Applications of Information Theory to Computer Graphics

Mateu Sbert
Miquel Feixas
Jaume Rigau

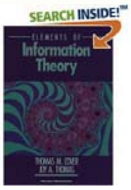
Ivan Viola

Miguel Chover

07 EUROGRAPHICS
Prague Czech Republic

---

(2) Information Theory Basics



Miquel Feixas
Jaume Rigau
Mateu Sbert

07 EUROGRAPHICS
Prague Czech Republic

Information Theory

- Claude Elwood **Shannon**, 1916-2001
- **A mathematical theory of communication**. The Bell System Technical Journal, July, October 1948
- Transmission, storage and processing of information
- Applications:
  - Physics, computer science, mathematics, statistics, economics, biology, linguistics, neurology, learning, etc
  - Medical image processing, computer vision, robot motion, etc
- **Shannon entropy** measures the information content or uncertainty of a random variable
- **Mutual information** measures the information transfer in a communication channel

---

Shannon Entropy

- Discrete random variable X

$$X : \{x_1 , x_2 , \ldots , x_n\} , \ p_i = p(x_i) = Pr \{ X = x_i \}$$

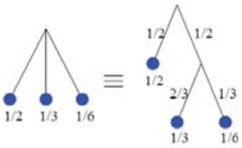
Shannon entropy of X : **uncertainty**, **information**

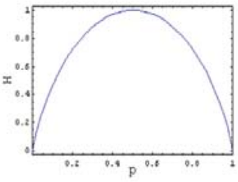
$$\boxed{H(X) = -\sum_{i=1}^{n} p_i \log p_i}$$

- How difficult it is to guess the values of a random variable
- Homogeneity or uniformity of a probability distribution

Shannon Entropy

- Properties
  - $0 \leq H(X) = \log n$
  - $H(X) = \sum_{i=1}^{m} q_i H(Y_i) - \sum_{i=1}^{m} q_i \log q_i$
  - Binary entropy





Information Channel

- Information channel

$$\underset{p_i}{X} \xrightarrow{p_{j\,|\,i}} \underset{q_j}{Y} \qquad \boxed{p_{ij} = p_i p_{j|i}}$$

- Conditional entropy

$$\boxed{H(Y\,|\,X) = -\sum_{i=1}^{n} \sum_{j=1}^{m} p_{ij} \log p_{j|i}}$$

- Joint entropy

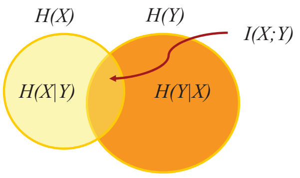
$$\boxed{H(X,Y) = -\sum_{i=1}^{n} \sum_{j=1}^{m} p_{ij} \log p_{ij}}$$

- Mutual information: **dependence**, **correlation**, **shared information**

$$\boxed{\begin{aligned} &I(X,Y) = H(X) - H(X\,|\,Y) \\ &= \sum_{i=1}^{n} \sum_{j=1}^{m} p_{ij} \log \frac{p_{ij}}{p_i q_j} \end{aligned}}$$

Information Channel

- Properties
  - $0 \le H(X|Y) \le H(X)$
  - $H(X,Y) = H(X) + H(Y|X)$
  - $H(X,Y) = H(X) + H(Y) - I(X,Y)$
  - $I(X,Y) = I(Y,X) \ge 0$
  - $I(X,Y) \le H(X)$



Inequalities

- Jensen's inequality: if *f(x)* is a convex function

$$\boxed{f(E[X]) \le E[f(X)]}$$

- Log-sum inequality

$$\boxed{\sum_{i=1}^{n} a_i \log\frac{a_i}{b_i} \ge \left(\sum_{i=1}^{n} a_i\right)\log\frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i}}$$

- Data processing inequality : if $X \to Y \to Z$ is a Markov chain, then $\boxed{I(X,Y) \ge I(X,Z)}$

Relative Entropy

- Kullback-Leibler distance

$$\boxed{D_{KL}(p\|q) = \sum_{i=1}^{n} p_i \log\frac{p_i}{q_j}}$$

- Properties
  - $D_{KL}(p\|q) \ge 0$
  - $I(X,Y) = D_{KL}(\{p_{ij}\}\|\{p_i q_j\})$

Jensen-Shannon Divergence

- Jensen-Shannon divergence

$$\boxed{JS(\pi_1,\ldots,\pi_N; p_1,\ldots,p_N) = H\left(\sum_{i=1}^{N}\pi_i p_i\right) - \sum_{i=1}^{N}\pi_i H(p_i)}$$

$$JS(\pi_1,\ldots,\pi_N; p_1,\ldots,p_N) = \sum_{i=1}^{N}\pi_i D_{KL}\left(p_i \,\|\, \sum_{i=1}^{N}\pi_i p_i\right)$$

- Properties
  - Concavity of entropy: $JS(\pi_1,\ldots,\pi_N; p_1,\ldots,p_N) \ge 0$
  - $JS(p(x_1),\ldots,p(x_n); p(y|x_1),\ldots,p(y|x_n)) = I(X,Y)$

f-Divergences

- Family of convex functions based on a convex function *f*

$$D_f(p,q) = \sum_{x\in X} q(x) f\left(\frac{p(x)}{q(x)}\right)$$

- $D_f(p,q)$ is convex on (p,q)
- $D_f(p,q) \ge 0$
- $D_f(p,q) = 0 \Leftrightarrow p=q$

- Kullback-Leibler distance $\boxed{D_{KL}(p\|q) = \sum_{x\in X} p(x)\log\frac{p(x)}{q(x)}}$
- Chi-square distance $\boxed{D_{\chi^2}(p,q) = \sum_{x\in X}\frac{(p(x)-q(x))^2}{q(x)}}$
- Hellinger distance $\boxed{D_{h^2}(p,q) = \frac{1}{2}\sum_{x\in X}(\sqrt{p(x)}-\sqrt{q(x)})^2}$

Continuous Channel

- Continuous entropy

$$\boxed{H^c(X) = -\int_S p(x)\log p(x)dx} \qquad \boxed{\lim_{\Delta\to 0} H(X^\Delta) \ne H^c(X)}$$

- Continuous mutual information

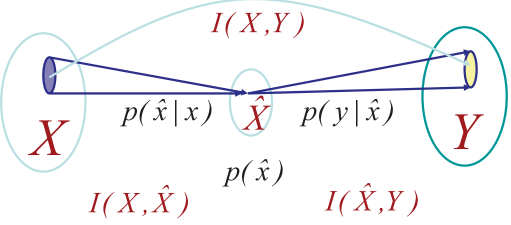
$$\boxed{I^c(X,Y) = \int_S\int_S p(x,y)\log\frac{p(x,y)}{p(x)p(y)}dxdy}$$

$$\boxed{\lim_{\Delta\to 0} I(X^\Delta,Y^\Delta) = I^c(X,Y)}$$

- ***I^c(X,Y)*** is the least upper bound for ***I(X,Y)***
- Refinement can never decrease ***I(X,Y)***

Information Bottleneck Method (IBM)

- Tishby, Pereira and Bialek, 1999
- **Find a compressed signal $\hat{X}$ that needs short encoding (small $I(X,\hat{X})$) while preserving as much as possible the information on the relevant signal ($I(\hat{X},Y)$)**



Agglomerative IBM

- Goal: find a clustering that minimizes the loss of mutual information
- Clustering or merging: loss of mutual information

$$I(X,Y)-I(\hat{X},Y)=$$
$$p(\hat{x})JS(p(\hat{x}_1)/p(\hat{x}),\ldots,p(\hat{x}_m)/p(\hat{x});p(y|\hat{x}_1),\ldots,p(y|\hat{x}_m))$$

$$p(\hat{x})=\sum_{k=1}^{m}p(\hat{x}_k)$$

- The quality of each cluster $\hat{x}$ is measured by the Jensen-Shannon divergence between the individual distributions in the cluster

---

Generalised Entropy

- Harvda-Charvát-Tsallis entropy (HCT)

$$\boxed{H_\alpha(X)=k\frac{1-\sum_{i=1}^{n}p_i^\alpha}{\alpha-1}} \qquad k>0,\alpha\in R\setminus\{1\}$$

$$H_1(X)\equiv\lim_{\alpha\to 1}H_\alpha(X)=-k\sum_{i=1}^{n}p_i\ln p_i$$

- Generalised mutual information

$$\boxed{I_\alpha(X,Y)=\frac{1}{1-\alpha}\left(1-\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{p_{ij}^\alpha}{p_i^{\alpha-1}q_j^{\alpha-1}}\right)}$$

---

(3) Refinement Criteria for Radiosity

Jaume Rigau
Miquel Feixas
Mateu Sbert

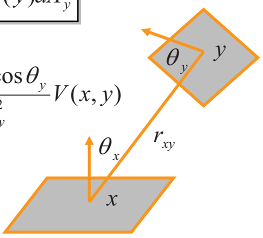
07 EUROGRAPHICS
Prague Czech Republic

Radiosity Method

- The radiosity method solves the problem of illumination in an environment of diffuse surfaces
- Continuous radiosity equation

$$\boxed{B(x)=E(x)+\rho(x)\int_S F(x,y)B(y)dA_y}$$

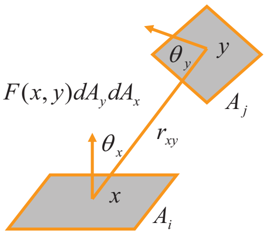
$$F(x,y)=\frac{\cos\theta_x\cos\theta_y}{\pi r_{xy}^2}V(x,y)$$



Radiosity Method

- Discrete radiosity equation

$$\boxed{B_i=E_i+\rho_i\sum_{j=1}^{n_p}F_{ij}B_j} \qquad F_{ij}=\frac{1}{A_i}\int_{A_i}\int_{A_j}F(x,y)dA_ydA_x$$



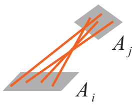
- Form factor properties
  - Reciprocity $\boxed{A_iF_{ij}=A_jF_{ji}}$
  - Energy conservation $\boxed{\sum_{j=1}^{n_p}F_{ij}=1}$

Form Factor Computation

- Analytical solutions
  - Between two spherical patches

$$F_{ij} = \frac{A_j}{A_S}$$

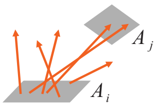
- Monte Carlo computation
  - Uniform area sampling



$$\hat{F}_{ij} = A_j \frac{1}{N} \sum_{k=1}^{N} F(x_k, y_k)$$

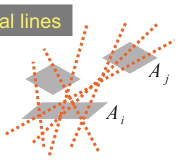
  - Uniformly distributed lines

Local lines



$$\hat{F}_{ij} = \frac{N_{ij}}{N_i}$$

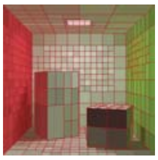
Global lines



Refinement Criteria for HR

- In hierarchical radiosity (HR), the mesh is generated adaptively
- Oracles based on
  - Transported power

$$\rho_i A_i F_{ij} B_j < \varepsilon$$

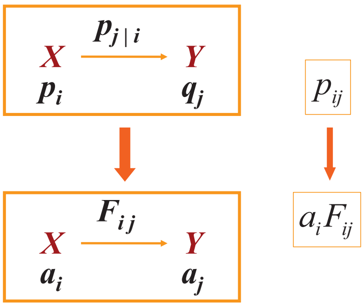


  - Kernel smoothness

$$\rho_i \max(F_{ij}^{\max} - F_{ij}^{av}, F_{ij}^{av} - F_{ij}^{\min}) A_j B_j < \varepsilon$$

Scene Information Channel

- The scene is modelled as an information channel



$$X \xrightarrow{p_{j\,|\,i}} Y$$

p_i q_j

p_{ij}

$$X \xrightarrow{F_{ij}} Y$$

a_i a_j

$a_i F_{ij}$

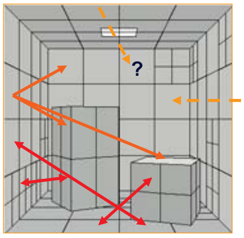
Scene Information Channel

Positional entropy

$$H_P = -\sum_{i=1}^{n_p} a_i \log a_i$$

Scene entropy

$$H_S = -\sum_{i=1}^{n_p} a_i \sum_{j=1}^{n_p} F_{ij} \log F_{ij}$$



$$a_i = \frac{A_i}{A_T}$$

Scene mutual information

$$I_S = H_P - H_S = \sum_{i=1}^{n_p} \sum_{j=1}^{n_p} a_i F_{ij} \log \frac{F_{ij}}{a_j}$$

Continuous Mutual Information

- By discretising a scene, a distortion or error is introduced: information loss
- **From** discrete **to** continuous
  - $\Sigma \rightarrow \int$
  - $F_{ij} \rightarrow F(x,y)$
  - $a_i = A_i / A_T \rightarrow 1 / A_T$

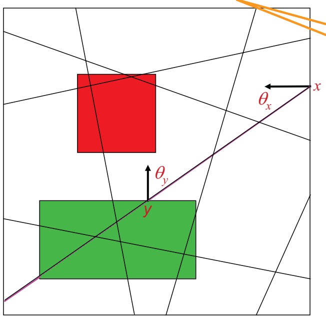
$$I_s^c = \int_{x \in S} \int_{y \in S} \frac{1}{A_T} F(x,y) \log(A_T F(x,y)) dxdy$$

Monte Carlo Computation

$$I_S^c \approx \frac{1}{N} \sum_{1}^{N} \boxed{\log\left(\frac{A_T \cos\theta_x \cos\theta_y}{\pi r_{xy}^2}\right)}$$

contribution of each segment

Total area = A_T



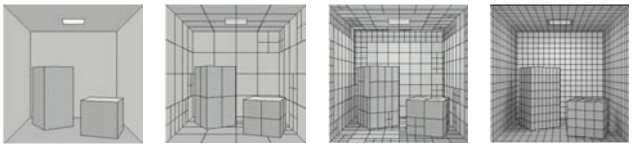
Lines cast = K

Line segments = N

Dicretisation Error

- Two basic results
  - If any patch is subdivided, I_S increases or remains the same
  - I_S^c is the least upper bound to I_S
- Discretisation error $\boxed{I_S^c - I_S \geq 0}$

$$I_S^c = 3.273$$



$I_S = 0.690$ $I_S = 2.199$ $I_S = 2.558$ $I_S = 2.752$

Information Transfer

- Mutual information matrix

$$I_S = \sum_{i=1}^{n_p} \boxed{\sum_{j=1}^{n_p} \boxed{a_i F_{ij} \log \frac{F_{ij}}{a_j}}} \Leftarrow I_{ij}$$

information transfer between patches ***i*** and ***j***

$\Uparrow I_i$

information transfer from patch ***i***

$\Downarrow$

$$I_S^c = \sum_{i=1}^{n_p} \boxed{\sum_{j=1}^{n_p} \boxed{\int_{A_i}\int_{A_j} \frac{1}{A_T} F(x,y)\log(A_T F(x,y))dxdy}}$$

Discretisation Error Between Two Patches

- Discretisation error between two elements: loss of information transfer

$$\boxed{\delta_{ij} = I_{ij}^c - I_{ij}}$$

$\Downarrow$

Monte Carlo integration

log-sum inequality

$$\delta_{ij} \approx \frac{A_i A_j}{A_T}\left[\begin{array}{l}\frac{1}{N_{ij}}\left(\sum_{k=1}^{N_{ij}} F(x_k, y_k)\log F(x_k, y_k)\right) \\ -\left(\left(\frac{1}{N_{ij}}\sum_{k=1}^{N_{ij}} F(x_k, y_k)\right)\log\left(\frac{1}{N_{ij}}\sum_{k=1}^{N_{ij}} F(x_k, y_k)\right)\right)\end{array}\right] \geq 0$$

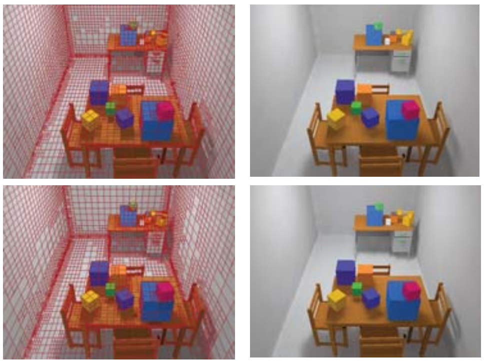
MI-based Oracle

- **From** radiosity equation and kernel-smoothness-based oracle
  - $B_i = E_i + \sum_{j=1}^{n_p} \boxed{\rho_i F_{ij} B_j}$
  - $\rho_i \boxed{\max(F_{ij}^{\max} - F_{ij}^{av}, F_{ij}^{av} - F_{ij}^{\min}) A_j} B_j < \varepsilon$
- **to** MI-based oracle
  - $\boxed{\rho_i (I_{ij}^c - I_{ij}) B_j = \rho_i \delta_{ij} B_j < \varepsilon}$

Oracles for HR

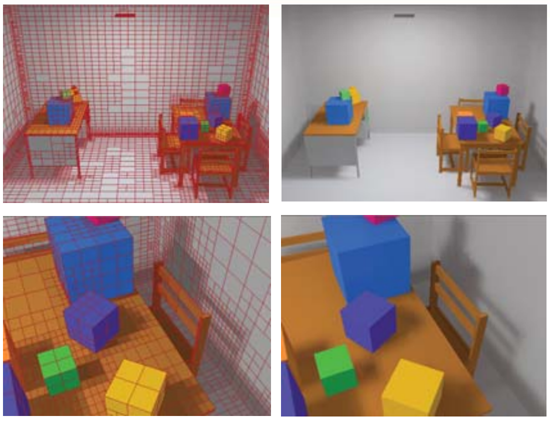
Kernel-smoothness-based

MI-based



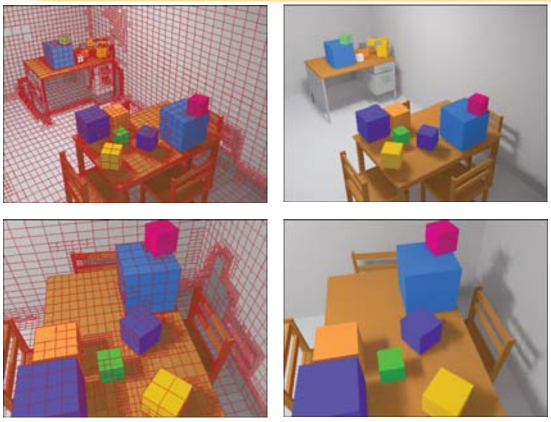
2684000 rays - 19000 patches - 10 lines FF

MI-based Oracle for HR



2684000 rays - 19000 patches - 10 lines FF

Generalised MI-based Oracle

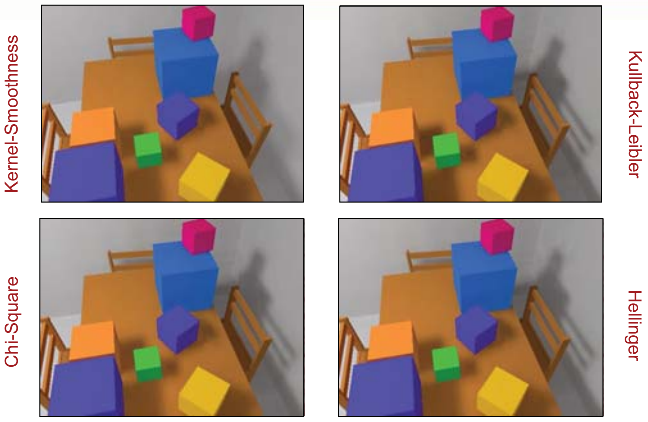


2684000 rays - 19000 patches   α=0.50 - 10 lines FF

Generalised MI-based Oracle

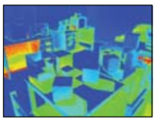
α=0.50 - 10 lines FF - 9268000 rays - 10000 patches

f-Divergence-based Oracles



10 lines FF - 2684000 rays - 19000 patches

(4) Refinement Criteria for Ray-Tracing

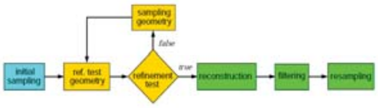


Jaume Rigau
Miquel Feixas
Mateu Sbert

07 EUROGRAPHICS Prague Czech Republic

Adaptive Sampling

- Adaptive control of the sampling rate



- Image-Space
    - Intensity Comparison
    - Intensity Statistics

$$C(S)=\frac{S_{\max}-S_{\min}}{S_{\max}+S_{\min}}$$ [Mitchell, 87]

$$\Pr\{\overline{S}_{\mathrm{T}}\in[\overline{S}-t,\overline{S}+t]\}=1-\alpha$$

[Purgathofer, 87]
[Tamstorf and Jensen, 97]

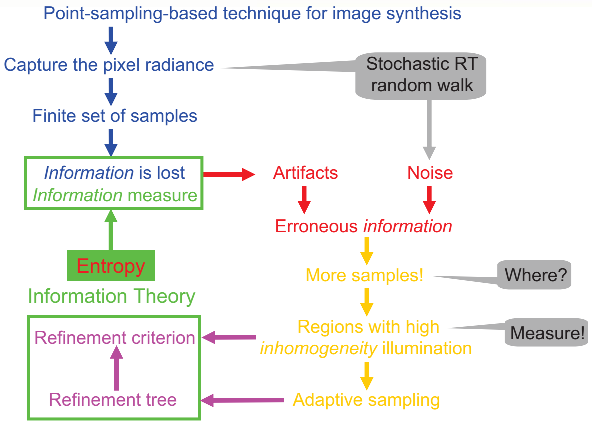
- Object-Space

$$p_g=1-\frac{d_{\min}}{d_{\max}}$$

[Simmons and Séquin, 00]

- Hybrid (image+object spaces)

Pixel Measures



Point-sampling-based technique for image synthesis

Capture the pixel radiance

Stochastic RT random walk

Finite set of samples

Information is lost
Information measure

Artifacts

Noise

Erroneous *information*

Entropy

Information Theory

More samples!

Where?

Regions with high *inhomogeneity* illumination

Measure!

Refinement criterion

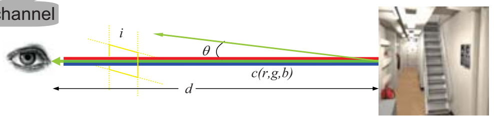
Refinement tree

Adaptive sampling

Pixel Colour Quality

$p_i^c = colour\ \ fraction\ of\ a\ ray$

For each channel



US cruiser
Saba Rofchaei and Greg Ward

pixel channel entropy $H^c = -\sum_{i=1}^{N_s} p_i^c \log p_i^c$

Number of samples

pixel channel quality $Q^c = \frac{H^c}{\log N_s}$

Channel perception coefficient

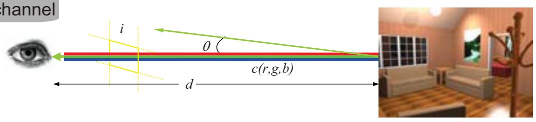
pixel colour quality $\mathbf{Q}^{\mathbf{c}} = \frac{\sum_{c \in \mathbf{c}} w^c Q^c}{\sum_{c \in \mathbf{c}} w^c}$

Colour system

Pixel Colour Contrast

$p_i^c = colour\ \ fraction\ of\ a\ ray$

For each channel



Cabin
Cindy Larson

pixel channel contrast $C^c = 1 - Q^c$

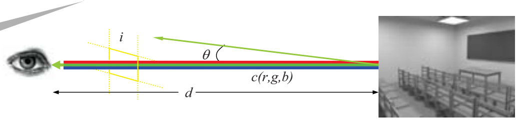
Pixel channel colour average

pixel colour contrast $\mathbf{C}^{\mathbf{c}} = \frac{\sum_{c \in \mathbf{c}} w^c \overline{c}\, C^c}{\sum_{c \in \mathbf{c}} w^c \overline{c}}$

Pixel Geometry Contrast

$p_i^g = geometric\ \ fraction\ of\ a\ ray$

$\frac{\cos\theta}{d^2}$



Class room
Peter Shirley

pixel geometric entropy $H^g = -\sum_{i=1}^{N_s} p_i^g \log p_i^g$

pixel geometric quality $Q^g = \frac{H^g}{\log N_s}$

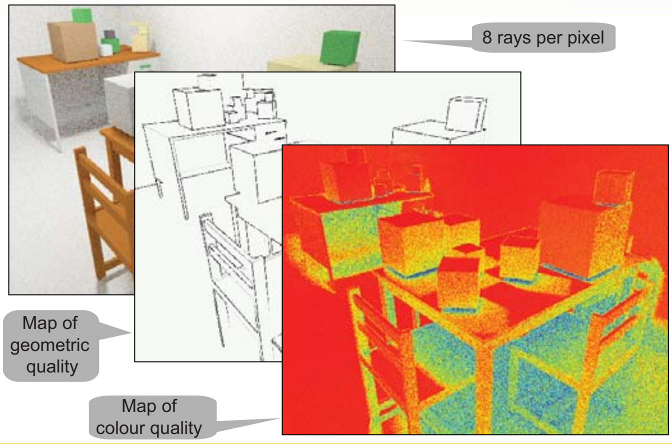
pixel geometric contrast $C^g = 1 - Q^g$

Combination coefficient

pixel contrast $C^{\mathbf{c}} = \delta \mathbf{C}^{\mathbf{c}} + (1-\delta) C^g$

Combination of colour and geometry

Quality Map

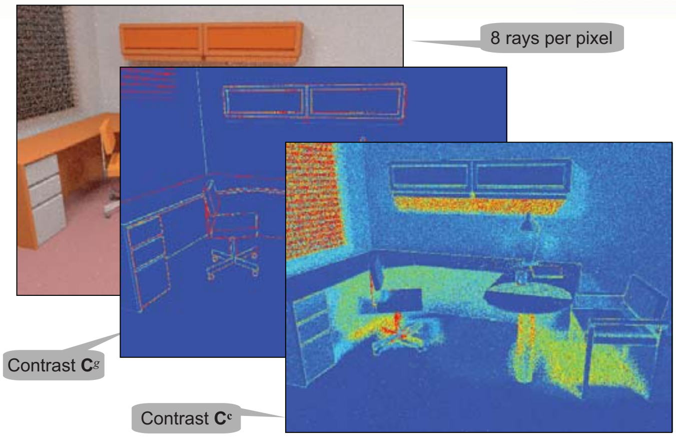


8 rays per pixel

Map of geometric quality

Map of colour quality

Contrast Map

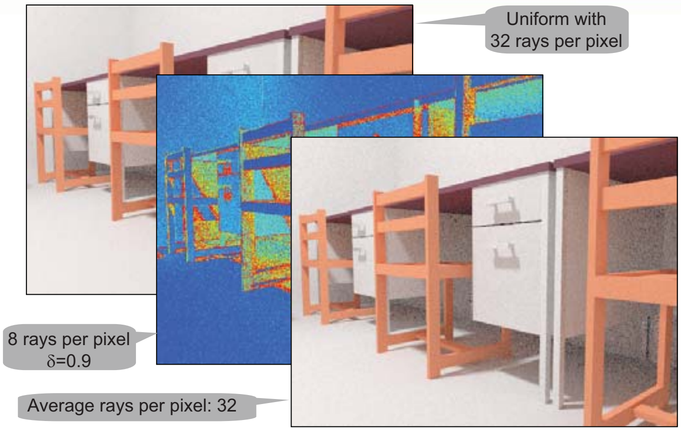


8 rays per pixel

Contrast $\mathbf{C}^g$

Contrast $\mathbf{C}^{\mathbf{c}}$

Supersampling



Uniform with 32 rays per pixel

8 rays per pixel
δ=0.9

Average rays per pixel: 32

Entropy-based Adaptive Sampling

Grouping property of Entropy

image information

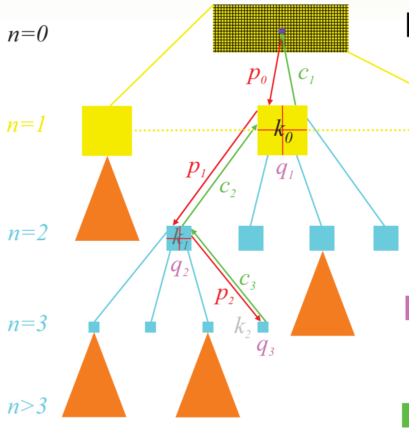
$$H(X) = \underbrace{\boxed{-\sum_{i=1}^{m} q_i \log q_i}}_{\text{information acquired}} + \underbrace{\boxed{\sum_{i=1}^{m} q_i H(Y_i)}}_{\text{hidden information}}$$

- $H(X) \equiv$ entropy of the whole image
- $H(Y_i) \equiv$ entropy of each root pixel
- $q_i \equiv$ colour probability of pixel i

The decomposition of H can be recursively extended to the subpixels

---

Contrast Tree



$k=(k_o,\ldots,k_{n-1})$ is the tree-path

p_n is the probability of the tree-branch

$$q_n = \prod_{\ell=1}^{n-1} p_\ell \approx \frac{\bar{c}_n}{N_r^{n-1}}$$

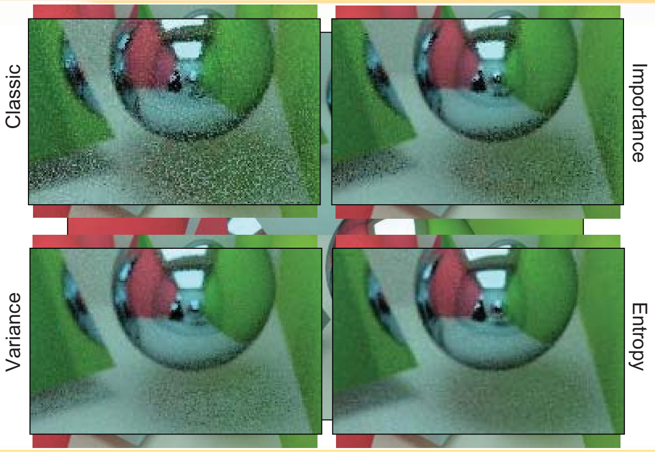
q_n is the importance of a node

$$\mathbf{C}_n^{\mathbf{c}} = \sum_{c\in\mathbf{c}} w^c C_n^c q_n^c$$

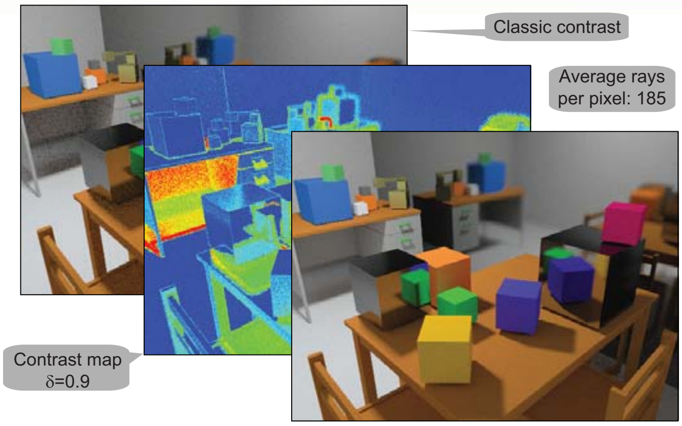
$$C_n^{\mathbf{c}} = \delta\,\mathbf{C}_n^{\mathbf{c}} + (1-\delta) C_n^g$$

c_n is the final colour of a region

Results



Results



Classic contrast

Average rays per pixel: 185

Contrast map δ=0.9

---

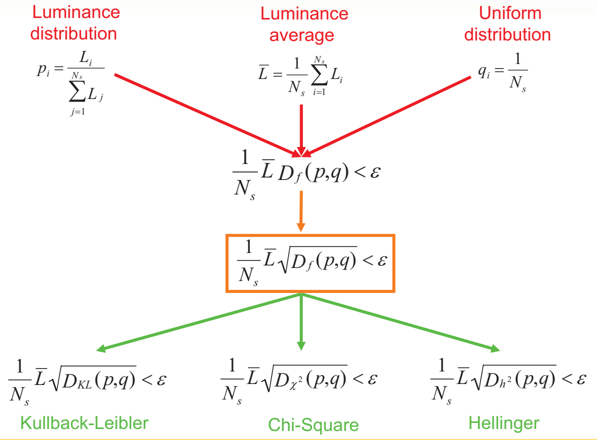
f-Divergences

- f-Divergences as refinement criteria in RT ?
- Distributions
  - $\{p\}$ = Luminance L of N_S-samples
  - $\{q\}$ = Uniform $1/N_S$
- Homogeneity: $D_f(p,q)$
- Weights for D_f
  - Importance: $avg(L_i)$
  - Convergence: $1/N_S$

$$D_f(p,q) = \sum_{x\in X} q(x) f\left(\frac{p(x)}{q(x)}\right)$$

---

f-Divergence-based Adaptive Sampling



Luminance distribution

$$p_i = \frac{L_i}{\sum_{j=1}^{N_s} L_j}$$

Luminance average

$$\bar{L} = \frac{1}{N_s}\sum_{i=1}^{N_s} L_i$$

Uniform distribution

$$q_i = \frac{1}{N_s}$$

$$\frac{1}{N_s}\bar{L}\, D_f(p,q) < \varepsilon$$

$$\boxed{\frac{1}{N_s}\bar{L}\sqrt{D_f(p,q)} < \varepsilon}$$

$$\frac{1}{N_s}\bar{L}\sqrt{D_{KL}(p,q)} < \varepsilon$$

Kullback-Leibler

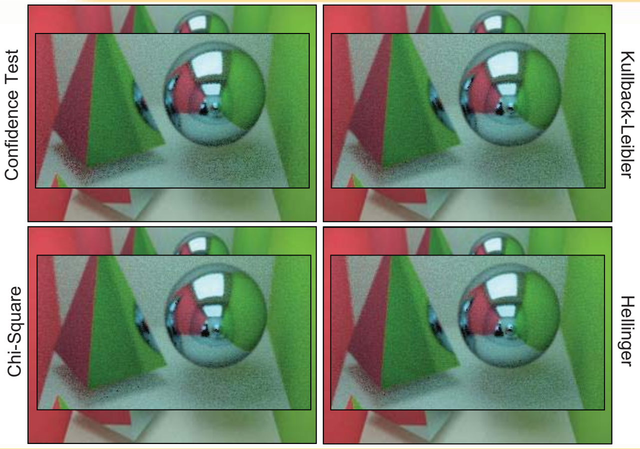
$$\frac{1}{N_s}\bar{L}\sqrt{D_{\chi^2}(p,q)} < \varepsilon$$

Chi-Square

$$\frac{1}{N_s}\bar{L}\sqrt{D_{h^2}(p,q)} < \varepsilon$$

Hellinger

Results



(5) Viewpoint Selection and Mesh Saliency

Miquel Feixas
Mateu Sbert
Francisco González

07 EUROGRAPHICS
Prague Czech Republic

Introduction

- Viewpoint selection is an emerging area in computer graphics with applications in fields such as scene understanding, volume visualization, image-based modeling, and molecular visualization
- We present a unified framework for viewpoint selection and mesh visibility / saliency / simplification based on an information channel between a set of viewpoints and the polygons of an object
- Tools: entropy, mutual information, Jensen-Shannon divergence
- This framework is based on the geometric characteristics of the object, but it can be extended to other characteristics
- It is also valid for any set of viewpoints in a closed scene
- What is a good viewpoint? Depending on our objective, the best viewpoint can be the most representative one or the most unstable one (maximally changes when it is moved within its close neighborhood) or …
  - Representative views can help us to understand the object
  - Unstable views enable us to obtain critical viewpoints to capture the structure of the object

---

Background and Related Work

- **Information Theory**
  - Discrete random variable X

$$X: \{x_1, x_2, \dots, x_n\},\ p(x_i) = Pr\{X = x_i\}$$

  - Shannon entropy of X : uncertainty, ignorance

$$\boxed{H(X) = -\sum_{x\in\mathcal{X}} p(x)\log p(x).}$$

Background and Related Work

- **Information Theory**
  - Information Channel

$$\begin{tikzcd} \underset{\{p(x)\}}{X} \arrow[r, "\{p(y|x)\}"] & \underset{\{p(y)\}}{Y} \end{tikzcd}$$

  - Conditional Entropy

$$H(Y|X) = -\sum_{x\in\mathcal{X}} p(x)\sum_{y\in\mathcal{Y}} p(y|x)\log p(y|x),$$

  - Mutual Information

$$\begin{aligned} I(X,Y) &= H(X) - H(X|Y) = H(Y) - H(Y|X) \\ &= \sum_{x\in\mathcal{X}} p(x)\sum_{y\in\mathcal{Y}} p(y|x)\log\frac{p(y|x)}{p(y)}. \end{aligned}$$

  - Jensen-Shannon inequality

$$\boxed{JS(p_1,p_2,\dots,p_N) = H(\sum_{i=1}^{N}\pi_i p_i) - \sum_{i=1}^{N}\pi_i H(p_i) \geq 0,}$$

Background and Related Work

- **Related Work**
  - Heuristic measure
    Plemenos et al. [1996]

$$\boxed{C(v) = \frac{\sum_{i=1}^{n}\lceil\frac{P_i(v)}{P_i(v)+1}\rceil}{n} + \frac{\sum_{i=1}^{n}P_i(v)}{r},}$$

  - Viewpoint Entropy

$$\boxed{H(v) = -\sum_{i=0}^{N_f}\frac{a_i}{a_t}\log\frac{a_i}{a_t},}$$

  - Kullback-Leibler distance

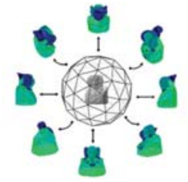
$$\boxed{KL(v) = \sum_{i=1}^{N_f}\frac{a_i}{a_t}\log\frac{\frac{a_i}{a_t}}{\frac{A_i}{A_T}},}$$

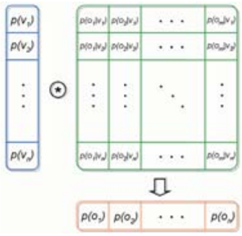
  - **Origins Rigau et. al [2000], Vázquez et al. [2001-2006], Sbert. Et al [2005]**

Viewpoint Information Channel

- We formalize the viewpoint selection using an information channel

$$\underset{\{p(v)\}}{V} \xrightarrow{\{p(o|v)\}} \underset{\{p(o)\}}{O}$$





$$p(o) = \sum_{v\in\mathcal{V}} p(v)p(o|v)$$

- This framework is based on geometric characteristics

Viewpoint Information Channel

- **Viewpoint Mutual Information**
  - Conditional Entropy

$$\begin{aligned} H(O|V) &= -\sum_{v\in\mathcal{V}} p(v) \sum_{o\in\mathcal{O}} p(o|v)\log p(o|v) \\ &= \frac{1}{N_v}\sum_{v\in\mathcal{V}} H(v), \end{aligned}$$

- *H(v)* depends on the polygonal discretization
- *MI* converges to a finite value when the mesh is infinitely refined

  - Mutual Information: degree of correlation, dependence

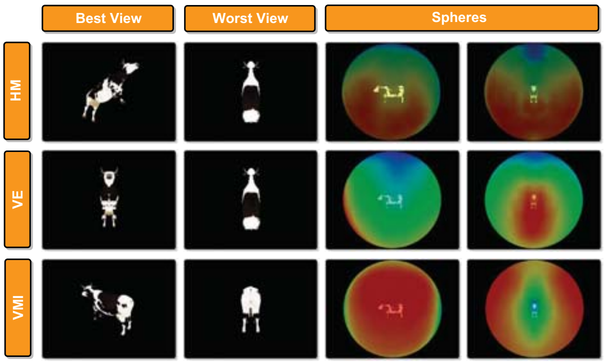
$$\begin{aligned} I(V,O) &= \sum_{v\in\mathcal{V}} p(v) \sum_{o\in\mathcal{O}} p(o|v)\log \frac{p(o|v)}{p(o)} \\ &= \sum_{v\in\mathcal{V}} p(v)I(v,O) \end{aligned}$$

- **Low values:** representative views
- **High values:** highly coupled views

$$I(v,O) = \sum_{o\in\mathcal{O}} p(o|v)\log\frac{p(o|v)}{p(o)}$$

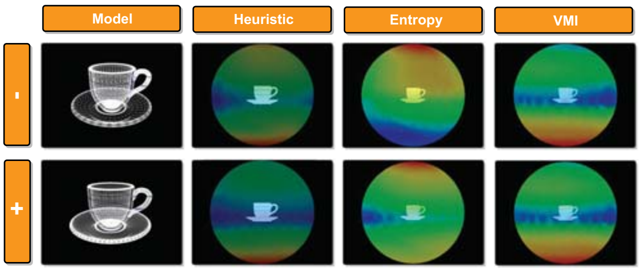
Viewpoint Information Channel

- **Viewpoint Mutual Information evaluation (I)**



Viewpoint Information Channel

- **Viewpoint Mutual Information evaluation (II)**



Viewpoint Information Channel

- Viewpoint Similarity and Unstability
  - Viewpoint Similarity
    - Any clustering over $V \rightarrow \hat{V}$ or $O \rightarrow \hat{O}$ reduce $I(V,O)$

$$\begin{aligned} \delta I(v_i,v_j) &= I(V,O) - I(\widehat{V},O) \\ &= (p(v_i)I(v_i,O) + p(v_j)I(v_j,O)) - p(\widehat{v})I(\widehat{v},O) \\ &= p(\widehat{v})\left(\frac{p(v_i)}{p(\widehat{v})}I(v_i,O) + \frac{p(v_j)}{p(\widehat{v})}I(v_j,O) - I(\widehat{v},O)\right) \\ &= p(\widehat{v})D(v_i,v_j), \end{aligned}$$

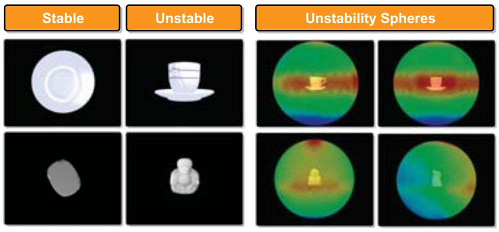
$$D(v_i,v_j) = JS\left(\frac{p(v_i)}{p(\widehat{v})},\frac{p(v_j)}{p(\widehat{v})};p(O|v_i),p(O|v_j)\right)$$

Viewpoint Information Channel

- Viewpoint Similarity and Unstability
  - Viewpoint Unstability

$$U(v_i) = \frac{1}{N_n}\sum_{j=1}^{N_n} D(v_i,v_j)$$

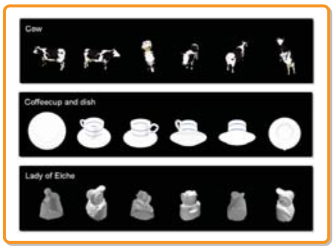
- The maximum change in view that occur when the camera position is shifted within a small neighborhood

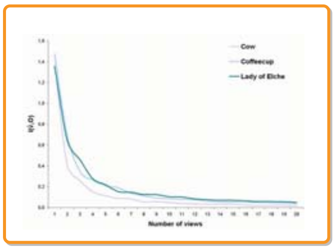


Viewpoint Information Channel

Selection of n Best Views

- **Objective:** to select the minimal set of representative views
- **Ideal proposal:** n views that maximize their JS (to capture the maximum information of the object)
- **Greedy strategy:** to select successive views that maximize JS





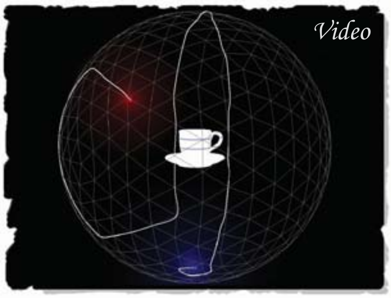
Viewpoint Information Channel

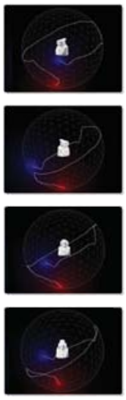
Viewpoint Clustering

- Clustering algorithm
  - Select the n best views
  - Assign each viewpoint to the ***nearest*** best viewpoint

Scene Exploration

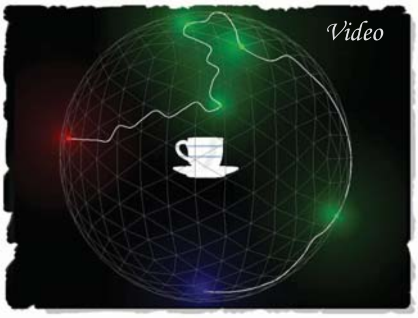
Exploratory Tour

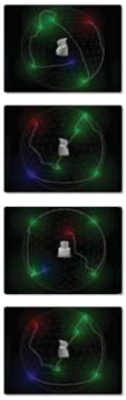




Scene Exploration

Guided Tour





Mesh Visibility

Reversion of the Channel

- Channel is reversed using the Bayes theorem

$$p(v,o) = p(v)p(o|v) = p(o)p(v|o)$$

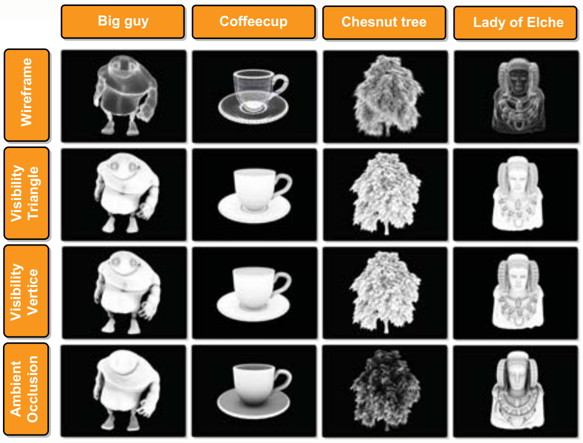
$$I(V,O) = \sum_{v\in\mathcal{V}} p(v) \sum_{o\in\mathcal{O}} p(o|v) \log \frac{p(o|v)}{p(o)}$$

$$\begin{aligned} I(V,O) &= \sum_{o\in\mathcal{O}} p(o) \sum_{v\in\mathcal{V}} p(v|o) \log \frac{p(v|o)}{p(v)} \\ &= \sum_{o\in\mathcal{O}} p(o) I(V,o), \end{aligned}$$

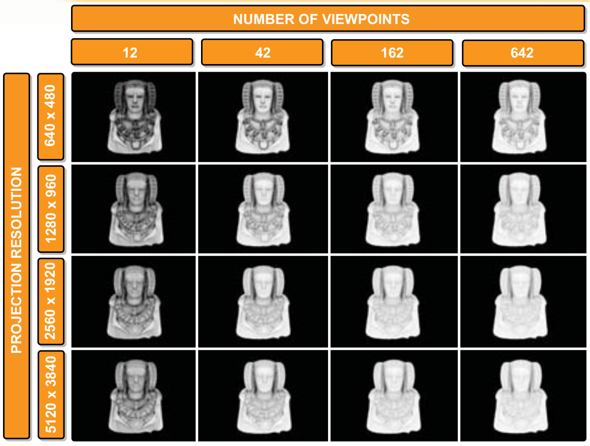
- *$I(V,o)$* is the polygonal mutual information
- Degree of correlation between the polygon o and the set of viewpoints

---

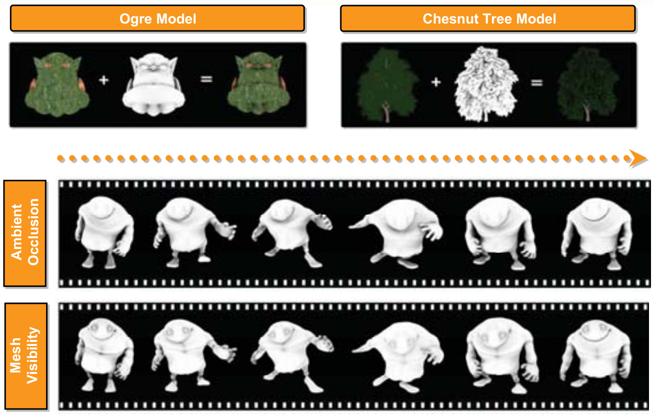
Mesh Visibility



Mesh Visibility



Mesh Visibility



Mesh Visibility

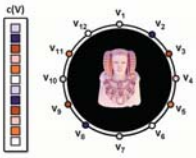
Mesh Visibility

Mesh Visibility

- **Applications**
  - Important viewpoints
    - Importance at the viewpoint space
    - Selection according to geometry and saliency

Mesh Visibility

- **Applications**
  - Relighting for *Non-Photorealistic* Rendering
    - Warping a color palette texture to the viewpoint sphere



    - Color ambient occlusion + *NPR* technique

Mesh Visibility

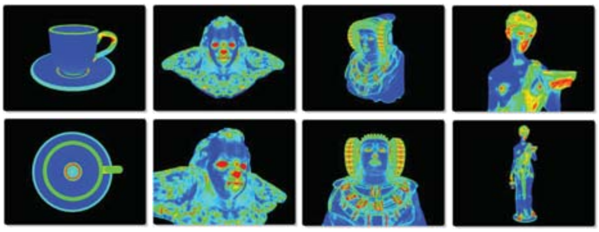
- **Applications**
  - Relighting *NPR* + *Coloroid Palettes*

Mesh Visibility

Mesh Saliency

$$S(o_i) = \frac{1}{N_o}\sum_{j=1}^{N_o} JS(p(V|o_i), p(V|o_j)) \geq 0,$$

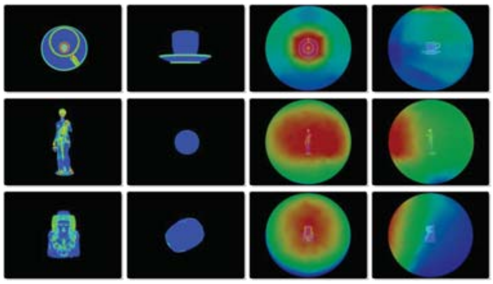
Coffeecup **Angel** **Lady of Elche** **Hebe**



Viewpoint Saliency

$$S(v) = \sum_{o\in\mathcal{O}} S(o)p(v|o)$$

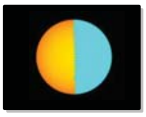
Most Salient **Least Salient** **Saliency Spheres**



Importance-based Viewpoint Mutual Information

Model

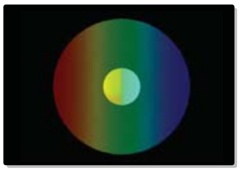
Importance Map



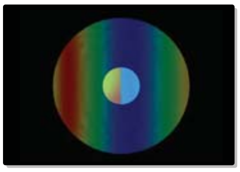
$$I'(v,O) = \sum_{o\in\mathcal{O}} p(o|v)\log\frac{p(o|v)}{p'(o)},$$

$$p'(o) = \frac{p(o)i(o)}{\sum_{o\in\mathcal{O}} p(o)i(o)}$$

Importance-based VMI Sphere

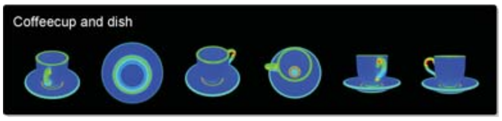


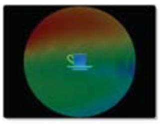
Importance-based VE Sphere

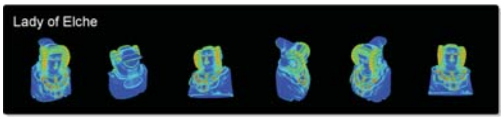


Importance-based Viewpoint Mutual Information

Saliency-based Best *N* Views **Saliency VMI Spheres**







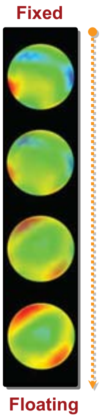
View-based Object Recognition

System features

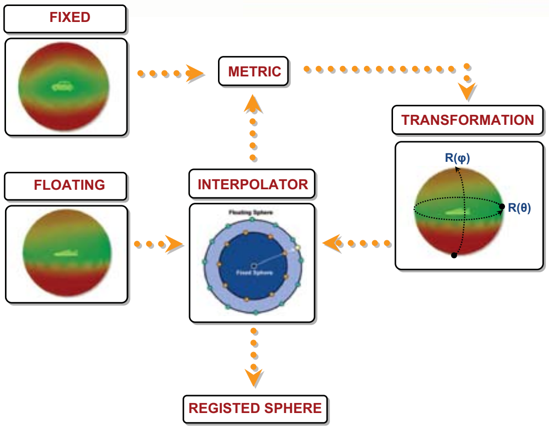
- VMI Sphere → View-based Shape descriptor
- Rigid registration system → Rotations (θ, φ)
- 642 viewpoints
- Fixed & Floating Sphere
- Metric

$$MSE(A,B)=\sqrt{\sum_{i=1}^{N}(a_i-b_i)^2}$$

- Interpolator → Nearest Neighbour



View-based Object Recognition


FIXED

METRIC

TRANSFORMATION

FLOATING

INTERPOLATOR

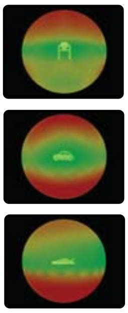
REGISTED SPHERE

View-based Object Recognition

Results

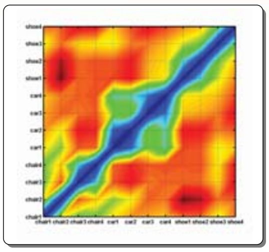
VMI Spheres

Models



View-based Object Recognition

Results



(6) View Selection in Scientific Visualization

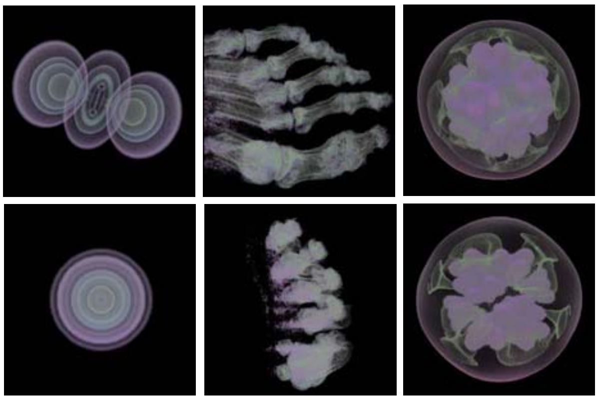
Ivan Viola
University of Bergen
Norway

View Selection for Volume Data

- Viewpoint quality = visibility of data
- Visibility computation
- Information-theoretic measures for characteristic viewpoint estimation
  - Viewpoint entropy
  - Mutual information
- View selection approaches for
  - 3D scalar fields
  - 3D + time scalar fields
  - Objects in volume data

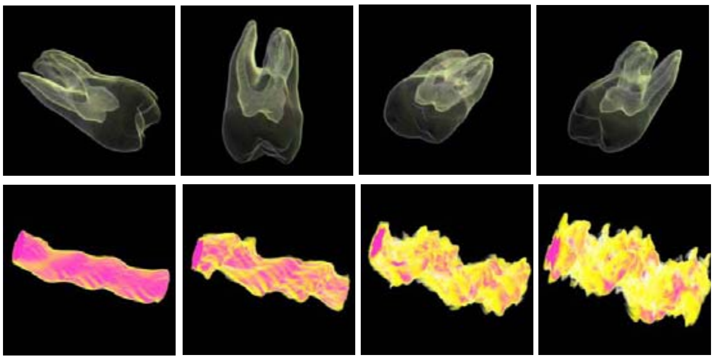
View Selection for Set of Iso-Surfaces

[Takahashi et al. Vis05]



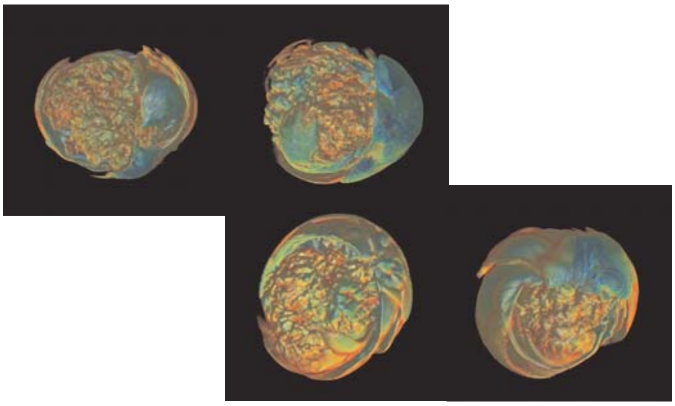
View Selection for Scalar Volumes (+ Time)

[Bordoloi and Shen Vis05]



Dynamic Views for Time-Varying Volumes

[Ji06 and Shen Vis06]



Focus of Attention

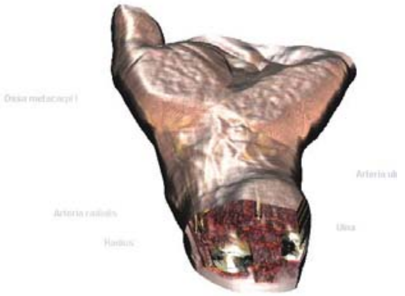
View Selection for Volumetric Objects

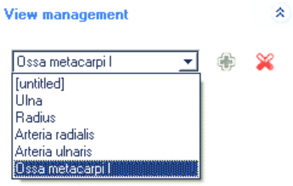
Focus of Attention

- Importance distribution among objects controls:
  - Characteristic view computation
  - Interactive focusing
- Characteristic view computation
  - View rating image and object weights
  - For every object + context
- Interactive focusing
  - Visual emphasis and cutaways
  - Changing the focus among objects

Goal

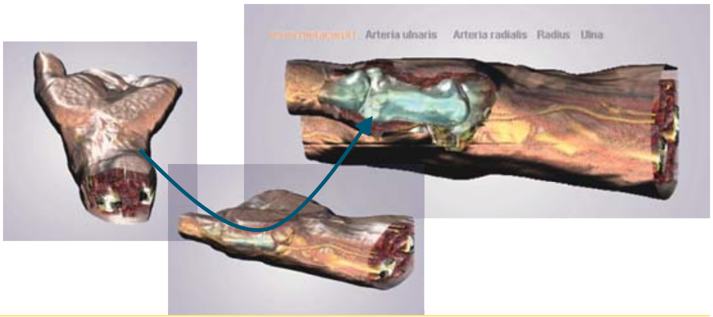
- Input: known and classified volumetric data
- High level request: show me object *X*
- Output: guided navigation to object *X*



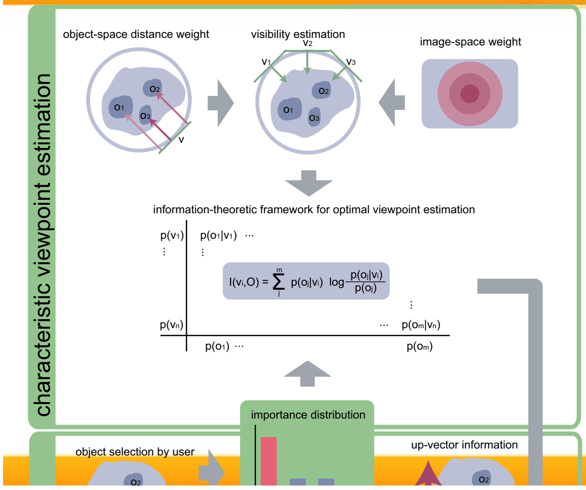


Focusing Considerations

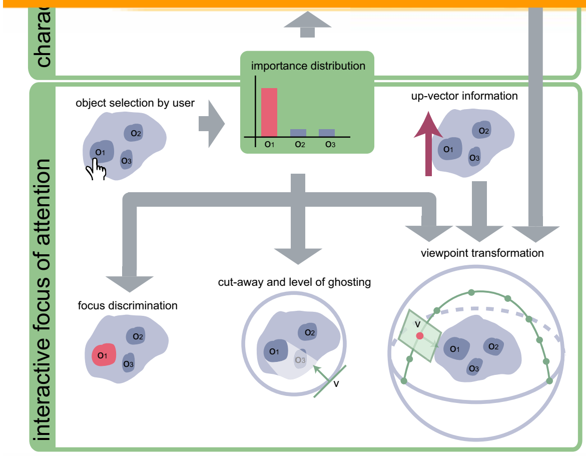
- Characteristic view
- Emphasis of focus object
- Guided navigation between characteristic views



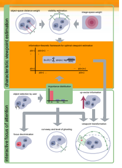
Framework



Framework

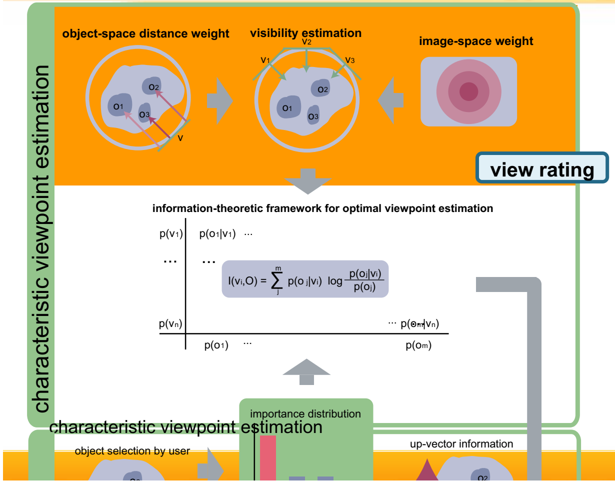


Characteristic Views

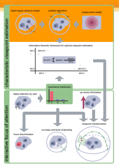


- Overview
  - All objects are visible
  - Visibility of objects is balanced
- Characteristic view of focus object
  - High visibility for focus object
  - If possible other objects also visible

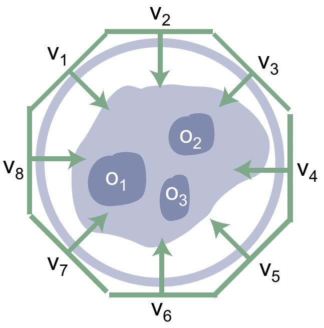
Characteristic View Estimation



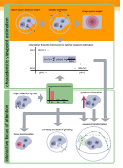
View rating



- For every view
  - For every object

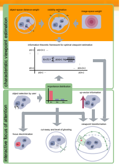


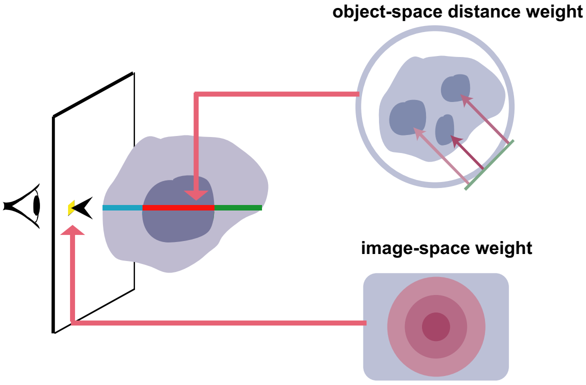
View Rating



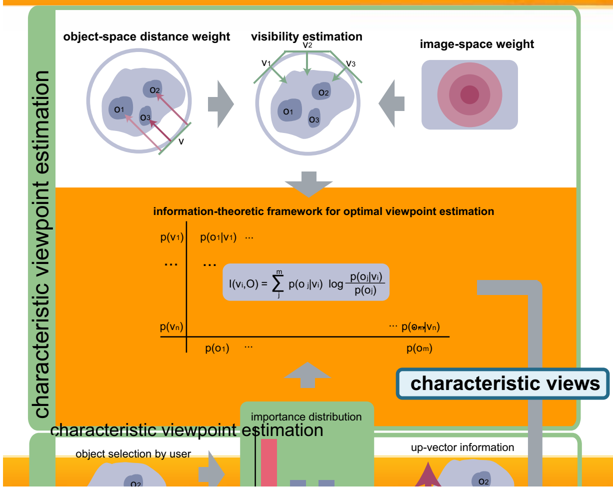
- Visibility
  - High
  - Low
- Location in image
  - In image center
  - Outside center
- Distance to the viewer
  - Object close to the viewer
  - Far from the viewer

View Rating Weights

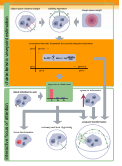




Characteristic Viewpoint Estimation

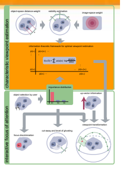


Characteristic Views



- Overview
  - All objects are visible
  - Visibility of objects is balanced
- Characteristic view of focus object
  - High view rating (visibility) for focus object
  - If possible other objects also visible

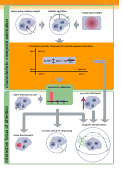
Obtaining Characteristic Views



- Sets of views and objects are random variables
  - Views V=(v_1, v_2, v_3, ... , v_n)
  - Objects O=(o_1, o_2, o_3, ... , o_m)
- View rating (visibility, weights)
  - Information channel between V→O
  - Conditional probability $p(o_j|v_i)$
- Mutual information between V and O expresses degree of dependance

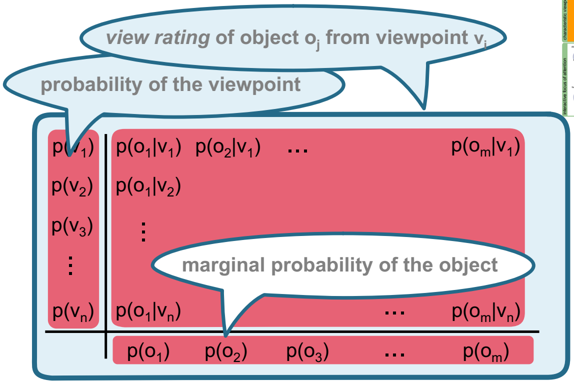
---

Obtaining Characteristic Views



- Viewpoint mutual information is dependance between v_i and O
  - High values = high dependance
    - Small number of objects
    - Low average visibility
  - Low values = low dependance
    - Maximum objects visible
    - Object visibility is balanced
- Minimal VMI determines the best view

Probability Transition Matrix

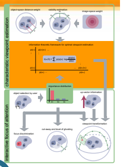


view rating of object o_j from viewpoint v_i

probability of the viewpoint

marginal probability of the object

$p(v_1)$	$p(o_1 \mid v_1)$	$p(o_2 \mid v_1)$	…		$p(o_m \mid v_1)$
$p(v_2)$	$p(o_1 \mid v_2)$				
$p(v_3)$	⋮				
⋮					
$p(v_n)$	$p(o_1 \mid v_n)$			…	$p(o_m \mid v_n)$
	$p(o_1)$	$p(o_2)$	$p(o_3)$	…	$p(o_m)$

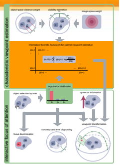


Viewpoint Mutual Information

- Degree of correlation $v_j \leftrightarrow O$

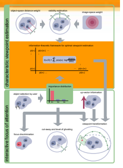
$$I(v_i,O)=\sum_j p(o_j \mid v_i)\log\frac{p(o_j \mid v_i)}{p(o_j)}$$

$p(v_1)$	$p(o_1 \mid v_1)$	$p(o_2 \mid v_1)$	…		$p(o_m \mid v_1)$
$p(v_2)$	$p(o_1 \mid v_2)$				
$p(v_3)$	⋮				
⋮					⋮
$p(v_n)$	$p(o_1 \mid v_n)$			…	$p(o_m \mid v_n)$
	$p(o_1)$	$p(o_2)$	$p(o_3)$	…	$p(o_m)$



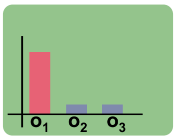
Characteristic Views

- Overview
  - All objects are visible
  - Visibility of objects is balanced
- Characteristic view at focus object
  - High view rating for focus object
  - If possible other objects also visible

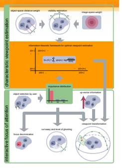


Incorporating Importance

importance distribution

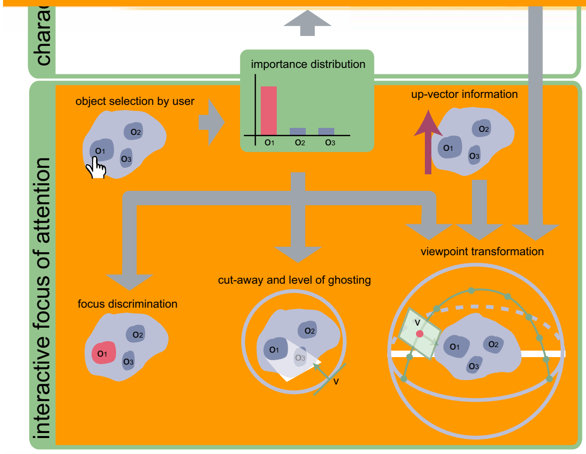


$$I(v_i,O)=\sum_j p(o_j \mid v_i)\log\frac{p(o_j \mid v_i)}{\frac{p(o_j)im(o_j)}{\sum_k p(o_k)im(o_k)}}$$



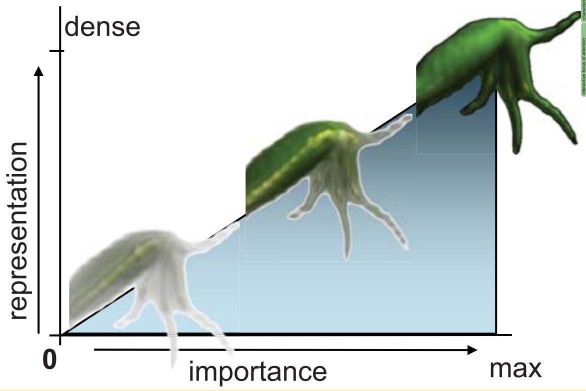
Resulting Characteristic Viewpoints

Interactive Focus of Attention



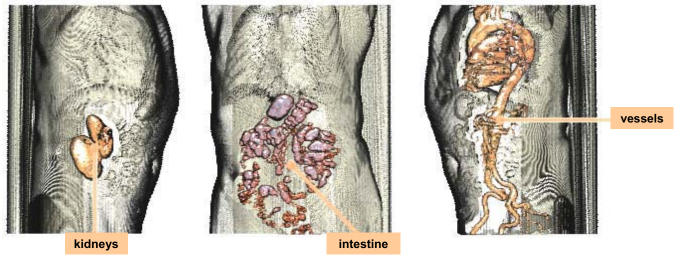
Emphasis of Focus Object

- Levels of sparseness



Emphasis of Focus Object

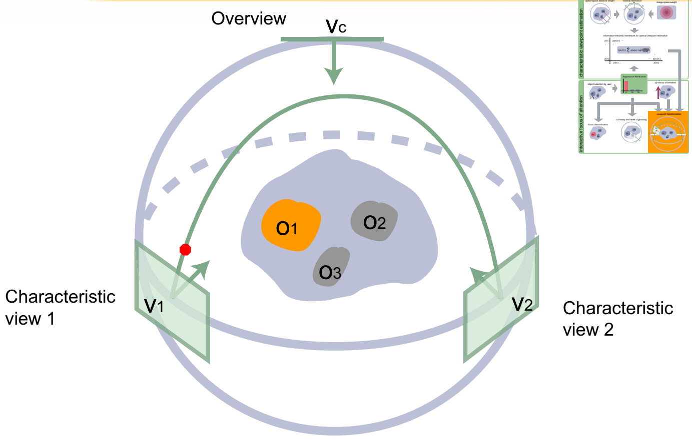
- Cut-aways to unveil internal features
- Labeling to add textual information



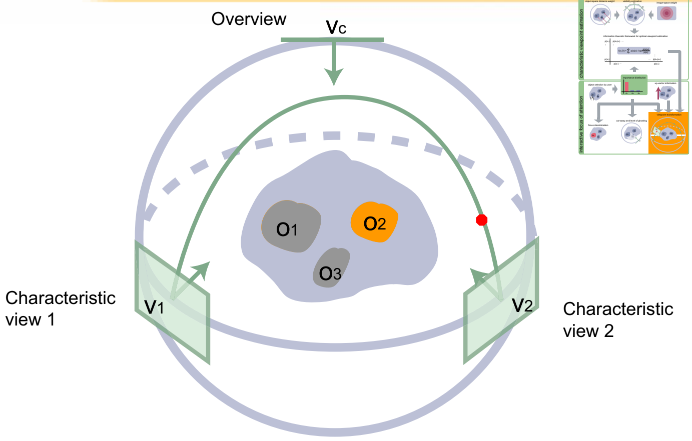
Guided Navigation Between Objects

- Decreasing importance of Object *X*
  - De-emphasis of Object *X*
  - Change to *overview*
- Increasing importance of Object *Y*
  - Emphasis of Object *Y*
  - Change to characteristic view of *Y*

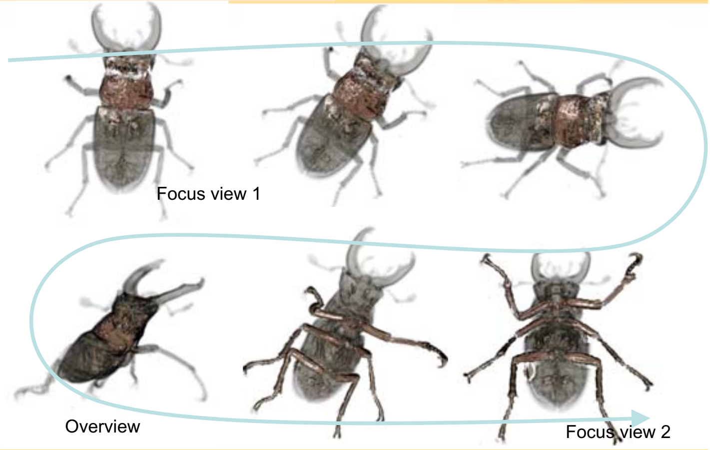
Refocusing



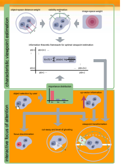
Refocusing



Example - Stagbeetle

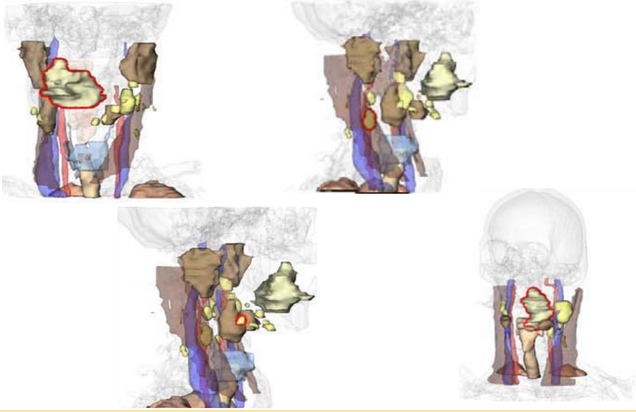


Example - Hand



Application-Driven View Selection

[Mühler et al. EuroVis07]



---

(7) Viewpoint-driven Simplification

Pascual Castelló
Miguel Chover

Universitat Jaume I

07 EUROGRAPHICS Prague Czech Republic

Introduction

- Most simplification methods use some geometric distance to guide the simplification process
- Recently, some works have developed methods guided by visual error metrics
- In some real-time applications like computer games the main requirement is visual similarity
- We propose new simplification metrics which produce closer approximations to the original model based on Information Theory

---

Recent Work on Simplification

- Geometry-Based
  - Appearance-Preserving Simplification [Cohen98]
  - Simplifying Surfaces with Color and Texture using Quadric Error Metrics [GH98]
  - New quadric metric for simplifying meshes with appearance attributes [Hoppe99]
  - Mesh Saliency [LVJ05]
- Viewpoint-Based
  - Image-Driven Simplification [LT00]
  - Perceptual-Driven Simplification for Interactive Rendering [LH01]
  - Visibility-Guided simplification [ZT02]
  - Viewpoint Entropy-driven Simplification [CSCF07]

Pros and Cons

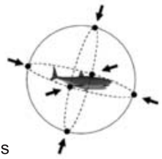
- Geometry- Based
  - The algorithm runs faster
  - Manage complex meshes
    - CAD, Scanned
  - Adjust geometric tolerance
- Viewpoint-Based
  - The algorithm runs slower
  - Deal with simple meshes
    - Games, Virtual Reality
  - Remove interior parts and preserves silhouette

"Current Game Artists make the simplifications by hand"

Viewpoint Entropy

- Definition
    - The Viewpoint Entropy gives a measure of the information provided by a point of view
    - We take as a probability distribution the relative area of the projected polygons over the sphere of directions centered in the viewpoint v

$$H_v = -\sum_{i=0}^{N_f} \frac{a_i}{a_t} \log \frac{a_i}{a_t}$$



- Where:
    - N_f: number of polygons in the scene
    - a_i: projected area of polygon i over the sphere
        - a_0: projected area of background in open scenes
        - a_t: total area of the sphere

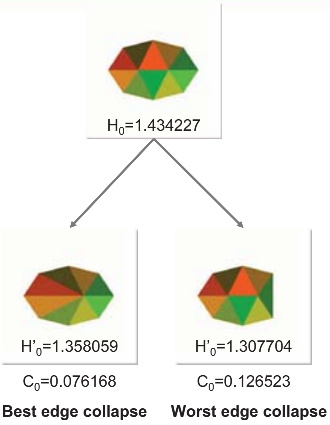
The best viewpoint is the one that has maximum entropy, i.e., maximum information captured

Viewpoint Entropy

- The error metric
    - Defined as the sum of variations of viewpoint entropy for all viewpoints V

$$c = \sum_{v \in V} \left| H_v - H'_v \right|$$

- Where:
    - H_v is the viewpoint entropy before an edge collapse
    - H'_v is the viewpoint entropy after an edge collapse



Simplification algorithm

```
/* Compute $I_v$ for the original mesh M */
Compute $I_v$ where $v=\{1,..,n\}$
/* Build initial heap of edge collapses */
for ( $e \in M$ )
  Perform collapse $e$
  Compute $I'_v$ where $v=\{1,..,n\}$
  Compute collapse cost $C_e$
  Insert $(e, C_e)$ in heap $h$
  Undo collapse $e$
end for
/* Update the mesh */
while ( heap $h$ not empty )
  Remove from heap $h$ the edge $e$ with lowest $C_e$
  Perform collapse $e$
  for ( each $e'$ in neighborhood )
    Compute collapse cost $C_{e'}$
    Update $(e', C_{e'})$  location in heap $h$
  end for
end while
```

Experiments

- Comparison
    - Algorithm
        - QSLIM v2.0 [Gar97] Well-Know geometric simplification algorithm
    - Tools
        - Geometric error: METRO v4.06 [Cig98]
        - Visual error: RMSE [Lin00]
- 20 viewpoints regularly distributed over a sphere
- Resolution: 256x256 images
- PC, Xeon 2.4 GHz, 1GB RAM, NVIDIA 7800 GTX 512MB
- C++ implementation with OpenGL
    - Vertex Buffer Objects & Frame Buffer Objects

Experiments H_V

Original Fish T=815

QSlim T=100

H_v C=20 T=100

Original Galleon T=4.698

QSlim T=500

H_v C=20 T=500

Experiments H_V

Original Galo T=6.592

QSlim T=500

H_v C=20 T=500

Original Octopus T=8.468

QSlim T=1.000

H_v C=20 T=1.000

Experiments H_V

Original Porsche T=10.474

QSlim T=1.000

H_v C=20 T=1.000

Original Unicycle T=13.810

QSlim T=1.000

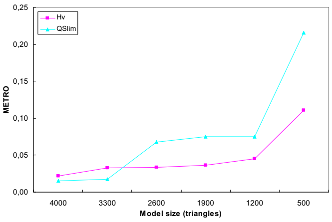
H_v C=20 T=1.000

Experiments H_V

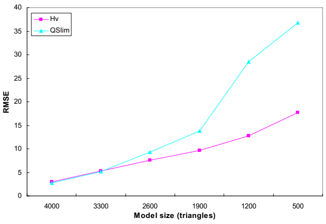
Model	Triangles		RMSE Error		METRO Error		Time	
	Original	Final	QSlim	H_v	QSlim	H_v	QSlim	H_v
Fish	815	100	22,83	**11,40**	0,09	**0,05**	0.02	11.16
Galleon	4.698	500	36,84	**17,74**	0,22	**0,11**	0.06	92.64
Galo	6.592	500	12,40	**9,03**	0,12	**0,08**	0.08	152.29
Octopus	8.468	500	25,84	**17,35**	0,05	**0,03**	0.09	224.89
Porsche	10.474	1.000	8,28	**7,48**	0,16	**0,09**	0.13	299.47
Unicycle	13.810	1.000	11,06	**10,32**	0,10	**0,04**	0.20	451.76

Experiments H_V

Geometric Error



Visual Error



Comparison at several degrees of simplification of the **Galleon model**

Mutual Information

- Definition
  - The Viewpoint Mutual Information defines an information channel between V and O

$$p(v) = \frac{1}{N_f}$$

$$p(o) = \sum_{v \in V} p(v) p(o \mid v) = \frac{1}{N_f} \sum_{v \in V} p(o \mid v)$$

  - The conditional probabilities of $p(o|v)$ are given by the relative area of the projected polygons over the sphere of directions centred at viewpoint v

$$I(V,O) = \sum_{v \in V} p(v) \sum_{o \in O} p(o \mid v) \log \frac{p(o \mid v)}{p(o)} = \frac{1}{N_v} \sum_{v \in V} I(v,O)$$

Mutual Information

- The mutual information for a given viewpoint

$$I(v,O) = \sum_{o \in O} p(o \mid v) \log \frac{p(o \mid v)}{p(o)}$$

- High values mean high degree of dependence “highly coupled view”
- Low values correspond to low dependence “more representative view”
- Observe that

$$I(v,O) = KL\big((p(O \mid v) \mid p(O)\big)$$

Mutual Information

- The error metric
  - Defined as the sum of variations of viewpoint mutual information for all viewpoints V

$$c = \sum_{v \in V} \left| I_v - I'_v \right|$$

Model	Triangles		RMSE Error		Time	
	Original	Final	H_v	VMI	H_v	VMI
Shark	734	80	14,78	**14,65**	10,24	10,23
Galo	6592	500	9,05	**8,38**	141,75	142,24
Greekship	9510	600	13,37	**12,85**	241,78	246,72
Tree	11.136	600	17,23	**16,60**	321,06	332,49
Hammer	13.380	500	8,13	**7,43**	404,33	423,05
Elephant	31.548	900	13,75	**11,60**	2197,67	2309,79

Experiments VMI


Original Shark T=734


QSlim T=80


VMI C=20 T=80


Original Galo T=6.592


QSlim T=500


VMI C=20 T=500

Experiments VMI


Original Greekship T=9.510


QSlim T=600


VMI C=20 T=600


Original Tree T=11.136


QSlim T=600

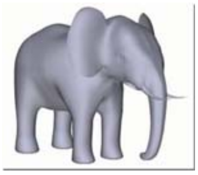

VMI C=20 T=600

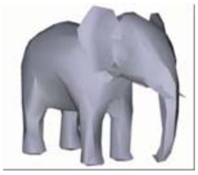
Experiments VMI

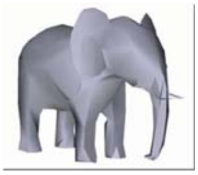

Original Hammer T=13.380


QSlim T=500

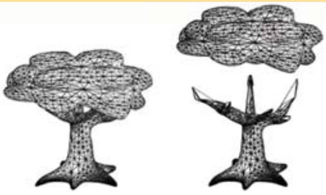

VMI C=20 T=500


Original Elephant T=31.548


QSlim T=900

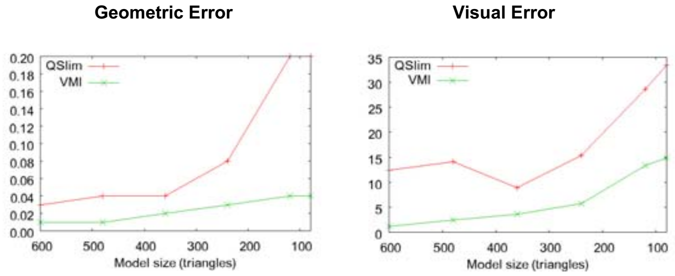

VMI C=20 T=900

Experiments VMI



Model	Triangles		RMSE		Metro		Time	
	Original	Final	QSlim	VMI	QSlim	VMI	QSlim	VMI
Shark	734	80	33,41	**14,65**	0,20	**0,04**	0,02	10,20
Galo	6592	500	12,40	**8,38**	0,05	**0,01**	0,08	142,24
Greekship	9510	600	17,20	**12,85**	0,21	**0,09**	0,11	246,72
Tree	11.136	600	20,73	**16,60**	**0,11**	0,13	0,20	332,49
Hammer	13.380	500	8,99	**7,43**	**0,03**	0,04	0,20	423,05
Elephant	31.548	900	25,32	**11,60**	0,08	**0,03**	0,52	2309,79

Experiments VMI


Geometric Error

Visual Error

Comparison at several degrees of simplification of the **Shark model**

Kullback-Leibler

- Definition
  - The f-divergences quantifying the degree of discrimination between two probability distributions
  - Kullback-Leibler distance

$$\boxed{KL(p\,|\,q)=\sum_{x\in X}p(x)\log\frac{p(x)}{q(x)}}$$

  - Viewpoint Kullback-Leibler distance

$$\boxed{KL_v=\sum_{i=1}^{N_f}\frac{a_i}{a_t}\log\left(\frac{\frac{a_i}{a_t}}{\frac{A_i}{A_T}}\right)} \qquad \boxed{A_T=\sum_{i=1}^{N_f}A_i} \qquad \boxed{a_t=\sum_{i=1}^{N_f}a_i}$$

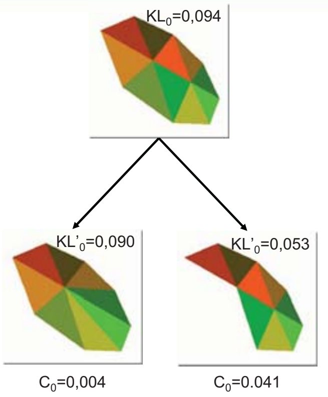
Where a_i is the projected area of the polygon i, A_i is the actual area of the polygon i and A_T is the total area of the object

Kullback-Leibler

- The error metric
  - Defined as the sum of variations of Kullback-Leibler distance for all viewpoints V

$$c = \sum_{v \in V} \left| KL_v - KL'_v \right|$$

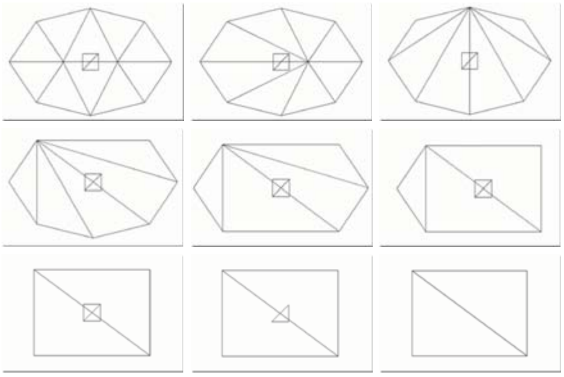
  - The cost of the algorithm is higher than Entropy or Mutual Information due to the A_T computation
  - Hidden polygons will be removed according with their actual area



Best edge collapse **Worst edge collapse**

Kullback-Leibler

- Simplification example using KL_v

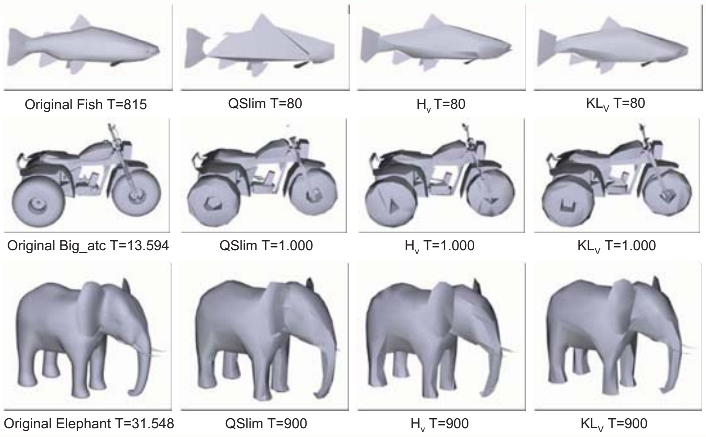


Experiments KL_V

Model	Triangles		RMSE			Metro		
	Original	Final	QSlim	H_v	KL_v	QSlim	H_v	KL_v
Fish	815	100	22.83	**11.57**	12.98	0.09	**0.03**	**0.03**
Galo	6.592	500	12.40	**9.34**	10.48	0.05	0.03	**0.01**
Al Capone	7.124	1.000	17.66	**11.47**	12.07	0.03	0.08	**0.03**
Tree	11.136	600	20.73	**16.98**	18.04	0.11	0.13	**0.04**
Big_atc	13.594	1000	16.50	15.97	**15.44**	0.08	0.05	**0.03**
Elephant	31.548	900	25.32	**13.18**	13.40	0.08	0.14	**0.05**

Model	Time		
Original	QSlim	H_v	KL_v
Fish	0.03	10.01	11.31
Galo	0.08	141.75	237.30
Al	0.08	150.90	273.18
Simpletree	0.20	332.49	605.49
Big_atc	0.27	535.23	835.88
Elephant	0.52	2197.67	4016.78

Experiments KL_V



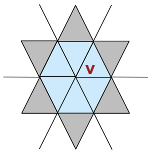
Original Fish T=815 QSlim T=80 H_v T=80 KL_V T=80

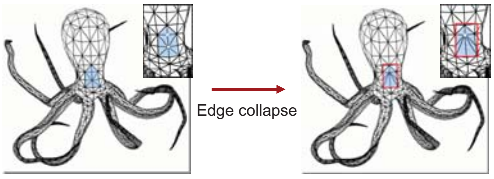
Original Big_atc T=13.594 QSlim T=1.000 H_v T=1.000 KL_V T=1.000

Original Elephant T=31.548 QSlim T=900 H_v T=900 KL_V T=900

Simplification Algorithm

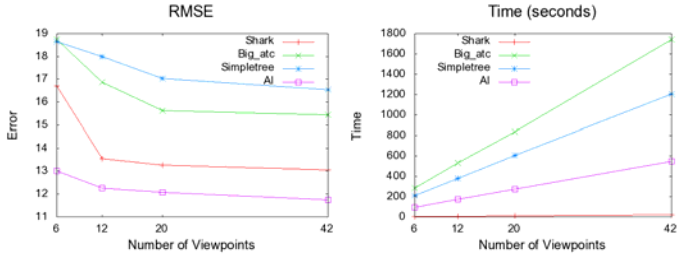
```
/* Update the mesh */
while ( heap $h$ not empty )
  Remove from heap $h$ the edge $e$ with lowest $C_e$
  Perform collapse $e$
  for ( each $e'$ in neighborhood )
    Compute collapse cost $C_{e'}$
    Update $(e', C_{e'})$ location in heap $h$
  end for
end while
```





Simplification Algorithm

- Analysis on the number of cameras using Mutual Information



Conclusions and future work

- New viewpoint-driven simplification metrics based on Information Theory has been proposed
- The metrics will be improved incorporating attributes (textures)
- We are working to reduce the computation time, although the simplification is an off-line process