Haptic rendering: How do we touch an object?

Subhasis Chaudhuri
IIT Bombay
www.ee.iitb.ac.in/~sc

IIT Roorkee, March, 2014

Outline

- Introduction
- Virtual reality & tele-robotic applications
- What is haptic rendering?
- Various examples
- Conclusions and homeworks
- References for those who needs.

Introduction

- **Haptics** is the science of touch

- **Human haptics** - study of human sensing and manipulation through touch.
- **Machine haptics** - study of design, construction, and use of machine to replace or augment human touch.
- **Computer haptics** - deals with models and behavior of virtual objects together with rendering algorithm for real time interaction.

About Touching

- Unlike others, touch is an active sensing mechanism.
- Based on force feedback
- Two types – kinesthetic and tactile
- Our study is on kinesthetic aspect.

Haptic Devices

- Consumer products:
  - Mouse
  - Joystick
  - Steering wheel
- Professional devices:
  - CyberGrasp
  - CyberForce
  - Phantom

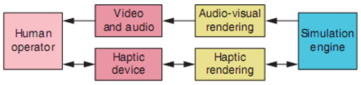
Source: Microsoft

Source: Immersion

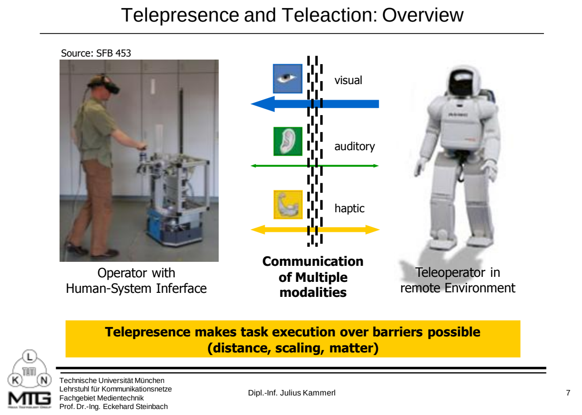
Source: SensAble Technologies, www.sensable.com

Virtual Reality System

- Haptic interface (Haptic device)
  - Electro-mechanical system
- Simulated Graphical objects
  - Contain shape and other properties
- Haptic rendering algorithm
  - Connects the first two components



(Courtesy: K Salisbury et. al.)



Telepresence and Teleaction: Overview

Source: SFB 453

Operator with
Human-System Inferface

Communication of Multiple modalities

Teleoperator in
remote Environment

Telepresence makes task execution over barriers possible (distance, scaling, matter)

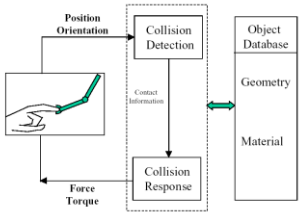
Technische Universität München
Lehrstuhl für Kommunikationsnetze
Fachgebiet Medientechnik
Prof. Dr.-Ing. Eckehard Steinbach

Dipl.-Inf. Julius Kammerl

7

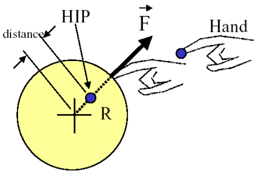
Haptic Rendering

- The process associated with generating and displaying the touch and feel of virtual object to human operator through a force reflecting device



(Courtesy: C Basdogan et. al.)

Illustration of simple rendering



Reaction force follows spring law F=kx

k: stiffness of object

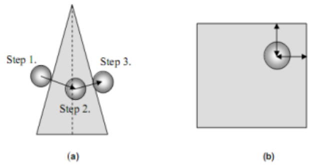
x: depth of penetration

Direction of force along surface normal

(Courtesy: C Basdogan et al.)

Problems with Single Point Interaction

- Thin objects
- When HIP is equidistant from two faces



(Courtesy: S D Laycock et. al.)

God Object Rendering

- Define additional variables to represent the *virtual location* of the haptic interface (god-object, ideal haptic interface point(IHIP), proxy)
- In free space, HIP and IHIP are co-located
- When HIP moves into an object, the IHIP remains on the surface
- IHIP computed such that its distance from the HIP is minimized

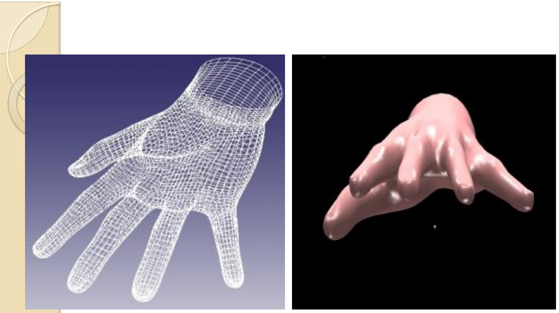
God Object Method (Contd.)

- Define a polygonal mesh of the bounding surface
- Location computation using Lagrange multipliers
  - x, y, z: coordinates of IHIP
  - x_p, y_p, z_p: coordinates of HIP
  - Constraints added as planes

$$Q = \frac{1}{2}(x - x_p)^2 + \frac{1}{2}(y - y_p)^2 + \frac{1}{2}(z - z_p)^2. \quad (1)$$

$$A_n x + B_n y + C_n z - D_n = 0 \quad (2)$$

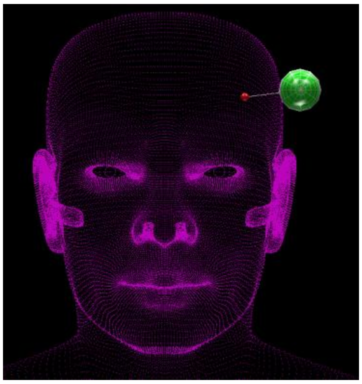
Problems moving the proxy across sharp edges and sinking in small gaps.



(a) (b)

(a) Mesh model of hand. (b) Visually rendered hand. Sphere represents the position of the haptic device in the 3D environment.

[Data Courtesy: http://www.zbrushcentral.com/showthread.php?t=66847]



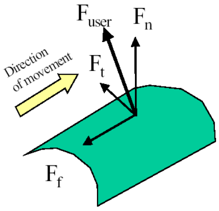
God-object renderer. Green represents proxy and red the HIP.

Home work #1

- Given a 3D mesh for an object, find an efficient way to detect collision.
- If you are given a 3D dense point cloud instead of a mesh, how would you detect the collision?

Surface Properties

- Stiffness
- Friction
- Smoothness
- Texture
- Rigidity



(Courtesy: C Basdogan et al.)

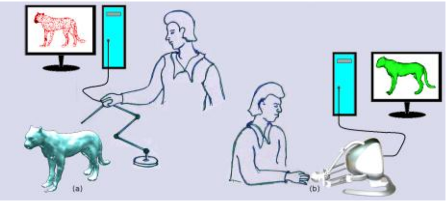
Issues

- How to handle point cloud data?
- Can we handle data sparsity?
- How to make sure that the proxy does not sink?
- How to render it faster than 1 ms?
- How does one handle scale of an object?
- How do we avoid dynamic meshing?

DISTANCE FIELD BASED OBJECT RENDERING

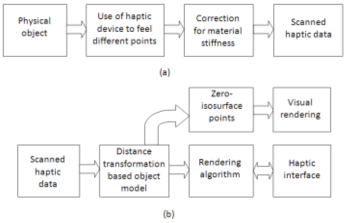
Data (**P**, **N**) Generation

- Hold a haptic device and try to feel the object. Force and position values are sampled.
- Render sparse data using a haptic device.



Data Generation (Cont..)

- Perform stiffness correction
- From object boundary, compute distance field
- Render the object haptically.
- Visual rendering for hapto-visual immersion.


Physical object

Use of haptic device to feel different points

Correction for material stiffness

Scanned haptic data

(a)

Scanned haptic data

Distance transformation based object model

Zero-isosurface points

Visual rendering

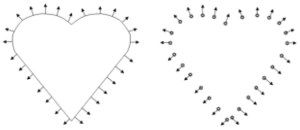
Rendering algorithm

Haptic interface

(b)

Haptic Scanning of Object

- Advantage-
  - scanned 3D points also provide the normal direction.
- Disadvantage-
  - Scanned points are highly sparse



Volume Reconstruction

- The problem: Given a sparse set of augmented points S(P,N), find the bounding surface and hence the distance field over a uniform grid of voxels.
- Distance field is computed in regular 3D grid of voxel from available haptic data through sphere packing. Computational advantage is due to following lemma.
- Lemma 1. *When a single point and the direction of normal at that point is given only one additional point is needed to uniquely define the sphere on which they both lie.*

Circle Embedding of Point Set

- Explanation in 2D with circle instead of sphere
- P_1,P_2,P_3,P_4 are arbitrary points in the set, the radius of the embedded circle between points P_1 and P_2 is

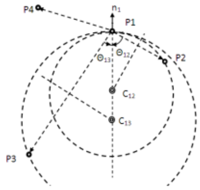
$$r_{12} = \frac{1}{2}|a_{12}|\sec(\theta_{12})$$

$$c_{12} = P_1 - r_{12}.\vec{n}_1$$

Where a_{12} is the vector connecting the points P_1 to P_2 .

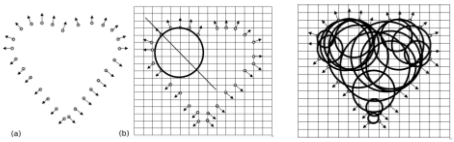
- The radius of the embedded circle at P_i is $r_i = \min_{j \neq i} r_{ij} \forall j$

and C_i is the corresponding center



Circle Embedding of Point Set

- Point set is mapped on to a regular 2D grid
- Embed circle at each point
- Take the union of spheres (ك)



Distance Field Computation

Distance at a point x is approximated as

Φ (x) =   0 if x !є ك

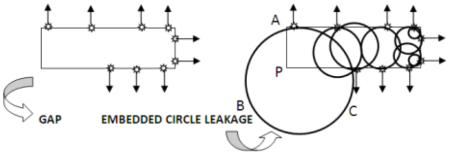
= {n | x є ∂(ك Ɵ nB) }

boundary of n-times eroded set

B is a unit radius structuring element

Problems with Sparse Sampled Data

- Circle ABC leaks out as there is no sampled point near P.
- Also, may not be topologically connected.



Haptic Rendering

- Collision detection

If X_h is the HIP, no collision detected if $\Phi(X_h) = 0$ when rendered force $F(X_h)=0$

- Force computation

If $\Phi(X_h) > 0$, reaction force is proportional to gradient of the computed distance field.

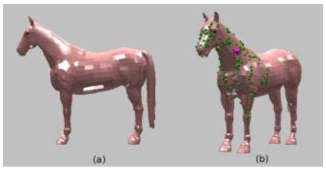
$$F(X_h) = k\,|\phi(X_h)|\,\frac{\nabla\phi(X_h)}{|\nabla\phi(X_h)|}$$

Sample Results

- **Implementation**
  - in visual C++ in a windows XP platform with a CORE 2QUAD CPU @ 2.66GHZ with 2GB RAM.
- **Data set generation**
  - created virtual objects using polygonal meshes
  - Objects scanned manually with a 3-DOF haptic device
- **Reconstruction of objects**
  - reconstructed in a fixed 300X300X300 regular grid.
- **Haptic rendering**
  - force calculation takes only 0.02ms (less than 1ms).

Sampling Polygonal Mesh

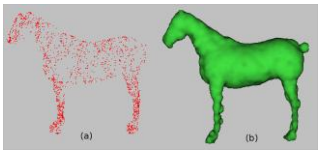
- Model of a horse with 5K polygon



(a) Polygonal mesh of horse (courtesy www.turbosquid.com') b) mesh model with a different viewing angle

Reconstructed Object

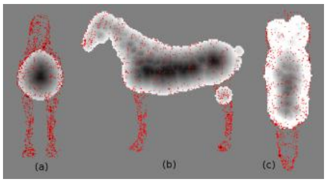
Model	- Horse (5079 polygon)
Number of sampled points	- 1836
Grid size	- 59X157X178
Number of inside grid nodes	- 2,74,075
Time required	- 11.7 Seconds



(a) Pre-processed haptic samples, as obtained after scanning, representing the object. (b) Zero-isosurface points of the reconstructed horse for visual rendering.

Cross-section of Distance Field

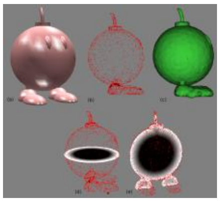
- White pixels are near the boundary of the object. Darkness increases as we move inside the object



Distance field at selected cross-sections of the horse along XY, YZ and XZ planes, respectively.

More results

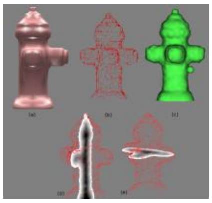
Model	- Bomberman (2480 polygon)
Number of sampled points	- 3,751
Grid size	- 133X189X132
Number of inside grid nodes	- 11,30,637
Time required	- 2 m 43 s



(a) A .obj model of bomberman (courtesy 'www.turbosquid.com') (b) sampled sparse point set after pre-processing (c) zero-isosurface points of the reconstructed object for visual rendering, (d) the distance field for a horizontal cross-section of bomberman, and (e) the same for a vertical cross-section.

More results

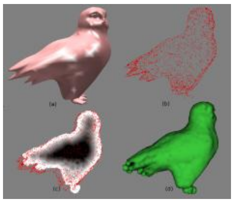
Model	- Hydrant (944 polygon)
Number of sampled points	- 3,756
Grid size	- 131X188X109
Number of inside grid nodes	- 5,91,775
Time required	- 44 Seconds



(a) A .obj model of a water hydrant (courtesy 'www.turbosquid.com') (b) The sampled sparse point set after preprocessing, (c) zero-isosurface points of the reconstructed object, (d) the distance field for a vertical cross-section of the reconstructed object, and (e) the same for a horizontal cross-section.

More results..

Model	- Owl (7652)
Number of sampled points	- 3,851
Grid size	- 188X184X89
Number of inside grid nodes	- 8,27,617
Time required	- 1 minute



(a) A .obj model of an owl (courtesy 'www.top3dmodels.com') (b) The sampled sparse point set after pre-processing, (c) the distance field for a vertical cross-section of the object, and (d) the zero-isosurface of the reconstructed object.

Homework #2

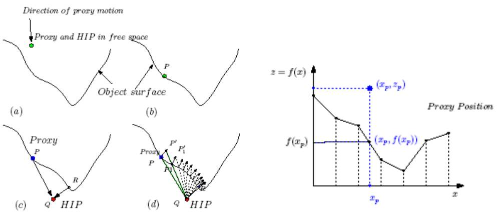
- May have a non-smooth/bumpy experience.
- Would smoothing $\Phi(x)$ help?
- Reconstruction (off-line) is a bit slow but interaction is fast.
- How to handle varying stiffness?
- Can we make it proxy-based?

Virtual Museum

Motivation

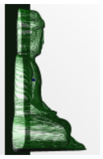
1. Access of cultural heritage objects to visually impaired people
2. Better realistic and immersive experience to sighted persons
3. Issues – scalability and stereoscopic display

Proposed Method



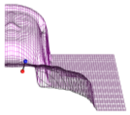
Rendering

1. Graphic Rendering using quad mesh



2. Haptic Rendering

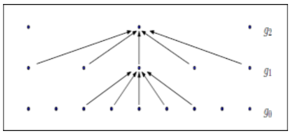
   Proxy based



Data Courtesy: www.archibaseplanet.com

Data Generation

Gaussian Pyramid - Gaussian low-pass filtering followed by down sampling to obtain depth image data at different level of details



One – dimensional graphical representation of Image Pyramid

The value of each pixel at level l is given by

$$g_l(i,j) = \sum_{m=-2}^{2} \sum_{n=-2}^{2} w(m,n) g_{l-1}(2i+m, 2j+n)$$

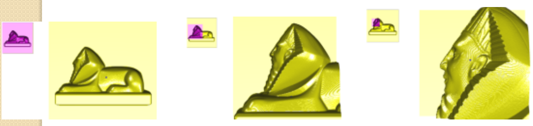
Results

Data Courtesy: www.archibaseplanet.com

Data Courtesy: www.gatech.edu/projects/large_models

Results

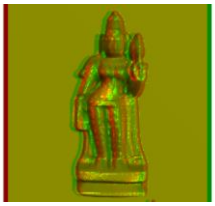
Data Courtesy: www.archibaseplanet.com



Data Courtesy: www.archibaseplanet.com

Results

Stereoscopic display



Data Courtesy: www.archibaseplanet.com

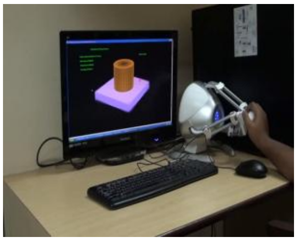
Volume Preserving Haptic Pottery

Haptics Symposium, Houston, TX 2014

Introduction

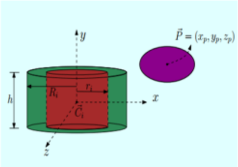
- **Objective**: Given a certain amount of clay develop a system for haptic pottery constrained to volume conservation.
- **Key points:**

1. External or internal interaction with tool.
2. Height of clay volume may increase due to interaction.
3. Mechanical stability aspects of the clay model is also looked into.



Bulk Clay Model

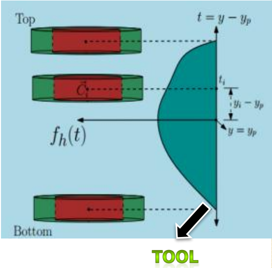
- Clay body is initially assumed to be composed of N cylinders vertically stacked up.
- The number of cylinders may increase during interaction.
- The ith cylinder has an external radius of R_i and internal radius of r_i.



Courtesy: http://imperfectspirituality.com/wp-content/uploads/2010/09/Stack-of-coins1.jpg

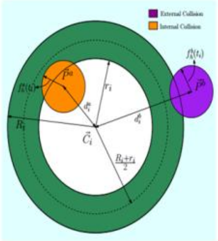
Structure of the haptic tool

- Radially symmetric haptic tool.
- Implemented by revolving a 2D curve about the y-axis.
- The indices *top* and *bottom* gives the interaction of the haptic tool with the clay body.



Collision Detection

- Collision is of two types: internal and external.
- Condition for collision detection are easily checkec using distances.
- Demarcating radius used to differentiate between external and internal collision.



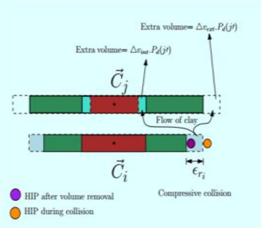
Clay redistribution

- On collision with the tool the radii are modified as

$$R_i^{new} = R_i - \varepsilon_{r_i}.s_i.$$

$$r_i^{new} = r_i + \varepsilon_{r_i}.s_i.$$

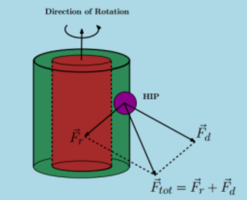
- Distribute removed clay following Rayleigh distribution.
- Movement of the haptic tool decides direction of clay redistribution.

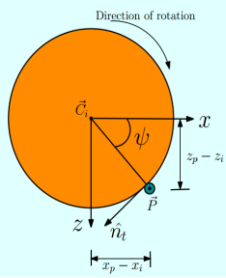

Extra volume= $\Delta v_{ext}.P_d(j')$
Extra volume= $\Delta v_{int}.P_d(j')$
$\vec{C}_j$
Flow of clay
$\vec{C}_i$
ϵ_{r_i}
● HIP after volume removal
● HIP during collision
Compressive collision

Force Feedback

- Force feedback is comprised of

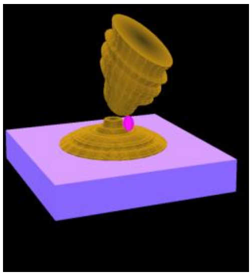
1. Frictional force
2. Spring force due to deformation.





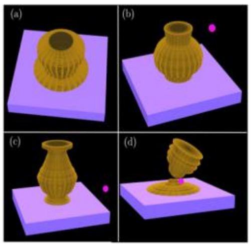
Mechanical Stability

- Stress at each cylinder level is calculated.
- If stress exceeds certain threshold value then the structure collapses.
- Further interaction is with the remaining clay in such a case.



Results

- A 3 degree of freedom Falcon haptic device from NOVINT was used for the interaction.
- Clay preservation inaccuracy : 0.44% of initial volume.
- User rating : 4.0/5.0



Hand Tremor Detection in a Virtual Haptic Environment

(Caltech, June 2013)

Motivation

- Detect hand tremor at an early stage.
- Accelerometry affects the dexterity and EMG is invasive.
- Haptic devices are now affordable and accurate
- As a gaming device your (grand)kids will have it at home
- Virtual environment can be altered as and when required.

Hand Tremor

- Hand tremor: unintentional and rhythmic movement
- Tremor types :
  - Essential (4-12 Hz)
  - Parkinson's (3-7 Hz)
  - Physiological (3-30 Hz)

Methodology

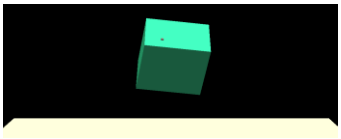
- Define a virtual haptic task.
- Subject is asked to perform the task.
- Collect the velocity information.
- Pre-process data to remove task related motion
- Perform spectral analysis of data
- Peak, if any, shows tremor

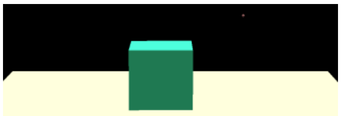
.

Haptic task specification

1. Moving a rigid object

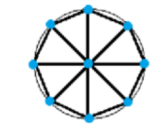
(a)

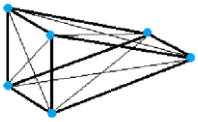

(b)

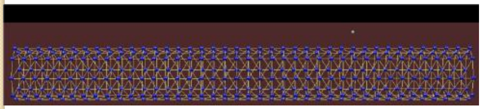

(c)

- Contacts are impulsive
- Generates transients
- Pin-ball type motion
- Effect of mass can be studied

2. Handling Soft, deformable object


Cross-sectional view of the cylindrical object


A finite volume tetrahedral element showing the diagonal springs.


Mass Spring Model

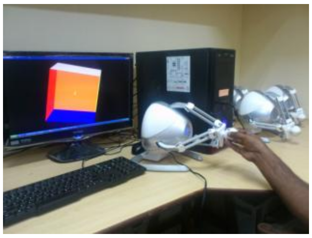
2. Soft Object Manipulation

- Smooth contact with object
- Prolonged interaction
- High computational cost
- Stereoscopic display for better hapto-visual immersion

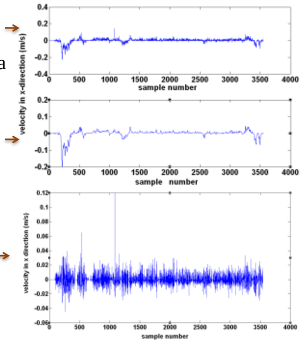
Stages of haptic task

3. Interacting with variable density medium

- To study nature of resting/static tremor
- Is there an effect of loading?



Tremor signal recovery (x-axis)

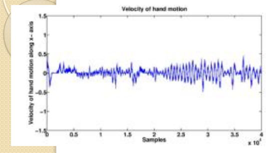


Raw data

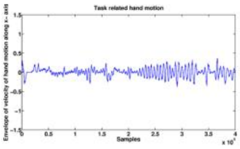
Baseline motion

Tremor signal

Another Tremor Signal

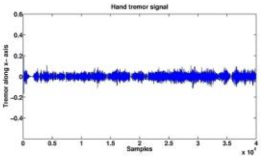


Raw haptic data



Baseline data

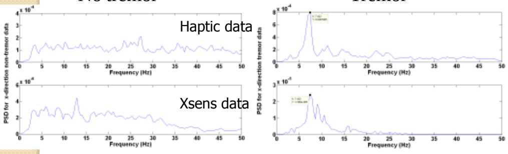
Tremor data



Validation of Method

Using accelerometer based motion sensor

No tremor Tremor

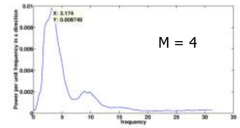


Haptic data

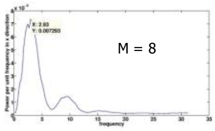
Xsens data

Power spectral density plots

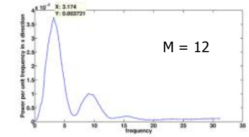
Results for rigid object manipulation



M = 4



M = 8

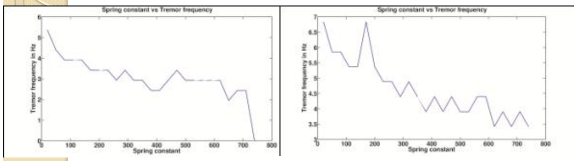


M = 12

Mass	Tremor frequency	Amplitude
4	3.17 Hz	0.0097
8	2.93 Hz	0.0073
12	3.17 Hz	0.0037

PSDF PLOT

Results of static tremor with loading



Tremor frequency vs. Spring constant plot

Conclusions

❑ Haptics is an interesting research area

❑ How efficiently can you move the proxy during the interaction?

❑ Computation has to be fast enough

❑ How do you render a deformable object?

❑ How to handle the communication issues?

❑ What all new applications can you develop?

THANKS TO

Sreeni, Praseedha, Priyadarshini,
Sajith & Subhojit

References

- [1] C. W. Baker, How Did They Do it? Computer Illusion in Film and TV. Alpha Books, 1994.
- [2] R. Kolluri, J. R. Shewchuk, and J. F. O'Brien, "Spectral surface reconstruction from noisy point clouds," in Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing, ser. SGP '04. New York, NY, USA: ACM, 2004, pp. 11–21.
- [3] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, G. Taubin, and S. Member, "The ball-pivoting algorithm for surface reconstruction," IEEE Transactions on Visualization and Computer Graphics, vol. 5, pp. 349–359, 1999.
- [4] N. Amenta, S. Choi, and R. K. Kolluri, "The power crust, unions of balls, and the medial axis transform," Computational Geometry: Theory and Applications, vol. 19, pp. 127–153, 2000.
- [5] N. Amenta, M. W. Bern, and M. Kamvysselis, "A new voronoi-based surface reconstruction algorithm," in SIGGRAPH, 1998, pp. 415–421.

References (Cont..)

- [6] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans, "Reconstruction and representation of 3d objects with radial basis functions," in Computer Graphics (SIGGRAPH 01 Conf. Proc.), pages 6776. ACM SIGGRAPH. Springer, 2001, pp. 67–76.
- [7] H. Hoppe, T. DeRose, T. Duchamp, J. Mcdonald, and W. Stuetzle, "Surface reconstruction from unorganized points," in COMPUTER GRAPHICS (SIGGRAPH 92 PROCEEDINGS), 1992, pp. 71–78.
- [8] H.-K. Zhao, S. Osher, and R. Fedkiw, "Fast surface reconstruction using the level set method," in Proceedings of the IEEE Workshop on Variational and Level Set Methods (VLSM'01), ser. VLSM '01. Washington, DC, USA: IEEE Computer Society, 2001, pp. 194–.
- [9] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in Proceedings of the fourth Eurographics symposium on Geometry processing, ser. SGP '06. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2006, pp. 61–70.

References (Cont..)

- [10] L. Kim, G. S. Sukhatme, and M. Desbrun, "A haptic rendering technique based on hybrid surface representation," IEEE Computer Graphics and Applications, Special Issue on Haptic Rendering - Beyond Visual Computing, vol. 24, no. 2, pp. 66–75, March 2004.
- [11] K. Salisbury, F. Conti, and F. Barbagli, "Haptic rendering: Introductory concepts," IEEE Computer Graphics and Applications, vol. 24, no. 2, pp. 24–32, 2004.
- [12] L. S.D. and D. A.M., "A survey of haptic rendering techniques," Computer Graphics Forum, vol. 26, no. 1, pp. 50–65, March 2007.
- [13] C. Zilles and J. Salisbury, "A constraint-based god-object method for haptic display," Intelligent Robots and Systems, IEEE/RSJ International Conference on, vol. 3, p. 3146, 1995.
- [14] D. C. Ruspini, K. Kolarov, and O. Khatib, "The haptic display of complex graphical environments," in Proc. of ACM SIGGRAPH, 1997, pp. 345–352.

References (Cont..)

- [15] S. M. A. Morgenbesser, H.B, "Force shading for haptic shade perception"," in Proceedings of the ASME Dynamic Systems and Control Division, vol. 58, 1996, pp. 407–412.
- [16] R. S. Avila and L. M. Sobierajski, "A haptic interaction method for volume visualization," Visualization Conference, IEEE, vol. 0, p. 197, 1996.
- [17] W. A. Mcneely, K. D. Puterbaugh, and J. J. Troy, "Six degree-of-freedom haptic rendering using voxel sampling," in Proc. of ACM SIGGRAPH, 1999, pp. 401–408.
- [18] M. Renz, C. Preusche, M. Ptke, H. peter Kriegel, and G. Hirzinger, "Stable haptic interaction with virtual environments using an adapted voxmap-pointshell algorithm," in In Proc. Eurohaptics, 2001, pp. 149–154.
- [19] L. Kim, A. Kyrikou, M. Desbrun, and G. Sukhatme, "An implicit-based haptic rendering technique," in In Proceeedings of the IEEE/RSJ International Conference on Intelligent Robots, 2002.

References (Cont..)

- [20] M. W. Jones, J. A. Brentzen, and M. Sramek, "3d distance fields: A survey of techniques and applications," IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, vol. 12, pp. 581–599, 2006.
- [21] B. A. Payne and A. W. Toga, "Distance field manipulation of surface models," IEEE Comput. Graph. Appl., vol. 12, pp. 65–71, January 1992.
- [22] A. Gu´eziec, "'meshsweeper': Dynamic point-to-polygonal-mesh distance and applications," IEEE Transactions on Visualization and Computer Graphics, vol. 7, pp. 47–61, January 2001.
- [23] F. Dachille and A. E. Kaufman, "Incremental triangle voxelization," in Graphics Interface, 2000, pp. 205–212.
- [24] S. Mauch, "A fast algorithm for computing the closest point and distance transform," California Institute of Technology, Tech. Rep., 2000.

References (Cont..)

- [25] C. Sigg, R. Peikert, and M. Gross, "Signed distance transform using graphics hardware," in Proceedings of the 14th IEEE Visualization 2003 (VIS'03), ser. VIS '03. Washington, DC, USA: IEEE Computer Society, 2003, pp. 12
- [26] A. Rosenfeld and J. L. Pfaltz, "Sequential operations in digital picture processing," J. ACM, vol. 13, pp. 471–494, October 1966.
- [27] F. Rhodes, "Discrete euclidean metrics," Pattern Recogn. Lett., vol. 13, pp. 623–628, September 1992.
- [28] J. C. Mullikin, "The vector distance transform in two and three dimensions," CVGIP: Graph. Models and Image Process., vol. 54, pp. 526–535, November 1992.
- [29] J. N. Tsitsiklis, "Efficient algorithms for globally optimal trajectories," IEEE Transactions on Automatic Control, vol. 40, no. 9, pp.1528–1538, 1995.

References (Cont..)

- [30] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts." Proceedings of the National Academy of Sciences of the United States of America, vol. 93, no. 4, pp. 1591–1595, 1996.
- [31] L. Kobbelt and M. Botsch, "A survey of point-based techniques in computer graphics," Computers and Graphics, vol. 28, pp. 801–814, 2004.
- [32] J.-K. Lee and Y. J. Kim, "Haptic rendering of point set surfaces," World Haptics Conference, vol. 0, pp. 513–518, 2007.
- [33] N. R. El-Far, N. D. Georganas, and A. E. Saddik, "An algorithm for haptically rendering objects described by point clouds," in Proceedings of the 21th Canadian Conference on Electrical and Computer Engineering, Niagara, ON, Canada, 2008.
- [34] I. Baran and J. Popovic, "Automatic rigging and animation of 3d characters," ACM Trans. Graph., vol. 26, no. 3, p. 72, 2007. April