKS11] Vít Kovalčík, Lukáš Kalčok, and Jiří Sochor. Modular system for haptic interaction with multi-contact collision detection. In *Proceedings of the 11th International Conference on Computational Science and Its Applications (ICCSA)*, pages 24–32, 2011.
- [Lee05] Kalev H Leetaru. Shadowlight: an immersive environment for rapid prototyping and design. In *Proceedings of SPIE*, volume 5664, pages 606–615. Spie, 2005.
- [Mar04] Lubomír Markovič. *Object with Roles*. PhD thesis, Faculty of Informatics, Masaryk University, Brno, 2004.
- [PJS03] L. W. Petterson, N. Jensen, and S. Seipel. A Virtual Laboratory for Computer Graphics Education. In *EUROGRAPHICS 2003 Education Presentation*, 2003.

# Use the palmprint for identification and authentication of persons: a new real-time method of data treatment based on graph construction of spatial interest points

Alain Simac-Lejeune  
Compilatio  
276, rue du Mont-Blanc  
74540, Saint-Félix, France  
alain@compilatio.net

## ABSTRACT

The identification and authentication of individuals by their palmprints is a recent approach in the family of biometric modalities, very interesting in appearance and non-contact non-intrusive. It was recently studied in topic research over the last decade. The proposed approaches are mostly based on methods of classification and learning. However, the complexity of the calculations leads to an inappropriate application in real time. In our work, we investigated the use of basic primitives pictures more precisely the Space Interest Points (SIO) in a real-time process of identification and authentication of palmprint. This process is based on construction and matching of graph. By setting few constraints and working with matching methods and matching specific, experimental results suggest a robust real-time solution as good as the best methods with an error rate authentication below 1% for a population of 20 individuals.

## Keywords

interest point, biometrics, palmprint, identification, authentication

## 1. INTRODUCTION

Securing property and data is a very important topic. An effective way to perform the security is the use of biometric characteristics that is unique to each individual. Thus, there are physical characteristics such as fingerprints, palm prints, DNA, iris, retina, facial features, the morphology of the hand, the spectrum of voice, smell and even the veins hand but also behavioral characteristics such as keystroke dynamics, the dynamics of signatures or of the voice. However, any identification system is based on one or more characteristics selected under specific parameters:

- **universality**: each individual must possess the characteristics;
- **singularity**: it is possible to separate individuals from each other;
- **continuously**: resilience in everyday life and aging

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

- as well as intentional or unintentional alteration;
- **collectability**: ease of acquisition of the measure;
- **performance**: accuracy, speed and robustness of technology used;
- **acceptability**: degree of approval of the technology;
- **bypass**: ease of use of a substitute.

In the palmar recognition, the conditions are the same as for fingerprints (they are unique for everyone, including identical twins). Moreover, this method has many advantages. First, the capture system is cheaper than for iris recognition, the features of the hand are more numerous than those of fingerprints and can be determined with low resolution images. The system has a very important level of acceptability. The bypass is rare because it usually does not have a photo of the palm of his hand and it is difficult to take pictures without the knowledge of a person. The bypass is unlikely. However, when it is made, there is more possibility to use the system. Finally, this type of system is advocated in France by the *Commission Nationale Informatique et Libertés*.

Our goal is to establish a fast approach (near in real-time) for a web context while using an equipment inexpensive and available to all: the on-board cameras in computers (the most are in a resolution at least of 0.3 million pixels).

In this paper we propose a recognition method of hand without contact. The Section 2 presents the state of the art in the palmar area of recognition. Section 3 describes the feature extraction of the palm. Section 4 describes the process of creating the graph, the algorithm for generating and matching method. Section 5 and 6 show respectively the experimental results and give conclusions.

## 2. STATE OF ART

Biometrics tries to answer to two operations: the verification and the identification. The verification is the comparison of a captured biometric data with a stored template to verify that the individual is who he claims to be. The identification is realized by comparing a biometric data captured with those present in a database in order to identify an unknown person. Many systems have been proposed to perform the identification based on biometric characters and especially around the hand. Some previously methods use the recognition of the shape of his hand that is rely only on the geometry of the hand (Kumar et al., 2004). Its use 90 characteristics to create a three dimensional shape of the hand, the length, width fingers and especially the shape of the joints, but these methods require expensive equipment and generally require a contact between the device and the hand. So, quickly, the idea emerge to use palm prints to identify individuals.

Methods for authentication based on the palm of the hand (a state of the art is available in the publication of Kong et al., 2009 with all references in the assessment section) are approaches based on the lines of the hand.

There are also methods based on the binarization of Otsu and characterizations of hierarchical texture. In most of these methods, the computation time is important and images of palm used are of good quality. In a context of fast authentication via webcam, these are the two constraints that must be raising.

## 3. INTEREST POINTS AND GRAPH

### Spatial Interest Points

In 1988, Harris (Harris et al., 1988) is an extension of the 2D gradient to highlight points of interest space (SIP noted for "Space Interest Points") defined as points where appears an evident change of marked in the image. For example, corners, intersections, points and isolated points on textures are specific points of interest. In practice, these points of interest correspond to a pixel having a large curvature radius of the intensity and is in a second order variations. The SIP is defined from the Hessian matrix  $H$  like:

$$M(x, y) = I(x, y) \times \begin{pmatrix} \frac{\partial^2 G_s(x, y)}{\partial x^2} & \frac{\partial^2 G_s(x, y)}{\partial x \partial y} \\ \frac{\partial^2 G_s(x, y)}{\partial x \partial y} & \frac{\partial^2 G_s(x, y)}{\partial y^2} \end{pmatrix} \quad (1)$$

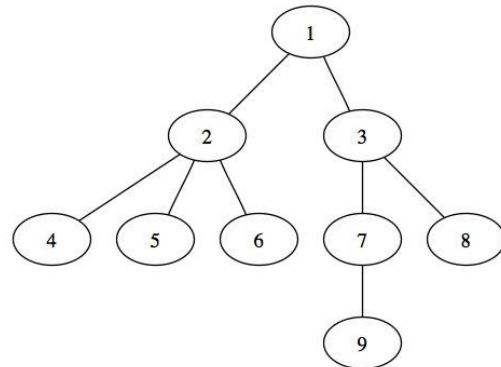
with  $I(x, y)$  representing the intensity of a pixel of coordinates  $(x, y)$  in the image  $I$   $G_s$  and  $(x, y)$  represents the impulse response filter performing Gaussian smoothing, mitigates the importance of the noise generated by the operations branch, and introduces a notion of scale factor. The extraction of points is carried out using a criterion of salience such as:

$$R(x, y) = \det(M(x, y)) - k \times \text{trace}^2(M(x, y)) \quad (2)$$

where  $k$  is the parameter used to manage the detection sensitivity of landmarks. Typical values of  $k$  are usually chosen in the interval  $[0.04, 0.15]$ . According to several authors, the experiment shows that a good value is obtained for a  $k$  about 0.04 (recognized as the best value in the community without any reference available on this).

### Rooted Tree

A rooted tree is a directed acyclic graph with a single root, and such that all nodes except the root have a single parent. This structure allows the use of recursive and technical courses in width and in depth. The root is the only node with no parent. The nodes are interconnected by an edge. The depth of a node is the distance, i.e. the number of edges from the root to the node. The height of a tree is the greatest depth of a leaf of the tree. The size of a tree is its number of nodes (counting the leaves or not).



**Figure 1.** Example of rooted tree with 9 nodes (1-9).

## 4. APPROACH

Two phases are considered: the insertion phase of a new identity in the system and the recognition phase of an individual. The insertion phase is to take multiple images of the palm of the hand and to extract the characteristic used to construct a signature. In our case, a signature consists of spatial landmarks to establish a particular graph (a rooted tree). This graph represents the stored signature. The

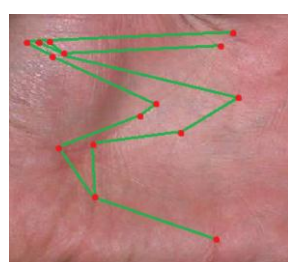
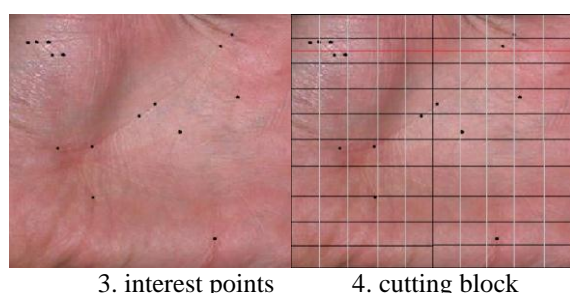
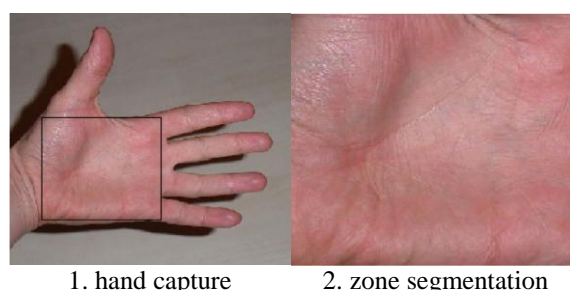


recognition phase is to extract features, and then build the signature matched the search.

### Signature creation

The extraction of landmarks is performed after pretreatment of isolating the hand to his environment (Fig. 2 - step 1). In this step, the hand is systematically rotated (continuous range of angles of rotation) to be always right fingers (arbitrary choice), which can be insensitive to the shooting conditions. The segment to retain only the main square in the palm of the hand (Figure 2 - step 2). This step is performed automatically using the method of (Doublet et al., 2006).

On this image we extract points of interest (Fig. 2 - step 3) with the following parameters: 1.5 for the sigma space and an adaptive threshold in order to obtain a maximal number of 25 points (usually around 180).



**Figure 2.** The different steps of the signature extraction of palmprint

The construction of the signature requires the establishment of a rooted tree from the interest points. The lowest point is considered like the root. Segmentation is realized horizontally and vertically into 10 zones allowing the construction of 100 cubes

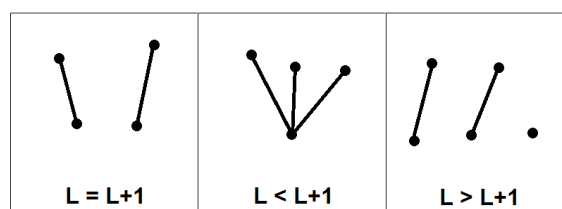
(Figure 2 - Step 4). We build from a point using the following algorithm:

### Algorithm 1 Creation of palmprint signature from SIP

```

1: for all line L do
2:   if NBpoints[l] = 1 then
3:     if NBpoints[l+1] = 1 then
4:       createLink(l,l+1)
5:     else
6:       for all points P (l+1)
7:         createLink(l,P)
8:       end for
9:     end if
10:  else
11:    if NBpoints[l+1]=1
12:      createLink(l,l+1-left)
13:    else
14:      if NBpoints[l]=NBpoints[l+1]
15:        then
16:          for each point n
17:            createlink(ln,l+1n)
18:          else
19:            for each point n
20:              createlink(ln,l+1n)
21:              and
22:              createlink(lmax,l+1n)
23:            end if
24:          end if
25:        end if
26:      end if
27:    end for

```



**Figure 3.** The various cases of production between lines

There are three cases of number of points of the line L: it is the same than that of the line L+1, it is greater than that of the line L+1 and is smaller than of the line L+1 (Fig. 3). We finally obtain a rooted tree (Fig. 2 - step 5) that represents the signature of the palm of the individual in question.

### Recognition of a signature

Several methods can be used to recognize a signature: the matching of graphs, the pattern recognition... We chose initially to use the specific algorithm (Peura, 2001) that allows mapping graphs with a very low cost. Further, it takes into account the type and provides a distance between trees allowing favoring speed of execution always with a view to obtain a real time system.

## 5. EVALUATIONS

To perform the evaluation of our method, we compared it to existing methods (Table 1). A database was developed specifically to validate our approach. The images were acquired with a Logitech webcam C500 at resolution of 1280x1024 set to capture at 640x480. The database contains 200 images from 20 individuals (10 women and 10 men ranging from 18 to 50 years) and 10 frames per individual. That is to say a database similar to almost all those used in the cited literature since their foundations were not directly accessible.

For all tests presented, the conditions are the same: plain background and light well known despite in our framework, it will need to be robust even in less favorable conditions. Finally, our approach shows interesting performances compared to other methods including the number of images required and the error rate.

<i>Method</i>	<i>Compare</i>	<i>U/A/E</i>	<i>TER</i>	<i>REF</i>
Ondelette	Euclidien	200/5/1	0.5%	Wu2002
LDA	-	300/10/-	0.82%	Wu2003
PCA	Euclidien	382/8/4	0.85%	Lu2003
ICA	Neurone	100/6/-	1%	Lin2010
Gabor	Euclidien	80/10/1	2.5%	Kumar2004
Ondelette	Manhattan	50/4/1	2%	Zhang2004
Gabor	Euclidien	80/10/1	1.6%	Kong2004
Filtrage	Euclidien	320/10/6	2.08%	Wu2004
Morpho.	Hamming	100/10/6	1.96%	Wu2005
Hu	Euclidien	189/10/2	1%	Noh2005
Gabor	Hamming	398/20/1	0.31%	Zhang2004
LBP	AdaBoost	50/10/2	2%	Wang2006
Hough	Hansdorff	100/6/3	1%	Li2006
<b>Your</b>	<b>Euclidien</b>	<b>50/4/3</b>	<b>0.54%</b>	<b>-</b>

**Table 1.** Comparison of palmprint recognition methods. A is the number of acquisition, by user U and E the number of images for enrollment. TER is the equal error rate.

## 6. CONCLUSION

In this paper, we investigated the use of biometrics and more particularly palm prints in order to perform identification and authentication of persons. The proposed approach, based on points of interest usable space in a web context, shows interesting results since it is rather reliable (low TER) and it has a rather low computation time. Finally, we can conclude that the proposed method with palmprint recognition is sufficiently robust and powerful to be used as a means of secure systems without the need for specialized equipment. It is now necessary to evaluate the system in a broader context and in less controlled conditions to validate its relevance.

## 7. REFERENCES

[Doublet2006] Doublet J., Lepetit, O., Revenu, M. « Reconnaissance biométrique sans contact de la main intégrant des informations de forme et de texture », CORESA, 2006.

[Harris1988] Harris C., Stephens M., « A Combined Corner and Edge Detector », Proceedings of the 4<sup>th</sup> Alvey Vision Conference, p. 147-151, 1988.

[Kong2004] Kong A. W.-K., Zhang D., « Competitive Coding Scheme for Palmprint Verification », Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 01, ICPR '04, IEEE Computer Society, Washington, USA, p. 520-523, 2004.

[Kong2009] Kong A., Zhang D., Kamel M., « A survey of palmprint recognition », Pattern Recogn., vol. 42, p. 1408-1418, July, 2009.

[Kumar2004] Kumar A., Shen H. C., « Palmprint Identification Using PalmCodes », Proceedings of the Third International Conference on Image and Graphics, ICIG '04, IEEE Computer Society, Washington, DC, USA, p. 258-261, 2004.

[Li2006] Li F., Leung M. K. H., « Hierarchical Identification of Palmprint using Line-based Hough Transform », Proceedings of the 18th International Conference on Pattern Recognition – Volume 04, ICPR '06, IEEE Computer Society, Washington, DC, USA, p. 149-152, 2006.

[Lin2010] Lin L., « Palmprint identification using PCA algorithm and hierarchical neural network », LSMS/ICSEE'10, Springer-Verlag, Berlin, Heidelberg, p. 618-625, 2010.

[Lu2003] Lu G., Zhang D., Wang K., « Palmprint recognition using eigenpalms features », Pattern Recogn. Lett., vol. 24, p. 1463-1467, June, 2003.

[Noh2005] Noh J. S., Rhee K. H., « Palmprint Identification Algorithm Using Hu Invariant Moments and Otsu Binarization », Proceedings of the Fourth Annual ACIS International Conference on Computer and Information Science, ICIS '05, IEEE Computer Society, Washington, DC, USA, p. 94-99, 2005.

[Peura2001] Peura, M., "Attribute trees as adaptive object models in image analysis", Earth Sciences, vol. 113, p. 1-80, 2001.

[Wang2006] Wang X., Gong H., Zhang H., Li B., Zhuang Z., « Palmprint Identification using Boosting Local Binary Pattern », Pattern Recognition, International Conference on, vol. 3, p. 503-506, 2006.

[Wu2004] Wu X., K.Q. W., Zhang D., « Palmprint recognition using directional line energy feature », Systems and Cybernetics, p. 475-478, 2004.

[Wu2005] Wu X., K.Q. W., Zhang D., « Palmprint recognition using valley features », Machine Learning and Cybernetics, p. 4881-4885, 2005.

[Wu2003] Wu X., Zhang D., Wang K., « Fisherpalm based palmprint recognition », Pattern Recogn. Lett., vol. 24, p. 2829-2838, November, 2003.

[Zhang2004] Zhang L., D. Z., « Characterization of palmprints by wavelet signatures via directionals », Systems, Man, and Cybernetics, p. 1335-1347, 2004.

# Map Point-Labeling with Rotation in Slider Model Using an Efficient Evolutionary Algorithm

Amir Zafar Asoodeh  
Department of Computer  
Science and Engineering  
Science and Research Branch,  
Islamic Azad University  
Tehran, Iran

Farshad Rostamabadi  
Department of Computer  
Science and Engineering  
Science and Research Branch,  
Islamic Azad University  
Tehran, Iran

Ali Ahmadi  
Electrical & Computer College,  
Khajeh Nasir Toosi University of  
Technology  
Shariati St., Seyedkhandan,  
Tehran, Iran

## ABSTRACT

Given  $n$  point coordinates and their various labels' length, our algorithm places a rotated collision-free label for each point. Using a combination of genetic algorithms and simulated annealing as an evolutionary algorithm, with qualification function consuming just  $O(n \log n)$  time, we achieve a fast near-optimal algorithm.

## Keywords

Map-Labeling, Point Feature Label Placement, Rotation, Slider Model, Genetic Algorithms, Simulated Annealing, Sweep-Line Algorithm.

## 1. INTRODUCTION

Map-Labeling is a crucial step in map generation and usage. Automated label placement, in its simple form is to automatically attach labels to special features of maps i.e. points, lines and areas. Point Feature Label Placement (PFLP) is one of the remarkable sub-problems of automated label placement that has received good attention. In a valid label placement, labels should be located adjacent to their feature points (could be attached to or parted with a defined space) and they must be pairwise disjoint. Moreover, map clarity is an optional parameter argued in some documents on account of its emotional and human-based nature in maps. The first approach towards automated map-labeling belongs to Edward Imhof in [Imh75a] who tried to distinguish different steps of labeling and gave a systematic solution for all features of maps. After proving time complexity of map-labeling problem which is NP-complete in [For91a], heuristic approaches to solve the problem arose significantly. The first heuristic-based map-labeling solution was published in 1984 by Noma in [Nom84a] which placed labels according to an abridged algorithm in non-colliding space. Wagner and Wolf in [Wag95a] implemented a heuristic approach with a quality guaranty of 50 percent of the optimal solution and running time  $O(n \log n)$  in all situations. As PFLP evolved, innovative map-

labeling techniques were proposed in order to fulfill unwanted targets. Zhu and Poon heuristic Map-Labeling solution in [Zhu99a] which placed a non-intersecting pair of circular or rectangular labels for each point on the map, is one of the obvious illustrations of this concept. Evolutionary Algorithms (EA) are widely employed in complicated optimization problems. However, Genetic Algorithms (GA) as a sub-category of EA's became prominent due to their power in optimization and parallel processing. The first GA approach to map-labeling was represented by Djouadi in [Djo94a] which was comprised of procedures to calculate overlap and aesthetic constraints on maps to place labels. There are distinct versions of GA-based solutions that are argued in [Dij00a, Bae10a]. One of the main reasons which make fast automated map-labeling seem less perspicuous than the human-made one is restricted candidate space to place labels in the automated technique. Consequently, map-labeling in slider model, which allocates an approximate continuous candidate space to labels, was introduced by Strijk in [Str02a]. Labels' intersection detection is the main part of the evaluation function of map-labeling heuristic approaches, undoubtedly. In this paper, the collision detection procedure originates from Bentley\_ Ottmann algorithm that was proposed in [Ben79a].

Applications for this problem can be found in computer graphics, GIS, Navigation systems, Computer games, flight animation and in other related fields.

The remainder of the article contains a complete description of the problem, including search space and cost function, the explanation of different parts of the algorithm and the results that were experimented to show the efficiency of the algorithm.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

## 2. DEFENITION OF OBJECTIVES

The problem is precisely defined as follows: We are given a valid labeling  $L$  composed of points  $P = \{p_1, p_2, \dots, p_n\}$  and variable length rectangular labels  $L = \{l_1, l_2, \dots, l_n\}$  where  $l_i$  is attached to  $p_i$  on the edges or vertexes (depending on labels' relocation values). For each position of map points in  $P$ , the problem is to make  $l_i$  values in a way that the minimum intersections occur. Additionally, it can be thought of as a combinatorial optimization problem with evident search space and cost function which are discussed afterwards.

### 2.1 Relocation Techniques

On account of decreasing the overlap between map features and labels, map providers use some replacement methods to position the labels in non-colliding space. We use two relocation techniques described as follows:

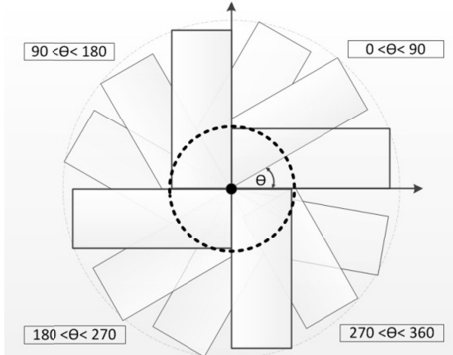
#### 2.1.1 Rotation method

It includes a rotation of rectangular label around the corresponding point with a degree ranging from 0 to 360. Based on the slider model, we utilize this method in a continuous space in order to increase candidate places. Figure 1 depicts the rotation method around a map point.

Note that  $\Theta$  indicates the degree between the horizontal axis and the rectangle's length which connects map point and label's vertex horizontally in the initialization stage.

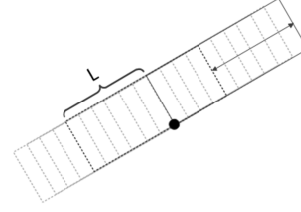
#### 2.1.2 Translation Method

This method incorporates translation with maximum displacement equal to the labels' length. It also employs slider model to relocate labels around the points. Figure 2 illustrates the translation method. Notice that  $L$  is the translation displacement value between 0 and each label's length.



**Figure 1. Demonstration of various label positions under the sliding label scheme with rotation method**

The relocation methods are called in an ordered sequence during algorithm runtime. By virtue of the aforementioned slider-based label replacement



**Figure 2. Presentation of a range of label positions in slider model with translation method**

Techniques, it seems that the candidate space required to attach labels to dense and complex maps is satisfied.

### 2.2 Cost Function

In combinatorial optimization problems, cost function is an estimation of distance between the optimal and the existing solution. Hence, defining a rational and reasonable cost function leads to higher proficiency. In this paper, cost function is comprised of two different values:

- 1-Overlaps of the candidate label with the other labels.
- 2-Clarity preference.

The combination of these two items can be used as a suitable cost function in order to clarify the closeness of the solutions to the optimal result. Let  $C$  be a set of  $n$  label cost,  $C = \{C_1, C_2, \dots, C_n\}$  where  $C_i$  belongs to  $L_i$  from labels. The attributes of label cost are defined as follows:

- $c_i$ : number of overlaps of  $l_i$  with  $\forall l_j \in L, i \neq j$
- $p_i$ : a penalty of the current position. In this paper, the inclination of initialized and relocated label of each point is the main parameter of this clarity option. In other words, labels with less rotation from horizontal axis receive lower penalty. If  $\alpha$  is the degree of rotation of a relocated label, positions between 0 and 90 ( $0 \leq \alpha \leq 90$ ) are considered as preferred locations and the penalty value of these candidate spaces are set to 0 and all the remaining positions have penalties equal to 1.

$C_i$  is then defined as follows:

$$C_i = a_1 \cdot c_i + a_2 \cdot p_i$$

Where  $a_1$  and  $a_2$  are constant values. In our experiment 1.0 and 0.1 are used for  $a_1$  and  $a_2$ , respectively. It is important to notice that owing to attached labels to the points in all degrees of rotation and translation variation, the algorithm does not need to calculate point/label collisions. Regarding that the cost function of map-labeling is  $\sum_{i=1}^n C_i$ ,  $F(s)$  definition as a cost function for a given set of points  $S$  is as follows:

$$F(S) = \sum_{i=1}^n (a_1 \cdot c_i + a_2 \cdot p_i)$$

By utilizing this function, the algorithm chooses labels with fewer overlaps and straighter positions as lower cost candidate solutions.

### 3. THE ALGORITHM

The algorithm designed to solve the previously described map-labeling problem is an evolutionary algorithm that utilizes a mixture of GA and SA ideas to offer a near-optimal map labeling solution. In this section, the basic principles of suggested genetic algorithm will be stated.

#### 3.1. GA

In this paper, we adapt a GA to solve map-labeling problem. The algorithm consists of different parts that are described as follows:

##### 3.1.1 Chromosome Structure

In map-labeling problem, there are many variables which have the capability to be placed inside chromosomes. However, in order to prevent extra complexity, we endeavor to design chromosomes as plainly as possible. Figure 3 illustrates the chromosome design of GA algorithm. In this structure, each pair of  $R_i$  and  $L_i$  relates with a label.



**Figure 3. Presentation of chromosome structure**

Note that  $R_i$  belongs to the degree of rotation of each label and  $L_i$  is described as follows:

$$L_i = \frac{\text{TranslationValue}_i}{\text{Length}_i}$$

Where  $\text{Length}_i$  is the label's length that could vary for each point. By this structure, utilizing additional techniques like masking to avoid chromosome disruption after operation is not required. The chromosome size equals to  $2 \cdot (\text{point num}) + 1$  where the addition of one unit pertains to the chromosome fitness value.

##### 3.1.2 Genetic Operators

In this paper, we use two different GA operators which are called after fitness calculation of each generation.

**Crossover:** It generates two offspring from two parents by swapping the information beyond random points. Owing to the large size of chromosomes in dense maps, we use a two-point crossover operator to function more drastically.

**Mutation:** It evolves chromosomes by transmuting some bits identified in a random manner. Provided that the selected bit belongs to the degrees of rotation, the offspring mutated bit is calculated as

$$R_3^* = 360 - R_3.$$

And if the selected location contains information of a transformation displacement, the mutated bit is computed as

$$L_1^* = 1 - L_1.$$

The amount of altered bits in a chromosome by mutation operation is defined as the number of genes

divided by 10. In other words, each 10 genes in a single chromosome include one mutation. After calling GA operators, the offspring and parents are saved in a chromosome pool to launch the selection procedure.

##### 3.1.3 Selection Methods

We use four different selection methods to compare their proficiency in this special GA solution. The methods are:

- Rank selection
- Roulette wheel selection
- Elitist Roulette wheel selection
- Elitist Rank selection

In which elitist methods perform selection procedure by taking advantage of the former generation directly. Note that running time and selection accuracy are two criteria to measure the competence of the selection methods that will be discussed later.

##### 3.1.4 Fitness Method

We have benefited from Bentley-Ottman algorithm as the basis of GA evaluation function due to its distinct advantages and have adjusted it to the specification of the problem. If  $n$  represents the number of lines and  $K$  is the number of overlaps in the map, the time complexity of Bentley-Ottman algorithm equals  $O((n + k) \cdot \log n)$ . The labels in a map are rectangles with expected diverse sizes clearly. If Bentley-Ottman runs on a map with rectangular labels, the vertices of labels are added to the intersection points undesirably owing to the fact that they should not be considered overlaps between the labels. To make an adjustment, the following definition is assumed.

**Definition 1** reduced rectangle is described as a rectangle which contains sides decreased by  $\epsilon$  and it is denoted as  $R_\epsilon$ .

**Lemma 1** Overlap detection by Bentley-Ottman algorithm in a map which contains  $R_\epsilon$  labels has lower time complexity than that one in a map with rectangular labels.

**Proof** The number of intersections between lines in the plane affects running time in Bentley-Ottman algorithm undoubtedly. If  $k$  is the number of overlaps between lines,  $p$  is the quantity of points in the map and  $i$  is the number of intersections between labels, rectangular label map has the following equation:

$$k = 4 \cdot p + i$$

On the other hand, number of overlaps between lines in the map which contains  $R_\epsilon$  labels is defined as:

$$k = i$$

Thus, time complexity with  $R_\epsilon$  labels is calculated as  $O((n + i) \cdot \log n)$  resulting in lower running time.

### 3.2 SA Function

In this paper, we take advantage of the SA idea to develop dynamic operator rates. Temperature as a significant parameter of SA is interpreted as average amount of fitness in a GA generation. In other words, if a generation is more fitted than the preceding one, the temperature will be reduced and vice versa.

## 4. RESULTS

In order to establish more efficacious results, we have implemented two other solutions and have made a comparison with the previously introduced algorithm. Figure 4 shows the running time of GA-SA with sweep-line and  $R_e$  labels (suggested algorithm), GA with sweep-line and  $R_e$  labels which is constructed with constant rates and simple GA which deploys the basic line-by-line overlap detection algorithm to count intersections. Notice that the running time discrepancy between GA's with static and dynamic rates becomes prominent with 220 points or more which exist on a map. Moreover, with 100 points or fewer, the GA with static rates consumes less time due to computational overheads of GA with dynamic rates. However, the more points are added, the bigger difference appears between running time of considered GA's. A simple GA has the time complexity of 46.61% on average higher than the GA with sweep-line and 43.58% higher than GA-SA with sweep-line which reveals the swiftness of suggested map-labeling algorithm. Figure 5 depicts the efficiency of the suggested selection methods by illustrating the overlaps after 10 seconds past running time. As the figure shows, elitist roulette wheel selection method has always the least amount of overlaps in comparison with other algorithms. Moreover, elitist rank selection method ranked second finally in spite of its lower ability to reduce overlaps compared with roulette wheel selection in lower density Maps.

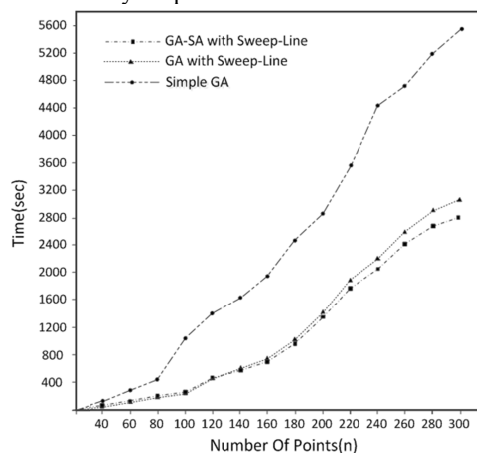


Figure 4. Running time comparison of three different GA algorithms

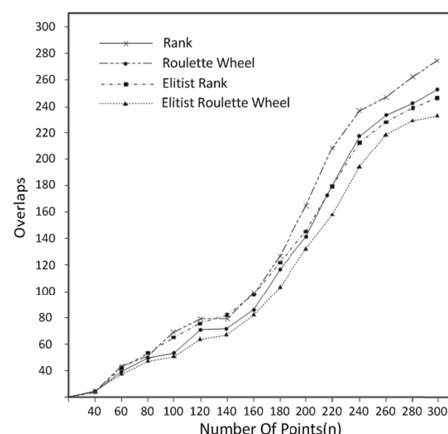


Figure 5. Performance Comparison of different selection methods

## 5. CONCLUSIONS

In this paper, we introduced an algorithm to label maps with new rotation techniques in slider model. Using a mixture of GA and SA ideas, the algorithm can perform swift and efficient labeling. By adapting the well-known segment intersection detection algorithm to the map point-labeling problem, we achieve a fast near-optimal solution.

## 6. REFERENCES

- [Bae10a] Bae W., Alkobaisi S. and Narayanappa S. and Y. Bae K., Convex Onion Peeling Genetic Algorithm: An Efficient Solution to Map Labeling of Point-Feature, SAC, pp. 892, 2010.
- [Ben79a] Bentley J. L. and Ottmann, T. A., Algorithms for reporting and counting geometric intersections, IEEE Transactions on computers, 1979.
- [Dij00a] Dijk S. V., Scalability and efficiency of genetic algorithms for geometrical applications, Springer-Verlag, pp. 683-692, 2000.
- [Djo94a] Djouadi M., Cartage: A cartographic layout system based on genetic algorithms, EGIS, 1994.
- [For91a] Forman M., A packing problem with applications to lettering of maps, ACM, 1991.
- [Imh75a] Imhof E., Positioning names on maps, Cartography and Geographic Inf. Science, 1975.
- [Nom84a] Noma E., Heuristic Method For Label Placement In Scatterplots, psychometrica, 1984.
- [Str02a] Strijk T. and Kreveld M. V., Practical Extensions of Point Labeling in the Slider Model, Springer, pp. 683-692, 2002.
- [Wag95a] Wagner F. and Wolff A., Map labeling heuristics: Provably good and practically useful, SoCG'95, pp.109-118, 1995.
- [Zhu99a] Zhu B. and Poon C. K., Efficient approximation algorithms for multi-label map labeling, Springer-Verlag, pp. 143-152, 1999.

# POSTER: Geometric modeling of pelvic organs with a discrete offset approach

Thierry Bay  
Aix-Marseille University,  
LSIS UMR CNRS 7296,  
Avenue Escadrille  
Normandie-Niemen, 13397  
Marseille, FRANCE  
thierry.bay@lsis.org

Romain Raffin  
Aix-Marseille University,  
LSIS UMR CNRS 7296,  
Avenue Escadrille  
Normandie-Niemen, 13397  
Marseille, FRANCE  
romain.raffin@lsis.org

Marc Daniel  
Aix-Marseille University,  
LSIS UMR CNRS 7296,  
Avenue Escadrille  
Normandie-Niemen, 13397  
Marseille, FRANCE  
marc.daniel@lsis.org

## ABSTRACT

In order to design a patient-specific simulator of pelvic organs, MoDyPe project considers the organs as thick surfaces. Starting from a closed parametric surface for the outer hull, an offset approach is applied. However, respecting the thickness on the surface and ensuring the absence of self-intersection is impossible if the shape has too important local curvatures. Two iterative approaches are compared. The first method is based on a B-spline formulation, and the second one on a mass-spring system (MSS). Our second method on discrete representation permits to obtain more accurate results with a more flexible formulation of the problem.

**Keywords:** Offset approach, parametric fitting, B-spline, mass-spring system.

## 1 INTRODUCTION

Through simulations of organs, a better understanding of poorly known disorders is possible. The problem we are interested in refers to the pelvic area, whose the organs can suffer from an imbalance in their spatial configuration. Although surgery is used to heal the patient, the impact of a surgical operation is difficult to estimate. Many tools have been developed for this purpose, requiring a virtualization of the environment [ZG12].

This work is integrated into a preoperative and patient-specific process (described in [BCR<sup>+</sup>11]) to develop a decision support software. The invasive procedures to displace the organs will be evaluated. The real-time constraint is relaxed to be faithful to physiological reality. From MRI of patients, a segmentation is carried out, followed by a geometric modeling to get meshes and a physical modeling to simulate the organs behavior. The geometric modeling is based on physiological reality by considering thick surfaces, i.e. a volume mesh with an internal cavity. But the MRI datasets can be poor quality, so the inner boundary of the organs can not be segmented because of the noise and erroneous points. Therefore, starting from a B-spline describing the surface of the organs [BCR<sup>+</sup>12], a volume mesh is created

with an offset approach by building a second surface at a given distance from the first one (the thickness of the membrane, alias the offset-distance).

Firstly, related work is detailed and offset problems are put forward, followed by a brief description of the offset process with B-spline surfaces. Afterwards, a discrete formulation of the problem is presented. A qualitative comparison is carried out between the results obtained by a B-spline offset and the discrete offset.

## 2 RELATED WORK

In our case, the creation of an offset consists in providing a thick membrane composed of hexahedra without degenerate or crossed elements to apply finite element calculations. The known problem of parametric offsets is the local and global intersections when the minimal main curvature is too large according to the offset-distance (cf. Figure 1). The problem is clearly considered in the literature (cf. [KN02] about the parametric offsets): removing the loops by working on each line and column [KSP02] for ordered datasets, curvature reduction by an iterative repositioning of the control points [SNL04], or parametric restriction [SEK06]. Finally these methods may not detect small self-intersections.

In another point of view, models based on active contours could be used [YPH<sup>+</sup>06]. The parameterization of a B-spline surface makes it possible to control easily the sampling, so that a quality conform mesh is created. However, the advantage of active contours is to make a seed evolve with a set of forces, without taking into consideration the connections with the external mesh. Their drawback in our process comes from automatic

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

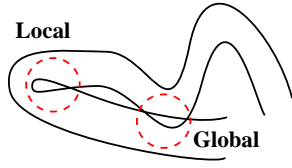


Figure 1: Local and global intersections for a curve.

feedbacks of the physical step that requires modifications in the surface sampling (i.e. local density change). These needs involve an expensive update of the mesh connections with the internal non-parameterized mesh to ensure the absence of crossed hexahedron.

### 3 THICKNESS AND B-SPLINE

Based on [BCR<sup>+</sup>12], a bidirectional energy function describes the connections between the B-spline sampling and the data points. The error is defined as a sum of distances between each sampled point with the closest data point in terms of Euclidean distance, and vice versa. A steepest descent method is used to reduce this energy and a fitting surface can be found.

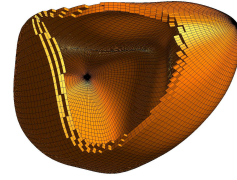
#### 3.1 Offset Operator and Process

From the surface fitted on the data of a patient (called *external surface*), the construction of an *offset-surface* at an *offset-distance* consists of four steps: uniform discretization of the external surface, computation of the normal at each point of its sampling, construction of the *offset-cloud* by moving each sampled point along the normal over the value of the *offset-distance*, and iterative fitting of a parametric closed surface on the *offset-cloud* to obtain the *offset-surface*. If several layers are needed by FEM experts to have a mesh with more uniform hexahedra, the same process is repeated with the value of the thickness divided by the number of layers.

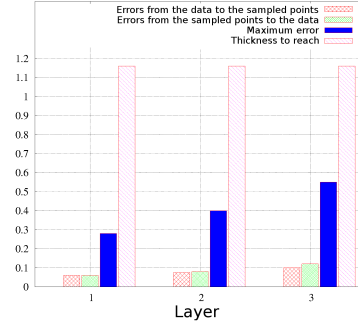
#### 3.2 Results

The cardinalities of the datasets are around 40K points. Since the real thicknesses are unknown, average values are used [SSS<sup>+</sup>10]: 3.5 mm for the bladder, 5.5 mm for the rectum. Uterus and vagina are not considered.

Figure 2(a) illustrates the bladder membrane. The offset does not have any problems because the shape is quite spherical, besides the thickness is not enough important to create global self-intersections. The histogram in Figure 2(b) sums up the errors during the process for each layer and the value of the local thickness to reach. The errors are the two energy functions [BCR<sup>+</sup>11] and the maximum error during the fitting. We notice that the maximum error for each new layer is low compared to the *offset-distance*. The results are different for the rectum. Figure 3(a) shows local self-intersections due to high curvature. Figure 3(b) illustrates effectively the problem of the maximum error between two layers.

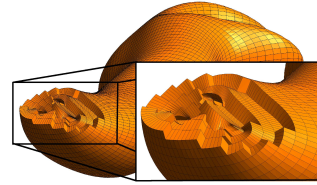


(a) Thickness with 3 layers

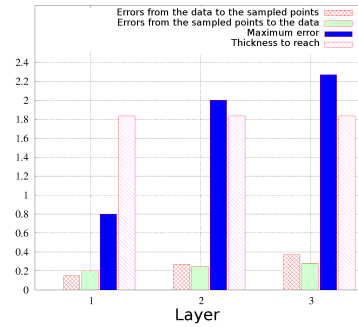


(b) Mean squared and maximum errors (mm)

Figure 2: Offset of a bladder wall.



(a) Self-intersections



(b) Mean squared and maximum errors (mm)

Figure 3: Offset of a rectum.

### 3.3 Assets and Limitations

For organs without complex shapes, the preceding method is satisfactory. The scattered knowledge on the real patient-specific thicknesses remains a problem. However, the use of a method based on the least squares has limitations. One of the solution is to apply a trimming algorithm as described in Section 2, nevertheless the parametric methods are unadapted especially for large *offset-distances*. Since we provide a mesh in output, working directly on the mesh would be an advantage. We have then developed a discrete approach for the offset construction based on a mass-spring system (MSS).



## 4 THICKNESS AND MSS

A MSS is designed to build the internal surface without accurate knowledge about the thicknesses. We differentiate our model from conventional mechanical approaches, since our objective is to use a dynamic approach to create a static mesh. In our case, no mass is considered. A set of  $(n+1) \times (m+1)$  particles  $\{P_{i,j}\}_{i,j=0}^{n,m}$  forms the system (the external surface sampling). Each one has a position  $p_{i,j}$ , a velocity  $\dot{p}_{i,j}$  and an acceleration  $\ddot{p}_{i,j}$  at a given time  $t$ . The system is governed by Newton's second law of motion  $F_{i,j} = \ddot{p}_{i,j}$ , with  $F_{i,j}$  the sum of all forces at  $P_{i,j}$ .

### 4.1 Creation of the Thickness

The choice of the lattice requires to define the field of application and the constraints to satisfy.

#### 4.1.1 Energies of the System

An external force  $F_{ext}$  (oriented as the normal vector at each  $P_{i,j}$ ) and a damping force  $F_{damp}$  (based on Kelvin-Voigt model) are added to the system. Finally, Hooke's law is used as part of internal force for elasticity and distances preservation:

$$F_{stretch}(P_{i,j}) = \sum_{k,l} k_{ij,kl} \left( (p_{k,l} - p_{i,j}) - L_{ij,kl}^0 \right) \frac{p_{k,l} - p_{i,j}}{\|p_{k,l} - p_{i,j}\|} \quad (1)$$

where  $k_{ij,kl}$  is the stiffness coefficient linking  $P_{i,j}$  and  $P_{k,l}$  if they are neighboring, and  $L_{ij,kl}^0$  is the rest length of the spring. Two springs are however added to constraint the internal mesh in an orthogonal position inside (cf. Figure 4). The total internal force consists of:

$$F_{int}(P_{i,j}) = F_{stretch}(P_{i,j}) + \hat{k}_{i,j} \left( \|S(u_i, v_j) - p_{i,j}\| - \hat{L}_{i,j}^0 \right) \frac{S(u_i, v_j) - p_{i,j}}{\|S(u_i, v_j) - p_{i,j}\|} + \tilde{k}_{i,j} \left( \|p_{i,j}^\perp - p_{i,j}\| - \tilde{L}_{i,j}^0 \right) \frac{p_{i,j}^\perp - p_{i,j}}{\|p_{i,j}^\perp - p_{i,j}\|} \quad (2)$$

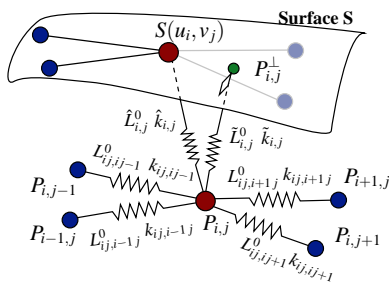


Figure 4: Lattice of the system.

#### 4.1.2 Integration Scheme

The total force of the system applied on  $P_{i,j}$  at time  $t$  is:

$$F_{tot}(P_{i,j}, t) = F_{int}(P_{i,j}, t) + F_{damp}(P_{i,j}, t) + F_{ext}(P_{i,j}, t) \quad (3)$$

The Euler explicit scheme is sufficient to compute the next position of  $P_{i,j}$  with small time steps:

$$\begin{cases} \ddot{p}_{i,j}(t + \Delta t) = F_{tot}(P_{i,j}, t) \\ \dot{p}_{i,j}(t + \Delta t) = \dot{p}_{i,j}(t) + \Delta t \ddot{p}_{i,j}(t + \Delta t) \\ p_{i,j}(t + \Delta t) = p_{i,j}(t) + \Delta t \dot{p}_{i,j}(t + \Delta t) \end{cases} \quad (4)$$

with  $\Delta t$  the fixed time step, whose the value must not exceed the natural period of the system [Pro96].

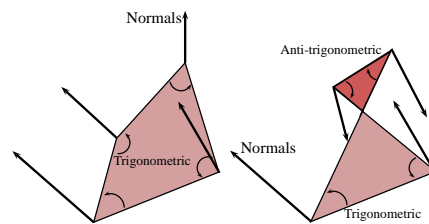
### 4.2 Parameters of the MSS

If we take into account the time step  $\Delta t$ , we end up with a set of  $12(n+1) \times (m+1) + 3$  parameters for the MSS. Determination of the optimal values of each parameter would be too much time-consuming and empirical.

Tests of local self-intersections are performed during the iterations, increasing these stiffness values settings if necessary. Similarly, the rest length  $L_{ij,kl}^0$  is replaced by  $L_{ij}^0$ . This length is defined initially through simulations, by increasing gradually the tension force until the OBB of the internal mesh has only 10% of its initial volume (fixed threshold). Concerning the parameters of the two additional springs in the lattice, the rest lengths are equals to the *offset-distance*, then  $\hat{L}_{i,j}^0 = d$  and  $\tilde{L}_{i,j}^0 = d$ . From an isotropic assumption for the external force, we set  $\hat{k}_{i,j} = \tilde{k}_{i,j} = K$ , with  $K$  the unique stiffness coefficient connecting the internal and external meshes. Finally, the parameters  $\gamma$ ,  $\eta$  and  $\Delta t$  are determined empirically, since the associated forces are difficult to match with physical quantities.

### 4.3 Correction of Local Self-Intersections

A local self-intersection is defined as an inversion of a quadrangle in the internal mesh. Our proposed solution consists in detected the crossed quadrangles and to increase locally the mesh tension with the stiffness parameters (their values do not go beyond a precalculated threshold to prevent instability in the system). The detection is achieved with oriented angles (see Figure 5). Moreover, this tension is spread on the 2-neighborhood, with a geometric decrease  $q^e$  (with  $q$  the proportion between two stiffness values, and  $e$  the depth of the neighborhood for the considered particle).

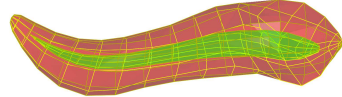


(a) Well-oriented angles (b) Crossed quadrangle

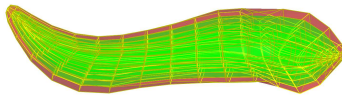
Figure 5: Oriented angles of crossed quadrangles.

## 4.4 Results

The input of the process is the external surface sampling (coarse for the visualization). Figure 6(a) illustrates the use of the total force for a rectum, building the internal mesh within the form. To improve the tests with the constitutive laws, the hexahedral mesh should be nearly uniform (they should look like as much as possible to cubes). Since the thickness is not neglected, several layers are generated by linear interpolation between the two layers (see Figure 6(b)).



(a) Application of the total force



(b) Linear interpolation for several layers

Figure 6: Construction of a rectum.

Tables 1 and 2 show a comparison between the discrete and parametric approach to construct the rectum thickness. For the parametric approach, we note that half as many cells have reached the expected thickness. In addition, the minimum value highlights the problem of proximity between the layers in some areas. With the discrete approach, the maximum is higher than with the B-spline. The *offset-distance* is indeed locally superior in the MSS to ensure stability with the internal force. A rate of 25% of elements remain to achieve, but the couple *tension/orthogonal force* is difficult to manage.

	Min.	Max.	Mean	St. dev.
B-spline	0.0004	6.4	3.89	1.96
MSS	2.3	11.6	6	1.77

Table 1: Analysis of the rectum thickness (in mm).

	> 5.5 mm (%)	> 90% of 5.5 mm (%)
B-spline	24.7 %	52 %
MSS	55.5 %	73 %

Table 2: Rectal wall thicknesses beyond 90% and 100% of the *offset-distance*.

## 4.5 Assets

The thickness management is handled by the combined action of external and internal forces. Local self-intersections are avoided by increasing the weight of the quadrangular tension on the mesh. Moreover, many of the system parameters can be estimated.

## 5 CONCLUSION

The parametric approach based on the least squares offers advantages for fitting problems. Obviously, in case

of high curvature of the surface, a proper offset can not be built. Our method allows us to have a direct control over the forces and corrects the local self-intersections. It is not based on a priori knowledge (corpus, skeleton) and is performant in noisy data environments (due to approximation and tension management). Results are validated by medical experts on experimental data. The hexahedral resulting mesh can be defined by FEM experts in terms of points density or number of layers.

Concerning the future works, a good asset would be to detect the global intersections. However the empirical dimension remains difficult to manage. A purely geometric orientation will therefore be investigated.

## 6 ACKNOWLEDGEMENTS

This work is supported by the French National Research Agency, under reference “ANR-09-SYSC-008”.

## 7 REFERENCES

- [BCR<sup>+</sup>11] T. Bay, J.-C. Chambelland, R. Raffin, M. Daniel, and M.-E. Bellemare. Geometric modeling of pelvic organs. In IEEE, editor, *33rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 4329–4332, Boston, USA, 2011.
- [BCR<sup>+</sup>12] T. Bay, Z.-W. Chen, R. Raffin, M. Daniel, P. Joli, Z.-Q. Feng, and M.-E. Bellemare. Geometric modeling of pelvic organs with thickness. In Atilla M. Baskurt and Robert Sitnik, editors, *Three-Dimensional Image Processing (3DIP) and Applications II*, volume 8290, pages 82900I–1–82900I–14, Burlingame, USA, 2012.
- [KN02] M. A. Kulczycka and L. J. Nachman. Qualitative and quantitative comparisons of B-spline offset surface approximation methods. In *CAD*, volume 34, pages 19–26, 2002.
- [KSP02] G. V. V. Ravi Kumar, K. G. Shastry, and B. G. Prakash. Computing non-self-intersecting offsets of NURBS surfaces. In *CAD*, volume 34, pages 209 – 228, 2002.
- [Pro96] X. Provot. Deformation constraints in a mass-spring model to describe rigid cloth behavior. In *Graphics Interface*, pages 147–154, 1996.
- [SEK06] J.-K. Seong, G. Elber, and M.-S. Kim. Trimming local and global self-intersections in offset curves/surfaces using distance maps. In *CAD*, volume 38, pages 183–193. Butterworth-Heinemann, 2006.
- [SNL04] Y. F. Sun, A. Y. C. Nee, and K. S. Lee. Modifying free-formed NURBS curves and surfaces for offsetting without local self-intersection. In *CAD*, volume 36, pages 1161–1169, 2004.
- [SSS<sup>+</sup>10] M. Schuenke, E. Schulte, U. Schumacher, L.M. Ross, E.D. Lamperti, M.M. Voll, and K.H. Wesker. *Neck and Internal Organs*, volume 20, chapter 2. Thieme, 2010.
- [YPH<sup>+</sup>06] P.A. Yushkevich, J. Piven, H.C. Hazlett, R. Gimpel Smith, S. Ho, J.C. Gee, and G. Gerig. User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. In *NeuroImage*, volume 31, pages 1116 – 1128, 2006.
- [ZG12] Bo Zhu and Lixu Gu. A hybrid deformable model for real-time surgical simulation. In *Computerized Medical Imaging and Graphics*, volume 36, pages 356 – 365, 2012.

# Convex Envelope Generation Using a Mix of Gift Wrap and QuickHull Algorithms

Charbel Fares

Holy Spirit University of Kaslik, Lebanon

charbelfares@usek.edu.lb

## ABSTRACT

The environment simulation is widely used nowadays. Training in many fields such as medicine and architecture heavily depends on virtual reality techniques. Since objects in real life do not have a deterministic shape it is not possible to have a geometric equation that might model them. Convex Hulls (or Convex Envelopes) are a must in such simulations. The need for convex envelopes rises with the intention of having realistic scenes with exact collision detection between objects in the virtual world. In this paper, four algorithms for generating the convex hull are discussed, implemented and compared. The first three algorithms are the Brute Force, the Gift Wrap and the QuickHull algorithm. The fourth one is a hybrid approach that combines the QuickHull and the Gift Wrap algorithms. Simulations were done in the medical environment, and algorithms are tested with the model of 3D wrist and knee bones.

**Keywords:** Convex Envelope, Medical Modeling, Computational Geometry, Virtual Reality.

## 1 INTRODUCTION

Virtual Reality (VR) is an advanced computer-generated technology which allows information to be displayed in a realistic environment that permits users and participants to interact with it. Recent developments in computer technologies have enabled VR to become a powerful product and analysis tool in computational science and engineering. Today, many studies and researches on surgical education depend heavily on VR simulators that become the training method in the medical area [1]. Simulators allow users and participants to examine and study parts and organs of the bodies, offering invaluable education for students and researchers. Unlike cadaver dissection that is not legally nor ethically accepted nowadays, VR models enable the user to perform on human body organs as if in real-world with anatomical accuracy and realism. Therefore, it reduces the risks to surgical patients and avoids the ethical issues associated with animal experimentation. Since computers are a way to simulate a physical environment, those environments are essentially geometric. Many of the computational problems involved in designing and building a VR system are geometric in nature. One principal important problem that must be addressed in order to make VR more realistic is the problem of real-time interactive collision detection.

Convex hull (CH) becomes a choice for modeling physical objects that do not have deterministic shape. The importance of the CH problems not only stems with Collision Detection (CD) but it has many applications such as cluster analysis, image processing and pattern recognition [2]. Other problems can be reduced to CH such as half-space intersections, Delaunay triangulation and Voronoi diagrams. Speeding up algorithms that compute CH is still a challenging issue in many fields and research areas in order to fulfill real-time requirements. In this paper a hybrid method to construct the CH and to decrease its running time is proposed. Simulations were done in the medical environment. The hybrid approach is an output sensitive algorithm that constructs the minimal convex envelope of a set of  $n$  points in two as well as three dimensions.

The rest of the paper is structured as follow: section 2 reviews some of the previous convex hull algorithms. Section 3 describes in details three different conventional techniques used to compute the convex envelope. Section 4 discussed the proposed Hybrid technique. In section 5 the results and simulations are shown. Finally in section 6 conclusions are given.

## 2 PREVIOUS WORK

One of the central problems in computational geometry is the computation of convex hulls. It has been an intensively studied subject up to present days. Early studies dealt primarily with the planer 2D case [3], then comes techniques for calculating CH in 3D space [4]. Brute Force is a simple algorithm that works in both 2D and 3D, it takes  $O(n^3)$  running time in 2D and  $O(n^4)$  in 3D. A lower bound algorithm presented by Yao [?] for computing the convex hull vertices in the quadratic decision tree model had a complexity of  $O(n \log n)$ . Another

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

approach known as Grahams scan achieves  $O(n \log n)$  running time in 2D. Jarvis March algorithm constructs the convex envelope in  $O(nh)$  time, where  $h$  denotes the number of vertices of the convex hull. This technique works also in 2D and it is output sensitive because it depends on  $h$  in its running time. Furthermore, 2D divide-and-conquer [5] is proposed after the sorting algorithms such as MergeSort and QuickSort and has  $O(n \log n)$  running time. Based on this algorithm Preparata and Hong presented their first  $O(n \log n)$  time algorithm in 3D. QuickHull [6] is also a fast technique that works in 2D and can be generalized to 3D. It takes as well  $O(n \log n)$  running time to compute CH. Moreover, Gift Wrapping is a 3D algorithm that constructs the convex envelope in  $O(nh)$  time. This output sensitive method proposed by Chand and Kapur was a generalization of Jarvis march and worked not only in 3D but also in arbitrary dimensions. More involved method in 3D was proposed by Chazelle and Matousek [7], they succeeded to accomplish an  $O(n \log h)$  running time algorithm. Edelsbrunner and Shi [8] made-up a deterministic technique having  $O(n \log^2 h)$  running time. The last two algorithms are not very practical and tend to be complicated thus; the problem of finding optimal and practical algorithms that construct convex envelope in 3D remained. Brute Force, Gift Wrapping and QuickHull algorithms were chosen to be tested, implemented and compared. They were also evaluated with respect to the proposed Hybrid method.

### 3 3D CONVEX HULL GENERATION

The convex hull or convex envelope of a finite set  $S$  of  $n$  points in the Euclidean space  $\mathbb{R}^d$  of dimension  $d$  denoted as  $CH(S)$  is defined by the smallest convex set containing all the points or simply the intersection of all half-spaces containing the set  $S$ . The convex hull in  $\mathbb{R}^d$  is the set of solutions to a finite system of linear inequalities in  $d$ -variables:

$$CH(S) = \{x \in \mathbb{R}^d : Ax \leq b\} \quad (1)$$

Where  $A \in \mathbb{R}^{n \times d}$  and  $b \in \mathbb{R}^n$ .

A solution of the above system can be written as the follow:

$$CH(S) = \sum_{i=1}^d \lambda_i p_i : \sum_{i=1}^d \lambda_i = 1, \lambda_i \geq 0 \quad (2)$$

Two algorithms for constructing the convex envelope in 3D are discussed and described in details. The first algorithm is the Gift Wrapping and the second one is the QuickHull. Then, a hybrid technique based on the last two algorithms is proposed.

#### 3.1 Gift Wrapping Algorithm

The Gift Wrapping algorithm known also as Jarvis March was created to work in arbitrary dimensions. It

consists of an initialization phase followed by a series of wrapping steps. The initialization phase begins first by finding a starting edge  $(a, b)$  by using the 2D algorithm on the projection of the points on the  $XY$  plane; it pivots an initial plane  $P$  around the edge  $(a, b)$  of the hull; it finds the smallest angle between the plane  $P$  containing the starting edge  $(a, b)$  and a plane  $T$  formed by point  $p_i$  and the edge  $(a, b)$ ; it replaces the point  $p_i$  by  $c$  and form a triangular face containing  $(a, b, c)$ . The plane  $(a, b, c)$  is a facet on the convex hull. All points now lie to the left of this plane. A set  $F$  of frontier edges is initially defined and contains the three edges  $(a, b)$ ,  $(a, c)$  and  $(b, c)$ . Each frontier edge in  $F$  is associated with a triangle or facet on the convex hull of  $S$ . The wrapping steps are repeated recursively for the edges  $(a, c)$  and  $(b, c)$  by finding other triangles adjacent to those edges. All steps for every explored edge are repeated until all facets have been explored. The Gift Wrapping algorithm needs  $O(nh)$  times operations to construct CH. It is clear that for every hull edge point discovered, the computer needs  $O(h)$  time where  $h$  is the number of hull points. Hence for  $n$  points in the set, the total time complexity is  $O(nh)$ . The worst case occurs when all of the input set of points occur on CH, the time complexity of the algorithm becomes  $O(n^2)$ .

#### 3.2 QuickHull Algorithm

The QuickHull algorithm finds the convex hull of  $n$  input points by recursively partitioning this given input set. It shares similarities with sorting algorithms such as it is recursive and each recursion step partitions the data sets into several subsets. QuickHull begins by dividing the set of points into two subsets with respect to a plane formed by: the vertices corresponding to the minimum ( $x_{min}$ ) and maximum ( $x_{max}$ ) coordinate, and the vertex corresponding to the maximum distance ( $x_d$ ) from the line joining  $(x_{min}, x_{max})$ . From this initial plane, QuickHull creates a polyhedral of new facets, called visible facets, by calculating the point that has the maximum distance ( $x_{dmax}$ ) with respect to the plane. Therefore, QuickHull builds new sets of points from the outside set of the located visible facets. If a point is above multiple new facets, one of the new corresponding facets is selected. If it is below all the new facets, the point is inside the convex hull and consequently it can be discarded.

Partitioning also records the furthest points of each outside set. Each point  $p$  in the outside set is processed to locate a visible facet. Visible facet means that the point  $p$  is above the specified facet. It constructs a polyhedral from the processed point  $p$  to the horizon edges of the visible facets. Finally it deletes this facet, therefore adds the newly created polyhedral of facets to the convex hull. The last three steps are repeated recursively for every point in the new outside set.

$O(n \log n)$  time operations are needed to compute the convex envelope using the QuickHull algorithm. The points will be partitioned into two equal sets and hence the depth of the recursion is  $(\log n)$ . At each level of recursion there are  $O(n)$  operations. Therefore, the overall average time is  $O(n \log n)$ .

#### 4 THE HYBRID ALGORITHM

Since the running time of CH algorithms depends on the number of points  $n$  that constitutes the object, many methods are used to speed up them by preprocessing the input points. Some techniques start by dividing the input points into two arbitrary sets, right and left, then computing the final convex hull. Divide-and-Conquer is one of those algorithms that starts recursively by computing the convex envelope of the right then the left set and finally merging the two hulls into a final convex output. Other techniques used to divide the input points into many subsets, Timothy Chan proposes the Chan algorithm [?] that starts by dividing the input points into  $(n/N)$  arbitrary disjoint subsets each of size  $N$ . Then, it computes the convex envelope of each group, to have an  $N$  partial hulls and then integrating the overall into a final output. The idea behind those techniques was always dividing the sets into many subsets trying to speed up the running time of the algorithm and to reduce its complexity. This paper proposes a hybrid technique that starts first by reducing the number of input set of points, then computing the corresponding convex envelope. It is based on QuickHull and Gift Wrapping algorithms. The wrapping step in the Gift Wrapping algorithm can be done faster if the set of input points is preprocessed. Therefore, preprocessing the input points by reducing their number will be a step forward to speed up the algorithm. The hybrid methods initiates by applying the QuickHull so that the input points are divided into two subsets (upper and lower) with an initial plane, then creates a polyhedron of new facets by calculating the point having the maximum distance with respect to this plane. Points that are inside the polyhedron are consequently inside the convex envelope and it is discarded. The same step is repeated for the lower set. This leads to the reduction of the number of input points constituting a new data set.

The new set is used as a new input for the Gift Wrapping algorithm. Consequently, wrapping steps were done by scanning the new data yielding into a final convex output. Furthermore, after preprocessing the input set of points and reducing its number, the hybrid method computes the facets of the hull one at a time, in counter clockwise order by the sequence of the wrapping steps. This algorithm is shown in figure 1.

#### 5 RESULTS AND SIMULATIONS

Table 1 contains the number of vertices and facets constituting the 3D original model of many wrist bones in

addition to vertices and facets constituting their corresponding convex hulls. It also shows the execution time of the conventional algorithms and the proposed Hybrid one. The Time shown in second is for computing the convex hull and drawing it with its corresponding model in the scene. Algorithms are tested on many bones that represent the 3D wrist model. One can remark that the new proposed hybrid algorithm is quicker than QuickHull and Gift Wrapping. Figure 2 shows the results of the Hamate and Ulna, two wrist bones.

Brute Forces takes long time to construct the convex envelope compared to other techniques. Therefore, this algorithm is not practically used specially in real time processing. On the other hand, Gift Wrapping and QuickHull are very fast in computing the CH for all the wrist bones. Moreover, the hybrid method outperforms the three conventional algorithms. Noticed that for the 3rdMetacarpal, the hybrid method decreases first the number of vertices from 675 to 559 then the wrapping process is used to construct the convex envelope. This yields the reduction of the running time to 0.12s compared to 0.26s for the Gift Wrapping and 0.21s for the QuickHull. The number of vertices in Scaphoid for example decreases from 2890 to 2539 yielding to decrease the running time from 1.31s for the Gift Wrapping and 1.22s for the QuickHull to 0.88s for the hybrid method.

For the 3rdMetacarpal, the proposed technique decreases the running time up to 53.87% compared to Gift Wrapping and 42.85% compared to QuickHull. Similarly, for the Scaphoid, the running time of the proposed algorithm decreases up to 32.82% compared to Gift Wrapping and 27.86% compared to QuickHull. The proposed technique performs very fast on small objects compared to big ones.

#### 6 CONCLUSIONS

Many 3D objects do not have a shape that could be modeled using precise mathematical equations, convex hull algorithms are a solution for these kinds of issues. The needs of convex hull algorithms rise with the intention of having realistic scenes with real-time interactive collision detection between objects in the virtual world. Since fast collision detection systems work almost exclusively with convex objects, quick convex hull algorithms are implemented in order to fulfill real-time requirements. In this paper, a Hybrid approach to construct the convex hull and speeding up its execution time is proposed and compared with three published methods: Brute Force, Gift Wrapping and QuickHull. The four techniques are implemented, discussed, tested and compared. The results are comparable in terms of execution time for each technique. 3D wrist and knee bones were shown with their corresponding convex envelopes. The proposed hybrid technique decreases the running time of the convex envelope computation for

```

1 find an initial plane from the min and max abscise and the max distance with respect to  $(x_{min}, x_{max})$ 
2 construct a polyhedron from the initial plane and the max distance to this plane
3 for each facet  $F$  of the polyhedron do
4   for each unassigned point  $p$  do
5     if  $p$  is above  $F$  then
6       assign  $p$  to  $F$ 's outside set
7     end if
8   end for
9 end for
10 Discard all points inside the polyhedron forming a new input set  $(n_{new})$ 
11 find a starting edge  $(a, b)$  using the 2D Gift Wrapping algorithm on the XY projection
12 for  $i=1$  to  $n_{new}$  do
13   find point  $p_i$  corresponding to  $min$  angle between plane  $P$  in XY containing  $(a, b)$  and plane  $T = (a, b, p_i)$ 
14   replace  $c \leftarrow p_i$ 
15   save  $(a, b, c)$  into  $Q$ 
16   wrap the edge  $(a, c)$ 
17   if facet has been explored then
18     wrap the edge  $(b, c)$ 
19     if facet has been explored then
20       return
21     end if
22   end if
23 end for

```

Figure 1: The Hybrid Approach

3D Model	Original Model		Convex Hull		Brute Force	Gift Wrap	QuickHull	Hybrid
	# vertices	# facets	# vertices	# facets	time (s)	time (s)	time (s)	time (s)
1 <sup>st</sup> Meta.	2179	4320	379	790	19014.22.1	0.66	0.61	0.43
2 <sup>nd</sup> Meta.	1168	2258	300	596	12015.41	0.47	0.38	0.25
3 <sup>rd</sup> Meta.	675	1272	150	296	2330.77	0.26	0.22	0.12
4 <sup>th</sup> Meta.	532	1002	147	290	2302.72	0.21	0.19	0.09
Hamate	2812	5620	394	784	19231.20	0.89	0.82	0.71
Ulna	977	1864	312	620	12153.03	0.41	0.37	0.22
Scaphoid	2890	5784	530	1056	22125.1	1.31	1.22	0.88
Capitate	3026	6048	635	1266	25300.35	1.46	1.38	1.25
Radius	2454	4754	288	572	11260.12	0.78	0.72	0.62

Table 1: Comparison of Execution Time For Computing The 3D Convex Hull

all objects in the scenes. The Hybrid method is an output sensitive algorithm that works in 2D as well as in 3D. It was shown that this method was very efficient, practical and usefulness in modeling 3D medical data and simulating their models in virtual environments.

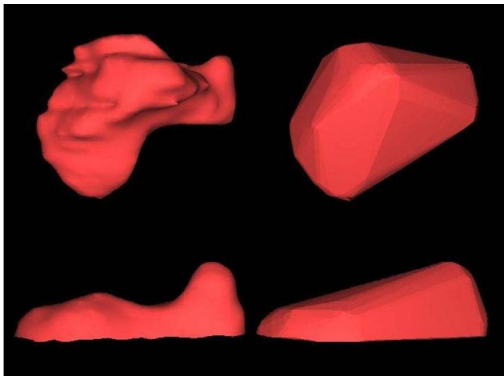


Figure 2: Hamate and Ulna with their convex hulls

## REFERENCES

- [1] S. Haque, S. Srinivasan, A Meta-Analysis of the Training Effectiveness of Virtual Reality Surgical Simulators, IEEE Transaction on IT in Biomedicine, (2006).
- [2] S. Akl, Efficient Convex Hull Algo. for Pattern Recognition, 4<sup>th</sup> Int. Conf. on Pattern Recognition, (1978).
- [3] V. Bayer, Survey of Algorithms for the Convex Hull Problem, Depart. of CS Oregon State Univ., (1999).
- [4] A. Day, Implementation of an Algo. to Find the Convex Hull of a Set of 3D Points, ACM (1990).
- [5] F. Preparata, S. Hong, Convex Hulls of Finite Sets of Points in Two and Three Dimensions, ACM Transactions on Mathematical Softwares, (1977).
- [6] C. Barber, D. Dobkin, H. Huhdanpaa, The QuickHull Algorithm for Convex Hulls, ACM Transactions on Mathematical Softwares, (1996).
- [7] B. Chazelle, J. Matousek, Derandomizing an Output-Sensitive Convex Hull Algorithm in 3D, Computational Geometry Theory and Applications, (1995).
- [8] H. Edelsbrunner, W. Shi, An  $O(n \log^2 n)$  Time Algorithm for the Three-Dimensional Convex Hull Problem, SIAM Journal on Computing, (1991).



# Implementing Interactive 3D Segmentation on CUDA Using Graph-Cuts and Watershed Transformation

Jan Kolomazník  
honza.kolomaznik@gmail.com

Václav Krajíček  
vajicek@cgg.mff.cuni.cz

Jan Horáček  
jan.horacek@gmail.com

Josef Pelikán  
pepca@cgg.mff.cuni.cz

Department of Software and Computer Science Education  
Faculty of Mathematics and Physics  
Charles University in Prague, Czech Republic

## ABSTRACT

In this paper we present a novel scheme for a very fast implementation of volumetric segmentation using graph cuts. The main benefit of this work is our approach to non-grid region adjacency processing on CUDA which to our knowledge has not been done yet in any efficient way. The watershed transform radically reduces the number of vertices for graph processing. Everything starting from watershed transformation and ending with graph cut was parallelized and is performed directly on the GPU.

**Keywords:** 3D segmentation, CUDA, watersheds, min-cut, push-relabel, max flow.

## 1 INTRODUCTION

Segmentation of 3D volumetric images brings a variety of problems. First of all, we are processing a large amount of data, rendering robust and precise algorithms prohibitively expensive.

In the case of 2D, the user can resort to manual segmentation if needed, while in 3D, manual segmentation becomes a very tedious task. To address this issue, an interactive (semi-automatic) method that would assist the user during the segmentation process is needed. In this paper, we propose an algorithm that segments the data based on initial user input (Figure 1), allowing the user to correct and improve the resulting segmentation by providing the algorithm with further hints.

We have chosen a segmentation approach based on the minimal graph cut algorithm. This is a well known image processing algorithm, however its computational complexity prevents it, at least without further improvements, from being practically useful for any kind of three dimensional data (CT, MRI). To alleviate these issues, we first transform the input data using the watershed transformation to an induced minor of the original graph, significantly decreasing the vertex count. We then process this data using the massively parallel architecture of programmable GPUs.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

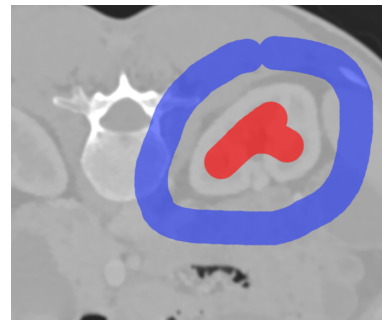


Figure 1: User input – CT image

## 2 RELATED WORK

Same overall (min-cut on watersheds) methods were chosen by authors in [8] for the interactive segmentation of liver and liver tumours, but they processed much smaller data in a longer time.

Several approaches can be used to compute the watershed transformation (there are also several equivalent definitions) [7] on CPU is a typical method based on flooding. The authors in [4] used Bellman-Ford shortest distance algorithm and cellular automata formulation on parallel architectures.

The min-cut problem is connected to finding the maximal flow through net. We needed an easily parallelizable algorithm. We therefore considered the push-relabel algorithm [1]. There are many papers on various implementations of the push-relabel algorithm. Most of them are focused on segmentation of 2D images, where the input graph is built as a neighborhood representation of image pixels – a grid where each vertex (except on borderlines) has a degree of 4 or 8 depending on pixel connectivity. Several CUDA implementations exist [3]

and all of them take the pixel/voxel based adjacency graph as their input, so they can easily optimize memory access of CUDA kernels. One of our goals is to implement on CUDA graph-cuts for general graphs.

### 3 OUR WORK

The reference single threaded CPU versions of the mentioned algorithms take minutes or at least tens of seconds to provide results on our data.

Because a CPU-only implementation of the algorithm is not feasible, a design decision has been made to make use of the computational power of the programmable GPUs. Furthermore, we observe that a lot of time can be wasted by copying data between the host computer's main memory and the GPU's memory, making a GPU-only, in our case CUDA based, implementation the approach of choice. While some of the algorithms, e.g. image filtering, are naturally parallelizable and therefore suited for parallel processing, others must be completely redesigned, ideally into the form of cellular automata, which can be easily executed in CUDA.

#### 3.1 Method Overview

Overview of method is listed in Algorithm 1.

---

##### Algorithm 1 Method overview

---

```

Denoising
Compute gradient magnitude of the image
Watershed transformation (Section 3.3)
Build region adjacency graph  $G$  (Section 3.4)
repeat
  Get user input
  Modify  $G$  according to user input
  Compute max-flow (Section 3.5)
  Find min-cut (Section 3.6)
  Present result to the user
until user satisfied

```

---

#### 3.2 CUDA

Current hardware for CUDA implements a massively parallel architecture with the ability to run certain algorithms much faster (even by one or two orders of magnitude) than common CPU. However, such power is not for free and the above-mentioned algorithms must fulfill many conditions to enable the hardware arithmetic units to perform with top performance. Care must be taken in the areas of memory access, conditional jumps and many others. See [6] for details. We addressed these problems in our implementation.

We use the newest CUDA framework's features (available in version 4.1) such as atomic operations, C++ templates and Thrust (*STL* like library of data-structures and algorithms executed on CUDA, see [5]). Templates are mostly used to allow process various

input data formats, but we also pass all limits known during compilation as template parameters, so compiler can easily optimize cycles and local variables. The *Thrust* library is used to ease up data transfer between host and device code. We also use algorithms like *thrust::inclusive\_scan* and *thrust::reduce*.

We developed a generic memory loading scheme for kernels executed on image data. Most of the memory reads are coalesced and warp divergence is prevented as much as possible. We load cube of 10x10x10 voxels into shared memory (internal 8x8x8 cube allows coalesced loading) for kernels working on 3x3x3 neighborhood of voxel.

#### 3.3 Watershed Transformation

Most of the watershed transformation algorithms are serial in nature because of the internal usage of a priority queue or a similar structure:

1. Initialize a set of markers - image local minima. Each labeled by different ID.
2. Insert the neighbors of the marked regions into a priority queue (priority is the gray level of the image element).
3. Pop an element from the queue. If all its neighbors have the same label, label the element with same ID. Insert the neighbors that are not yet in the priority queue.
4. While the queue is not empty redo step 3.

Several parallelization approaches for the serial versions are available (e.g. [7]), but none of them is suitable for GPUs. We decided to use a cellular automaton reformulation of the problem by [4].

First of all, we need to find the markers for the watershed transformation (local minima of gradient magnitude image). This is a fairly straightforward processing of each voxel in a 3x3x3 neighborhood. As a result we obtain an image with nonzero-labeled regions on zeroed background (each element with a different label). So as the next step we need to mark all compact regions with single label each. This is done by *connected component labeling (CCL)* [2].

We implement CCL as an iterative process which finds region equivalences (Algorithm 2). Two regions are considered equivalent (and should have the same label) when they are neighbors. So we are marking these equivalences in a lookup table and relabel the whole image in every iteration. At the end we do the final relabeling so that we produce a continuous sequence of labels.

Finally, in the cellular automata formulation of the watershed transformation, we try to optimize the distance of each voxel to the closest marker. All computations are based on local information only (in our case, 3<sup>3</sup> voxels) so it is tailored for CUDA (Algorithm 3).



---

**Algorithm 2** Connected component labeling

---

```

Allocate ID equivalence lookupTable
Init lookupTable
Scan labelBuffer for equivalences (element neighbors with different nonzero label)
while lookupTable updated do
    Update lookupTable
    Relabel elements in labelBuffer by equivalences from lookupTable
    Scan labelBuffer for equivalences
end while

```

---



---

**Algorithm 3** Watershed transformation

---

```

Initialize distance buffer (set local minima markers to zero and rest to infinity)
while distance buffer updated do
    for all voxels do
        Test neighbors for shorter path
    end for
end while

```

---

### 3.4 Construction of Adjacency Graph

The construction of the adjacency graph in CUDA is not straightforward, as dynamic data structures are difficult to implement effectively and using an adjacency matrix or similar datastructure is memory consuming.

We will construct a list of all edges and their weights. All algorithms are implemented for an undirected graph so each edge will be present in the list only once.

As a data structure we decided to implement a hash table using *open adress linear hashing*, where the edges are stored as a special 64-bit index (32-bit for each vertex). This way was chosen so we can atomically insert edge records into the hash table.

Because we detect edges multiple times (for every voxel on the common border), we have to check if an edge is already inserted into the table. We either accumulate information about the image gradient, or we insert a new edge record.

A typical situation is that we want to insert the same edge multiple times at one moment (threads in a block run locally) – so to prevent serialization of accesses in global memory, we have to use a two-level approach. The hash table is not created only in global memory, but also in shared memory for every block. Concurrent insertions are first handled in the context of one block and the accumulated data from these tables is then inserted into the global hash table at the end of kernel execution.

The hash table is then sorted so all records are in the beginning of the array. Because of the design of our graph algorithms we have not to create an adjacency list or other representation of the vertex neighborhood. All algorithms need only an edge soup and the vertex count.

### 3.5 Push-Relabel Algorithm

The basic version of the push-relabel algorithm [1] consists of the operations *push* and *relabel*, which are applied as long as the corresponding conditions are met (the algorithm ends, when neither of those operations can be used).

Each vertex has a *height* (label) and *excess* assigned (flow yet to be distributed to its neighbors).

During the computation of the flow net, our algorithm constructs the so called pre-flow – defined as the flow, where some vertices are assigned some excess flow. When the algorithm converges the only vertices with a nonzero excess are source and sink.

Operation *push* tries to decrease excess of some vertex  $u$  by sending maximal possible flow through unsaturated incident edges. Can be applied only when:

- $excess(u) > 0$
- $capacity(u, v) - flow(u, v) > 0$
- $h(u) > h(v)$

Operation *relabel* handles situations when some vertex has nonzero excess and cannot apply push:

- $excess(u) > 0$
- $h(u) \leq h(v), \forall v, c(u, v) - f(u, v) > 0$

If these conditions hold we assign minimal label to the processed vertex so it is higher than at least one its neighbors via unsaturated edge.

Both operations work on vertices and their neighbors. This is problematic for the CUDA implementation targeted on general graphs. CUDA kernel calls need to be as coherent as possible (same instructions, aligned memory accesses) to achieve maximal throughput. This is the reason why other CUDA graph-cut implementations (e.g. [3]) work only on grid graphs, where every vertex has the same number of neighbors (depending on connectivity).

We solved this issue by formulating *push* and *relabel* operations not over vertices but over edges. We process all edges and accumulate data for vertex update after the edge processing is finished. Atomic operations are used to update the vertex properties (excess, label) directly when it is needed.

CUDA implementation of *push* is quite straightforward (Algorithm 4), except we iterate over edges not over vertices (we do not have information about vertex neighbors).

*Relabel* must be implemented in three phases:

1. locating vertices with nonzero excess (parallel predicate check)
2. checking all edges to find new label to vertices marked in step 1 (Algorithm 5)
3. assigning new label to marked vertices

---

**Algorithm 4** Parallel push

---

```

for all  $Edge \in E$  do
  if  $Label[Edge.v1] > Label[Edge.v2]$  then
     $pushFromTo(Edge.v1, Edge.v2, Edge)$ 
  else if  $Label[Edge.v2] > Label[Edge.v1]$  then
     $pushFromTo(Edge.v2, Edge.v1, Edge)$ 
  end if
end for

```

---



---

**Algorithm 5** Parallel relabel – phase 2

---

```

for all  $Edge \in E$  do
  if  $Edge.v1$  enabled and relabel conditions hold then
    Use atomic min (so more suitable value cannot
    be overridden) to assign new possible height for
     $Edge.v1$  in buffer
  else
    disable  $Edge.v1$ 
  end if
  ... symmetrically for opposite orientation
end for

```

---



---

**Algorithm 6** Parallel BFS

---

```

Allocate CUDA arrays  $F1, F2$  and  $W$  (visited)
 $fill(F1, false)$ 
 $F1[start] = true; W[start] = true$ 
repeat
   $fill(F2, false)$ 
   $bfsKernel(E, V, F1, F2, W)$  (Algorithm 7)
   $markFrontierAsVisited(F2, W)$ 
   $swap(F1, F2)$ 
until no vertex added to frontier

```

---



---

**Algorithm 7** BFS kernel

---

```

for all  $Edge \in E$  do
  if  $Edge.v1 \in F1$  and  $Edge.v2 \notin W$  then
     $F2[Edge.v2] = true$ 
    mark frontier as not empty
  end if
  ... symmetrically for opposite orientation
end for

```

---

### 3.6 Consolidation of Results

Min-cut can be easily obtained from the flow net. Simply execute the breadth first search (Algorithm 6) either from the source or the sink and stop on saturated edges – these edges are our min-cut. All vertices visited during BFS form the first set of regions (either background or the segmented object) while the rest is the second set.

At the end we usually apply some morphological corrections (morphological opening), because the result is an union of several watershed regions, which have a typical property – jagged contours.

Algorithm	100 <sup>3</sup>	200 <sup>3</sup>	512x512x256
Markers	0.006s	0.035s	0.794s
Watersheds	0.049s	0.121s	2.648s
Min-cut	0.281s	0.436s	2.381s

Table 1: Speed of presented algorithms

## 4 RESULTS

We tested our implementation on NVIDIA Fermi GPU (GTX 560). We can compare our results (Table 1) with published results ([8], [4]). Our solution typically achieves a throughput increase by factor of 10x-20x.

In comparison to the CPU implementation, our CUDA algorithms demonstrate a 50x-70x speedup.

## 5 CONCLUSION

We have shown an effective implementation of a basic segmentation method based on the watershed transformation and a push-relabel max-flow algorithm.

The problem with large number of image voxels was partially solved by working on regions instead on image elements and to decrease the computation time even more we successfully used CUDA enabled GPUs.

The used algorithms were modified for massively parallel architectures. We removed the problematic push-relabel algorithm dependence on graph topology (limitation of the other implementations). Our modifications also made the requirement of vertex neighborhood in graph representation redundant.

## ACKNOWLEDGEMENTS

This work was supported by the Grant Agency of Charles University, Prague (project number 355311).

## REFERENCES

- [1] A V Goldberg and R E Tarjan. A new approach to the maximum flow problem. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, STOC '86, pages 136–146, New York, NY, USA, 1986. ACM.
- [2] K. A. Hawick, A. Leist, and D. P. Playne. Parallel graph component labelling with gpus and cuda. *Parallel Comput.*, 36:655–678, 2010.
- [3] Mohamed Hussein, Amitabh Varshney, and Larry Davis. On implementing graph cuts on cuda. *First Workshop on General Purpose Processing on Graphics Processing Units*, 2007.
- [4] C. Kauffmann and N. Piche. Cellular automaton for ultra-fast watershed transform on gpu. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4, 2008.
- [5] NVIDIA Corporation. CUDA Toolkit 4.0 Thrust Quick Start Guide, 2011.
- [6] NVIDIA Corporation. NVIDIA CUDA C programming guide, 2011. Version 4.1.
- [7] Jos B. T. M. Roerdink and Arnold Meijster. The watershed transform: definitions, algorithms and parallelization strategies. *Fundamenta Informaticae - Special issue on mathematical morphology*, 41(1-2):187–228, 2000.
- [8] Jean Stawiaski and Etienne Decenci. Interactive liver tumor segmentation using graph-cuts and watershed. *Liver Tumor Segmentation - MIDAS - Grand Challenge*, 2008.

# Visual Acuity and Comfortable Distance from a Display

Michal Seeman

Brno University of Technology  
Faculty of Information Technology  
Božetěchova 2  
612 66, Brno, Czech Republic  
seeman@fit.vutbr.cz

Pavel Zemčík

Brno University of Technology  
Faculty of Information Technology  
Božetěchova 2  
612 66, Brno, Czech Republic  
zemcik@fit.vutbr.cz

## ABSTRACT

The image perception in humans depends on many factors. One of the most important ones is certainly the observing distance. The distance from which human look at the images is important for studying image perception in general and in particular on display devices. This paper describes a method of measurement of the preferred observing distance among humans of different age, sex, etc. under controlled conditions.

## Keywords

visual acuity, display, perception

## INTRODUCTION

Presentation of images on display devices has been historically examined from many points of view. They include e.g. signal processing and image resampling methods, perception scale of luminance [2], colorimetry, psychophysical models of human visual system (HVS) [1], or physiology inspired tone-mapping operators for high dynamic range (HDR) images [3, 4]. Although the spatial response of HVS has been examined as well [5, 6], the data obtained through measurement of the spatial response of HVS is not commonly used for geometry perception optimization. Exploitation of the HVS space attributes in the display devices introduces one more problem. In general, the observing distance is not known. This paper presents an approach usable for measurement of the observer's preferred distance from a display device and measurement of the visual acuity under the same conditions based on the statistics obtained in several users.

## Aim of The Measurement

Users tend to view the display from the so-called “comfortable distance”; however, such term is very vague, so it needs to be narrowed in order to be more

specific. The aim is to find an ideal observation distance which is the best one for examination of details on a still photograph rendered via the display device. As the “examination of details” is still not specific enough and can even be different for different users, the “examination action” can be made more specific through preparation of a specific task the users should perform so that the image perception in users can be compared and the achieved quality of image perception measured. The task for the users is evaluation of different filters applied on a series of images. Each picture is processed by several filters and the users are asked to subjectively choose which version they like, while in fact, the result is not that important. This leads into spontaneous localization of the users in the optimal observation distance suitable for comparison of the image details.

The question is how does the comfortable observation distance correspond to the visual acuity. Unlike the standard visual acuity measurement, the table of patterns (optotypes) is placed at the chosen distance, onto the display surface. The real visual acuity value is, therefore, relative to the chosen distance, which is measured as well. This approach ensures that the conditions, mainly the focal plane of the eyes, are close to those that apply during observation of the display. In some subjects, the different distances would also cause a need to change glasses which also means that the results in the subject with and without the glasses might not correspond.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

## The Experiment

Each of the subjects is seated in front of a display on a movable chair and instructed to move freely and position itself into a comfortable position. Several images are then shown in a sequence. Each of the images is filtered by four slightly different hi-pass filters. All four variants of the same image are then rendered on the screen side by side. The order of the filters is always random. The subject is asked to rate all of the four variants by selecting “like” “dislike” or “neutral” icon. The testing screen is shown in Figure 1. The differences between the digital filters are very tiny; therefore, the observer spontaneously chooses the best conditions for careful and detailed examination of the images. After the image rating is finished, they are asked not to move. The screen is then turned white and overlplaced by the visual acuity measurement chart (see Figure 2). Subjects are asked to select the finest pattern of horizontal stripes they can resolve. The distance eye to display is measured during the test without disturbing the subjects (using triangulation). The optional reading aids are used throughout the test.

In the first test, many subjects reported difficulties in optotypes resolving, because with some of the densities, the pattern was visible at the field border only. To prevent the inaccuracy caused by this fact, another set of patterns were prepared with faded borders (Figure 3) and the subjects acuity was re-evaluated after the tests were repeated.

### Important data

<b>Display model</b>	HP LP2465
White absolute luminance	60.8 cd·m <sup>-2</sup>
Black absolute luminance	0.245 cd·m <sup>-2</sup>
Luminance range	1:248
Pixel spacing	0.270mm
Display was calibrated with sensor (Datacolor Spyder 3 Pro)	
Gamma correction	2.2
White balance	6500K
<b>Diffuser underneath the optotypes</b>	
Thickness	3mm
Luminance loss	0.673 (-1.7dB)

### Optotypes

Seven striped optotypes optically transferred to inverse film (Fuji Provia 100F RDP3)

Minimal density	3.35 cycles·mm <sup>-1</sup>
Maximal density	4.78 cycles·mm <sup>-1</sup>
White absolute luminance	15.1 cd·m <sup>-2</sup>
Black absolute luminance	0.202 cd·m <sup>-2</sup>
Contrast	1:74

### Testing room

Maximal illuminance measured at the desk	65 lx
--	-------

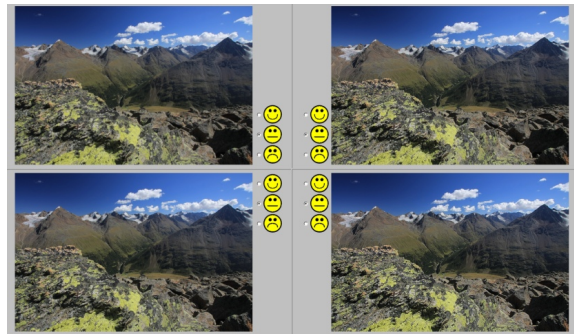


Figure 1: Testing screen

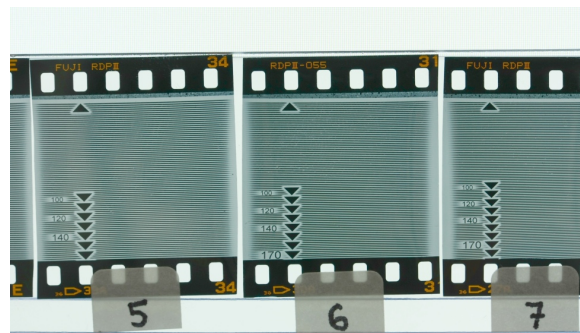


Figure 2: Optotypes



Figure 3: Optotype microscopy scan



Figure 4: Optotype marks for density measuring

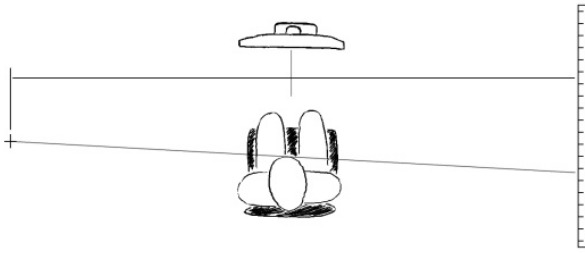


Figure 5: Distance triangulation

### Measuring Methods

The distance from eye to the display is read from a marked point on a scale placed far on the side, parallel to the display axis (see Figure 5). The parallax error is compensated by proper scaling of the meter.

The optotypes with exact density could not be made without specialized optics. The optotype set was made with approximate scaling. Control marks were printed among the stripes (see Figure 4). For each of the optotypes the distance between the marks was measured with precise caliper and the density was calculated from the scale factor. Therefore the optotype set does not follow linear nor logarithmic series.

Angular acuity is computed from the relative acuity and eye-to-optotype distance as follows:

$$A_a = 10 \cdot A_r \cdot d \cdot \frac{\pi}{180}$$

where  $A_a$  is the angular acuity in cycles per degree,  $A_r$  is the relative acuity in cycles per millimeter and  $d$  is the distance in centimeters. The 3mm thickness of the diffusing plate underneath the optotypes was omitted in this calculation because of the distance precision issue mentioned above.

The credibility of the result in each subject should be checked by the preferred filter variant. The order of the images was random, so if the subject shows no clear preference, the differences among the filters are likely to be unrecognizable and the subject should be excluded from the statistics.

### Conclusions

We propose an approach to measure relationship between visual acuity in humans and a distance selected by each individual user to examine details on a display device. The results of this measurement are important in application of the visual system parameters to the image processing for display devices. It follows that the relative spatial acuity could be possibly applied with the HVS spatial response. It is also possible to design a new, more

accurate model of HVS linked to the display device parameters.

In future works, video represents another field of investigation. The preferred observing distance could be possibly very different with moving objects, not mentioning that HVS behaves differently than with still images.

### References

- [1] Daly, S.: The Visible Differences Predictor: an Algorithm for the Assessment of Image Fidelity, *Digital Images and Human Vision*, pp. 179-206, MIT Press 1993
- [2] Purpura, K., Kaplan, E., Shapley, R.M.: Background light and the contrast gain of primate P and M retinal ganglion cells, *Proc. Natl. Acad. Sci.*, Vol. 85, pp. 4534-4537, Jun 1988
- [3] Durand, F. Dorsey, J.: Fast bilateral filtering for the display of high-dynamic range images, *Proc. Conf. on Computer Graphics and Interactive Techniques*, pp. 257-266, 2002, ISSN:0730-0301
- [4] Ledda, P., Santos, L.P., Chalmers, A.: A local model of eye adaptation for high dynamic range images, *Proc. Int. Conf. on Computer Graphics, Virtual Reality, Visualisation and Interaction, AFRIGRAPH*, 2004
- [5] Van Nes F.L., Bouman, M.A.: Spatial Modulation Transfer in the Human Eye, *J. Opt. J. Optical Society of America*, Vol. 57, Issue 3, pp. 401-406, 1967
- [6] Croner, L.J., Kaplan, E.: Receptive Fields of P and M Ganglion Cells Across the Primate Retina, *Vision Res.* Vol. 35, No. 1, pp. 7-24, 1995, pp. 151-160, 2004, ISBN:1-58113-863-6.



# POSTER: Geodesic Stripes Based Hierarchical Evaluation for 3D Facial Similarity

Hongyan Li  
Beijing Normal University  
100875, Beijing, China  
lihongyan@mail.bnu.edu.cn

Zhongke Wu  
Beijing Normal University  
100875, Beijing, China  
zwu@bnu.edu.cn

Donghua Huang  
Beijing Normal University  
100875, Beijing, China  
donghua0928@163.com

## ABSTRACT

Similarity evaluation of 3D face is the core issue in 3D face recognition. The article puts forwards a geodesic stripes based evaluation method which realizes assessment from global face to local parts. It can be used to make effective and comprehensive evaluation in many fields such as 3D face reconstruction, forensic science, archaeology etc. First, simplify each 3D face with a series of geodesic stripes and calculate distribution vector between each pair of stripes, which reflects 3D space distribution feature. Then through feature extraction on entire face, we get a distribution matrix which consists of all distribution vectors. The similarity between two distribution matrices directly shows the global similarity between two faces. We also extract geodesic stripes feature on local organs like mouth eye and nose to make a more accurate evaluation. The experimental results on SHREC2008 3D face database further testify that the hierarchical evaluation method is available and consistent with the subjective evaluation.

## Keywords

Posture standardization, ICP (Iterative Closest Point), Geodesic stripes, Similarity calculation

## 1. INTRODUCTION

In the information age, there have been many researches on identity authentication with biological characteristics like face, fingerprints, iris, etc. The common 3D facial similarity evaluation is achieved by artificial and subjective means. This method is mainly to gather a certain amount of volunteers, and design different strategies to evaluate similarity by subjective visual [Ste06a]. Although the method is consistent with the human cognitive principles, it's time-consuming and labor-intensive [Qua07a]. In this article, a method is proposed from the geometric and numerical perspective to evaluate the similarity.

Similarity evaluation methods of general 3D model can be divided into geometric feature-based similarity evaluation and topological feature-based similarity evaluation. In addition, from early 90 's of last century, similarity evaluation is widely used in some professional 3D models like Biological molecules [Can03a][Rog03a], terrain stripe and mechanical elements [Kri03a]. Generally, besides using geometric feature and topological feature mentioned above, similarity evaluation of professional 3D models also use specific domain knowledge.

Similarity evaluation of 3D face is the core issue in 3D face recognition. At present, feature point based method is widely used in morphometry [Ric02a]. Feature points based method needs to precisely extract feature points like tip of the nose, mouth corner and other parts, then construct feature vector

to measure the similarity between different faces [Tho05a]. However, it's hardly to extract feature point in smooth face area like cheek and forehead. Therefore this is a great challenging problem. Gupta proposes an efficient method of extracting feature points to overcome the difficulty [Gup07a], which uses these feature points defined by Farkas [Far87a]. Face curve based method uses meaningful contour curve to present the face model [Mip07a][Sam06a][Ter08a]. The contour curve actually consists of discrete points which have equal distance to nose tip. Then the curve characteristics are compared to calculate face similarity [Ter08b][Fen07a]. Here we apply a contour stripe based method to make an evaluation from global face to local organs.

## 2. FEATURE EXTRACTION

A stripe is a set of vertexes which share the same distance to the nose tip. Hence, the relationship between two stripes can be described by the space distribution feature between two sets of vertexes through statistical analysis. In the article, to calculate space distribution feature, we refer the 3DWW (3D Weighted walkthroughs) [Ber03a] between two entities in space. Meanwhile, Berretti also applied iso-Geodesic stripes to extract feature of 3D face. Compared with Berretti's 2010, we make two major improvements on the way of feature extraction and the way of similarity calculation. The following passage will illustrate the space distribution feature between two stripes.



### Definition of space distribution feature

Each stripe consists of a group of discrete vertexes, and the group of vertexes can approximately represent a part of surface. Here one improvement is use the discrete statistic to replace the numerical differentiation on face surface [Ber10b]. In this case, the computation efficiency is greatly increased, while the computation result is almost same. According to the different position in 3D space, each pair of points has different space distribution feature  $\omega(i,j,k)$ . as in (1-2)

$$\omega(i, j, k) = \omega(F(a_x - b_x), F(a_y - b_y), F(a_z - b_z)) \quad (1)$$

$$F(x) = \begin{cases} 1 & x > \delta \\ 0 & |x| \leq \delta \\ -1 & x < -\delta \end{cases} \quad (2)$$

Here  $\delta$  is a threshold, and the distribution feature  $\omega(i,j,k)$  has 27 kinds in total. There are two sets of vertices. Any vertex in set A and any vertex in set B can make up a pair of vertices, which corresponds to a distribution feature. According to the combination principle, the number of distribution feature between two sets of vertices is . Then calculate distribution feature, as in (3-14).

$$\omega H = \omega_{1,1,1} + \omega_{1,-1,1} + \omega_{1,1,-1} + \omega_{1,-1,-1} \quad (3)$$

$$\omega V = \omega_{-1,1,1} + \omega_{1,1,1} + \omega_{-1,1,-1} + \omega_{1,1,-1} \quad (4)$$

$$\omega D = \omega_{1,1,1} + \omega_{1,-1,1} + \omega_{-1,1,1} + \omega_{-1,-1,1} \quad (5)$$

$$\omega X Y = \omega_{-1,-1,1} + \omega_{1,1,1} + \omega_{1,1,-1} + \omega_{-1,-1,-1} \quad (6)$$

$$\omega X Z = \omega_{-1,-1,-1} + \omega_{1,-1,1} + \omega_{1,1,1} + \omega_{-1,1,-1} \quad (7)$$

$$\omega Y Z = \omega_{-1,1,1} + \omega_{1,1,1} + \omega_{1,-1,-1} + \omega_{-1,-1,-1} \quad (8)$$

$$\omega H_0 = \omega_{0,1,1} + \omega_{0,-1,1} + \omega_{0,1,-1} + \omega_{0,-1,-1} \quad (9)$$

$$\omega V_0 = \omega_{-1,0,1} + \omega_{1,0,1} + \omega_{-1,0,-1} + \omega_{1,0,-1} \quad (10)$$

$$\omega D_0 = \omega_{1,1,0} + \omega_{1,-1,0} + \omega_{-1,1,0} + \omega_{-1,-1,0} \quad (11)$$

$$\omega H V_0 = \omega_{0,0,1} + \omega_{0,0,-1} \quad (12)$$

$$\omega H D_0 = \omega_{0,1,0} + \omega_{0,-1,0} \quad (13)$$

$$\omega V D_0 = \omega_{1,0,0} + \omega_{-1,0,0} \quad (14)$$

The value of  $\omega(i,j,k)$  is the distribution probability in all  $m*n$  distribution features. The specific calculation method refers to the following equation.

$$\omega_{i,j,k}(A, B) = \frac{\text{num}(\{(a, b)\})}{m * n} \quad (15)$$

$$F(a_x - b_x) = i \ \& \ F(a_y - b_y) = j \ \& \ F(a_z - b_z) = k \quad (16)$$

The num ( $\{(a, b)\}$ ) is the number of the pair of vertexes (a, b) in the set. At last, the space distribution feature between two sets of A and B  $W(A, B)$  can be described by a vector  $(\omega H, \omega V, \omega D, \dots, \omega H V_0, \omega H D_0, \omega V D_0) T$ , whose dimension is 12.

### Geodesic stripes extraction

With the evolvement and optimization of the geodesic distance algorithm, there are many applications in the research area of mesh segmentation and mesh smoothing [Pey04a]. The research has testified that geodesic stripes perform well when used in 3D face recognition [Sam06a]. In order to calculate the geodesic distance from nose tip to every vertex on the 3D model, we apply the fast exact geodesic method which is demonstrated in paper [Vit05a]. Figure 1 is the face which is represented with the geodesic stripes.

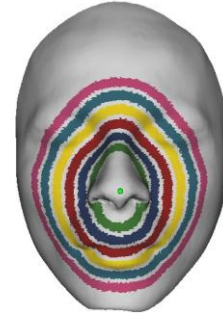


Figure 1. Face simplified with geodesic stripes

### 3. SIMILARITY CALCULATION

Based on the feature extraction in the above passage, the similarity between two faces can be converted into the similarity between distribution vectors. As for the distribution vectors, the article [Ber10b] calculates distribution vectors separately and make weighted sum of them. However, from a different perspective, we treat the 3D face as a whole entity and arrange all the distribution vectors into a distribution matrix, whose size is  $K * (K-1) / 2$  columns and 12 rows. Therefore, the similarity between two distribution matrices can directly reflect the similarity between two faces. In the next, the feature of two face models can be expressed by matrix  $M_1$  and  $M_0$ , the similarity is calculated, as in (17) and (18).

$$S(M_1, M_0) = R(M_1 - M_{\min}, M_0 - M_{\min}) \quad (17)$$

$$M_{\min}(i, j) = \min\{M_0(i, j), M_1(i, j), \dots, M_K(i, j)\} \quad (18)$$

$R(M_1, M_0)$  is 2D correlation coefficient which is often used to measure deformation (engineering), displacement, and it is also widely applied in many areas of science and engineering [Sut09a]. The value of correlation coefficients fall in the range between 0



and 1, where 1 indicates perfect similarity between two matrices, and 0 indicates no similarity. Any value in between is a measure of the extent of similarity [Guf06a].

#### 4. FEATURE EXTRACTION ON LOCAL ORGANS

In fact, it's necessary and meaningful to evaluate facial similarity on local organs (mouth, eye, nose). To make a more accurate and comprehensive assessment. Here we can segment local organs according to the geodesic distances from a reference point. As shown in Figure2, local organs represented with geodesic stripes.

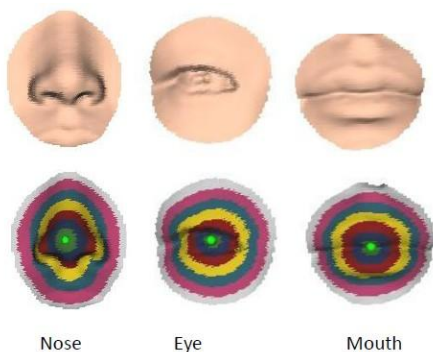


Figure 2. Local organs with geodesic stripes

#### 5. EXPERIMENTAL RESULTS

##### Experiments on SHREC2008 database

In order to check the effect of the method based on geodesic stripes, we test the algorithm with the database SHREC 2008 (3D Shape Retrieval Contest 2008). The experimental results are shown in Figure 3.

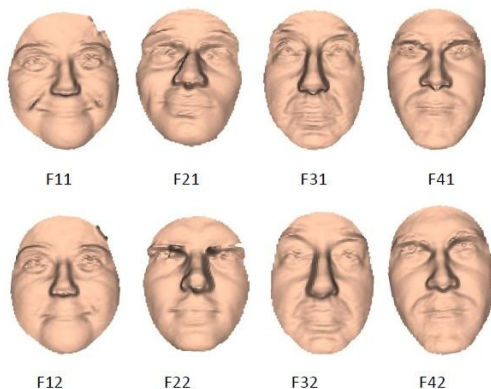


Figure 3. Face recognition results of 4 individuals

The first row (F1, F2, F3, F4) represents the inputs, and the second row represents the recognition results. Meanwhile, the table 1 is the similarity results between different individuals. It is not difficult to find that the similarity result of the same individual is big,

and the similarity result among different individuals is relatively small.

Global Face	Similarity Results			
	<i>F12</i>	<i>F22</i>	<i>F32</i>	<i>F42</i>
F11	<b>0.9736</b>	0.7386	0.7126	0.8548
F21	0.7420	<b>0.9386</b>	0.9098	0.8285
F31	0.8359	0.8797	<b>0.9907</b>	0.8688
F41	0.8933	0.8640	0.9003	<b>0.9809</b>

Table 1. Recognition results on SHREC2008

##### Experiments on evaluation of 3D facial reconstruction

In experiment we apply the previous the method to evaluate the similarity both on global face and local organs to further prove the effect of the hierarchical evaluation. Table2 is the similarity result to compare the quality of two reconstruction methods. They are the quantitative similarity results for global face, nose, eye and mouth. The experimental results show that F3 is more similar to F1 than F2 to F1, which also means the regional method perform better. The experimental results show that the similarity of organs contributes to the global similarity.

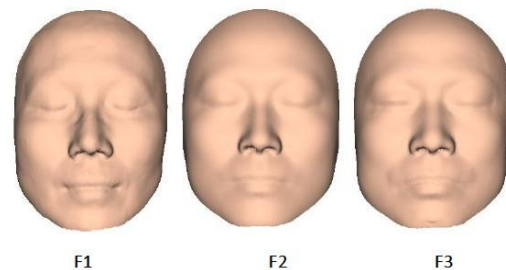


Figure 4. Reconstructed results for original face F1, F2 is reconstructed faces with global method and F3 is with the regional method

Face	Similarity Results			
	<i>Global</i>	<i>eye</i>	<i>nose</i>	<i>mouth</i>
F1:F2	0.8911	0.8542	0.8734	0.8132
F1:F3	0.9263	0.8701	0.8921	0.8347

Table 2. Evaluation of face reconstruction method

#### 6. REFERENCES

- [Ste06a] Stephan CN, Arthur RS. Assessing facial approximation accuracy: how do resemblance ratings of disparate faces compare to recognition tests. *Forensic Sci Int* 2006; 159S:S159–63.
- [Qau07a] Quatrehomme, G érald. Evaluation of the accuracy of three-dimensional manual face reconstruction: a series of 25 controlled cases. *International Journal of Legal Medicine*.2007. SN 0937-9827.

- [Can03a] T. Can, Y. F. Wang. CTSS: A Robust and Efficient Method for Protein Structure Alignment Based on Local Geometrical and Biological Distributions. In: Proceedings of the Second International IEEE Computer Society Computational Systems Bioinformatics Conference, Stanford, California, August 2003, 169~179.
- [Rog03a] P. Rogen, B. Fain. Automatic classification of protein structure by using Gauss integrals. PNAS. Jan. 2003, 100 (1): 119~124.
- [Kri03a] H.P. Kriegel, B. Stefan, P. Krbger, M. Pfeifle, M. Schubert. Using sets of distribution vectors for similarity search on voxelized CAD objects. In: Proceedings of the 2003 ACM SIGMOD international conference on Management of data, 2003, 587~598.
- [Ric02a] J.T. Richtsmeier, V.B. DeLeon, The promise of geometric morphometrics, *Am. J. Phys. Anthropol.* 35 (2002) 63~91.
- [Tho05a] C.D.L. Thomas, Three-dimensional quantification of facial shape, in: J.G. Clement, M.K. Marks (Eds.), *Computer-Graphic Facial Reconstruction*, Elsevier Academic Press, London, 2005.
- [Gup07a] S. Gupta, J.K. Aggarwal, M.K. Markey, A.C. Bovik, 3D face recognition founded on the structural diversity of human faces, in: *CVPR 2007*, 2007, 643~649.
- [Far87a] L.G. Farkas and I. R. Munro, *Anthropometric Facial Proportions in Medicine*. Springfield, IL: Thomas Books, 1987.
- [Far94b] L. G. Farkas, *Anthropometry of the Head and Face*. New York, NY: Raven Press, 1994.
- [Mpi07a] I. Mpiperis, S. Malassiotis, M.G. Strintzis, 3-d face recognition with the geodesic polar representation, *IEEE Trans. Inf. Forensics Security* 2 (3) (2007) 537~547.
- [Sam06a] C. Samir, Three-dimensional face recognition using shapes of facial curves, *IEEE Trans. Pattern Anal.* 28 (11) (2006) 1858~1863.
- [Ter08a] F. B. ter Haar and R. C. Velkamp. A 3D Face Matching Framework. In *Proc. Shape Modeling International (SMI'08)*, 2008.
- [Ter08b] F.B. ter Haar, R.C. Velkamp, *Shrec'08* entry: 3D face recognition using facial contour curves, in: *SMI'08: Proceedings of the IEEE International Conference on Shape Modeling and Applications*, Stony Brook, NY, USA, 2008.
- [Fen07a] S. Feng, H. Krim, I.A. Kogan, 3D face recognition using Euclidean integral invariants signature, in: *SSP'07: IEEE/SP 14th Workshop on Statistical Signal Processing*, 2007, 156~160.
- [Bes92a] P.J. Besl, N.D. McKay, A method for registration of 3-d shapes, *IEEE Trans. Pattern Anal.* 14 (2) (1992) 239~256.
- [Med03a] G. Medioni, R. Waupotish, Face modeling and recognition in 3-d, in: *IEEE Int'l Workshop Analysis and Modeling of Faces and Gestures*, 2003.
- [Cha05a] K.I. Chang, K. Bowyer, P. Flynn, Effects on facial expression in 3D face recognition in: *SPIE Conf. Biometric Technology for human Identification*, 2005.
- [Xlu06a] X. Lu, A.K. Jain, D. Colbry, Matching 2.5d face scans to 3D models, *IEEE Trans. Pattern Anal.* 28 (2006) 31~43.
- [Cha06b] K.I. Chang, K.W. Bowyer, P.J. Flynn, Multiple nose stripe matching for 3D face recognition under varying facial expression, *IEEE Trans. Pattern Anal.* 28 (10) (2006) 1695~1700.
- [Ber03a] S. Berretti, A. Del Bimbo, and E. Vicario, Weighted walkthroughs between extended entities for retrieval by spatial arrangement, *IEEE Transactions on Multimedia*, vol. 5, no. 1, pp. 52~70, March 2003.
- [Ber10b] S. Berretti, A. Bimbo and P. Pala, 3D face recognition using isogeodesic stripes, *IEEE transaction on pattern analysis and machine intelligence*, vol. 32, no. 12, December 2010
- [Pey04a] G. Peyré and L. D. Cohen. Surface Segmentation Using Geodesic Centroidal Tesselation. *Proc. 3D Data Processing Visualization Transmission 2004*, Sept. 2004.
- [Vit05a] Vitaly Surazhsky, Tatiana Surazhsky, Danil Kirsanov, Steven J. Gortler and Hugues Hoppe, Fast Exact and Approximate Geodesics on Meshes. *ACM Transactions on Graphics*, 24(3) SIGGRAPH 2005, (2005) pages 553-560.
- [Sut09a] M.A. Sutton, J. Orteu, H. Schreier, *Book - Image Correlation for Shape, Motion and Deformation Measurements*, Hardcover (2009), ISBN 978-0-387-78746-6.
- [Guf06a] Gufeng Wanga, John Karnesb, Christopher E. Bunker Two-dimensional correlation coefficient mapping in gas chromatography: Jet fuel classification for environmental analysis *Journal of Molecular Structure* 799 (2006) 247~25

# WSCG 2012

## Posters

### Index

Ahmadi,A.	poster	23
Bay,T.	poster	27
Bezděka,M.	poster	15
Daniel,M.	poster	27
Fares,C.	poster	31
Horáček,J.	poster	35
Huang,D.	poster	43
Hung,C.-J.	poster	11
Chmelík,J.	poster	15
Kheyfets,A.L.	poster	5
Kolomazník,J.	poster	35
Kovalčík,V.	poster	15
Krajíček,V.	poster	35
Li,H.	poster	43
Pelikán,J.	poster	35
Raffin,R.	poster	27
Rostamabadi,F.	poster	23
Seeman,M.	poster	39
Simac-Lejeune,A.	poster	19
Sochor,J.	poster	15
Vergeest,J.S.M.	poster	1
Wang,J.-S.	poster	11
Wu,Z., Huang,D.	poster	43
Yang,K.-C.	poster	11
Yu,C.-H.	poster	11
Zafar Asoodeh,A.	poster	23
Zemčík,P.	poster	39

