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

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

- Shannon entropy of a discrete random variable $X$:
  - $H(X) = -\sum_{i=1}^{n} p_i \log p_i$

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

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

Information Channel

- Information channel
  - $X \rightarrow Y$
  - $H(Y|X) = -\sum_{i=1}^{n} \sum_{j=1}^{m} p_{ij} \log p_{ij}$
  - $H(X,Y) = -\sum_{i=1}^{n} \sum_{j=1}^{m} p_{ij} \log p_{ij}$
  - $I(X;Y) = H(X) - H(Y|X)$

- Conditional entropy
- Joint entropy
- Mutual information: dependence, correlation, shared information
**Inequalities**

- **Jensen’s inequality**: if $f(x)$ is a convex function
  \[
  f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)]
  \]

- **Log-sum inequality**
  \[
  \sum_{i=1}^{n} a_i \log \frac{a_i}{b_i} \geq \left( \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i} \right) \log \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i}
  \]

- **Data processing inequality**: if $X \rightarrow Y \rightarrow Z$ is a Markov chain, then
  \[
  I(X,Y) \geq I(X,Z)
  \]

**Jensen-Shannon Divergence**

- **Jensen-Shannon divergence**
  \[
  JS(\pi, \ldots, \pi; p_1, \ldots, p_n) = \sum_{i=1}^{n} \pi_i D_{KL}(p_i \parallel q) - \sum_{i=1}^{n} \pi_i H(p_i)
  \]

- **Properties**
  - Concavity of entropy: $JS(\pi, \ldots, \pi; p_1, \ldots, p_n) \geq 0$
  - $JS(p(x_1), \ldots, p(x_n); p(y | x_1), \ldots, p(y | x_n)) = I(X,Y)$

**Continuous Channel**

- **Continuous entropy**
  \[
  H'(X) = -\int p(x) \log p(x) dx
  \]
  \[
  \lim_{x \to 0} H'(X^2) = H'(X)
  \]

- **Continuous mutual information**
  \[
  I'(X,Y) = \int p(x,y) \log \frac{p(x,y)}{p(x)p(y)}
  \]
  \[
  \lim_{x \to 0} I'(X^2,Y^2) = I'(X,Y)
  \]

- **$F(X,Y)$ 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 $Y$.
- $I(X,Y)$
- $p(\hat{x} | x) \quad \hat{X} \quad p(\hat{x})
- I(\hat{X}, Y)$
- $p(y | \hat{x})$

**Generalised Entropy**

- Harvda-Charvát-Tsallis entropy (HCT)
- $H_\alpha(x) = \frac{1}{1-\alpha} \left(1 - \frac{1}{k} \sum_{i=1}^{k} p_i^\alpha \right)$
- $H_\alpha(x) = \lim_{\alpha \to 0} H_\alpha(x)$
- Generalised mutual information
- $I_\alpha(X,Y) = \frac{1}{1-\alpha} \left(1 - \frac{1}{k} \sum_{i=1}^{k} \sum_{j=1}^{k} p_{ij}^{\alpha} \right)$

**Radiosity Method**

- The radiosity method solves the problem of illumination in an environment of diffuse surfaces
- Continuous radiosity equation
- $B(x) = E(x) + \rho(x) \int F(x,y)B(y)\,dy$
- $F(x,y) = \frac{\cos \theta_x \cos \theta_y}{\pi r_{xy}^2}V(x,y)$

**Agglomerative IBM**

- Goal: find a clustering that minimizes the loss of mutual information
- Clustering or merging: loss of mutual information
- $I(X,Y) - I(\hat{X}, Y)$
- $p(\hat{x})JS(p(\hat{x}), ..., p(\hat{x}_n); p(y | \hat{x}_1), ..., p(y | \hat{x}_n))$
- $p(\hat{x}) = \sum_{i=1}^{n} p(\hat{x}_i)$
- The quality of each cluster $\hat{x}$ is measured by the Jensen-Shannon divergence between the individual distributions in the cluster.

**Radiosity Method**

- Discrete radiosity equation
- $B_i = E_i + \rho \sum_{j} F_{ij} B_j$
- $F_{ij} = \frac{1}{A_i} \int_{A_i} \int_{A_j} F(x,y)\,dA_i dA_j$
- Form factor properties
- Reciprocity
- Energy conservation
- $\sum_{i} F_{ij} = 1$
**Form Factor Computation**

- Analytical solutions
  - Between two spherical patches
    \[ F_{ij} = \frac{A_i}{A_j} \]

- Monte Carlo computation
  - Uniform area sampling
    \[ \hat{F}_{ij} = \frac{1}{N} \sum_{k} F(x_k, y_k) \]
  - Uniformly distributed lines

**Scene Information Channel**

- The scene is modelled as an information channel
  \[ \begin{array}{c|c|c}
  X & p_j | i & Y \\
  \hline
  p_i & \multicolumn{2}{c}{q_j} \\
  \hline
  F_{ij} & a_i \cdot F_{ij} & a_j
  \end{array} \]

**Continuous Mutual Information**

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

\[ I_m = \int_{x,y} \frac{1}{A_T} F(x, y) \log(A_T F(x, y)) \, dx \, dy \]

**Refinement Criteria for HR**

- In hierarchical radiosity (HR), the mesh is generated adaptively
- Oracles based on
  - Transformed power
  - Kernel smoothness
  \[ \rho \cdot A_i \cdot F_{ij} \cdot B_j < \varepsilon \]

**Scene Information Channel**

- Positional entropy
  \[ H_x = -\sum_{i} a_i \log a_i \]

- Scene entropy
  \[ H_s = -\sum_{i} \sum_{j} a_i F_{ij} \log F_{ij} \]

- Scene mutual information
  \[ I_s = H_x - H_s = \sum_{i} \sum_{j} a_i F_{ij} \log \frac{F_{ij}}{a_i} \]

**Monte Carlo Computation**

- Total area = \( A_T \)
- Lines cast = \( K \)
- Line segments = \( N \)

\[ I^c_j = \frac{1}{N} \sum_{x,y} \log \left( \frac{A_i \cdot \cos \theta_i \cdot \cos \theta_j}{A_T} \right) \]

contribution of each segment
**Discretisation Error**

- Two basic results
  - If any patch is subdivided, $I_s$ increases or remains the same
  - $I_s'$ is the least upper bound to $I_s$

- Discretisation error $I_s - I_s' > 0$

  $I_s = 0.690$
  $I_s = 2.199$
  $I_s = 2.558$
  $I_s = 2.752$

**Discretisation Error Between Two Patches**

- Discretisation error between two elements: loss of information transfer

  $\delta_{ij} = I_{ij} - I_{ij}'$

  Monte Carlo integration

  $\delta_{ij} = \frac{AA}{k} \left[ \frac{1}{N_y} \sum_{x=1}^{N_x} F(x, y_j) \log F(x, y_j) \right] \geq 0$

**Information Transfer**

- Mutual information matrix

  $I_{ij} = \sum_{x=1}^{N_y} \sum_{y=1}^{N_x} a_{ij} F_i(x, y) \log \frac{F_i(x, y)}{a_{ij}}$

  Information transfer from patch $i$

  $F_i = \sum_{y=1}^{N_y} \sum_{x=1}^{N_x} \int_{x=1}^{N_y} \int_{y=1}^{N_x} F(x, y) \log a_i(x, y) dx dy$

**MI-based Oracle**

- From radiosity equation and kernel-smoothness-based oracle

  $B_j = E_i + \sum_{j=1}^{N_y} \rho F_i B_j^*$

  $\rho \max(F_{ij}^*, F_{ij}^*, F_{ij}^*, F_{ij}^*) B_j < \epsilon$

- To MI-based oracle

  $\rho (I_{ij} - I_{ij}') B_j = \rho \delta_{ij} B_j < \epsilon$

**Oracles for HR**

- Kernel-smoothness-based

- MI-based

**MI-based Oracle for HR**

2684000 rays - 19000 patches - 10 lines FF
Generalised MI-based Oracle

\( D = 0.50 \) - 10 lines FF - 2684000 rays - 19000 patches

Generalised MI-based Oracle

\( D = 0.50 \) - 10 lines FF - 9268000 rays - 10000 patches

\( \chi^2 \)-Divergence-based Oracles

Kernel-Smoothness

Chi-Square

Hellinger

Adaptive Sampling

- Adaptive control of the sampling rate
- Image-Space
  - Intensity Comparison
  - Intensity Statistics
- Object-Space
  - \( f = 1 \frac{dI}{dI_{\text{max}}} \)  
    - [Simmons and Séquin, 00]
- Hybrid (image+object spaces)

Pixel Measures

Point-sampling-based technique for image synthesis

Capture the pixel radiance

Finite set of samples

Information is lost

Information measure

Erroneous information

Adaptive sampling

Regions with high inhomogeneity illumination

Measure

More samples!
**Pixel Colour Quality**

- **For each channel** $p_i = \text{colour fraction of a ray}$

  - **pixel channel entropy** $H^c = -\sum_{i=1}^{n_c} p_i \log p_i$
  - **pixel channel quality** $Q^c = \frac{H^c}{\log N_c}$
  - **pixel colour quality** $Q^c = \sum_{i=1}^{n_c} w_i Q_i$

**Channel perception coefficient**

**Colour system**

**Number of samples**

**Pixel Geometry Contrast**

- **For each channel** $p_i^g = \text{geometric fraction of a ray}$

  - **pixel geometric entropy** $H^g = -\sum_{i=1}^{n_g} p_i^g \log p_i^g$
  - **pixel geometric quality** $Q^g = \frac{H^g}{\log N_g}$
  - **pixel geometric contrast** $C^g = 1 - Q^g$

**Combination coefficient**

**Combination of colour and geometry**

**Pixel Colour Contrast**

- **For each channel** $p_i^c = \text{colour fraction of a ray}$

  - **pixel channel contrast** $C^c = 1 - Q^c$
  - **pixel colour contrast** $C^c = \frac{\sum_{i=1}^{n_c} w_i C_i^c}{\sum_{i=1}^{n_c} w_i}$

**Pixel channel colour average**

**Quality Map**

- **Map of geometric quality**
- **Map of colour quality**

**Supersampling**

- **Uniform with 32 rays per pixel**
- **Average rays per pixel: 32**
Entropy-based Adaptive Sampling

Grouping property of Entropy

\[ H(X) = -\sum_{i=1}^{q_i} q_i \log q_i \]

{\text{information acquired}}

\[ H(Y_i) \]

{\text{hidden information}}

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

The decomposition of \( H \) can be recursively extended to the subpixels

Results

Classic contrast

Variance-based contrast

Entropy-based contrast

Results

\[ D_f(p,q) = \sum_{x} q(x) f\left( \frac{p(x)}{q(x)} \right) \]

\( f \)-Divergences as refinement criteria in RT?

- Distributions
  - \( \{p\} \) = Luminance \( L \) of \( N_S \)-samples
  - \( \{q\} \) = Uniform \( 1/N_S \)

- Homogeneity: \( D_f(p,q) \)
- Importance: \( \text{avg}(L_i) \)
- Convergence: \( 1/N_S \)

\( f \)-Divergence-based Adaptive Sampling

\[ p_n = \frac{1}{N_S} \sum_{i=1}^{N_S} p_i \]

\[ q_n = \frac{1}{N_S} \sum_{i=1}^{N_S} q_i \]

Kullback-Leibler

Chi-Square

Hellinger
Introduction

- Viewpoint selection is a new area in computer graphics with applications in fields such as scene understanding, volume visualization, image-based modeling, and molecular visualization.

- We propose a unified framework for viewpoint selection and mesh visibility/saliency/simplification based on an information channel between the 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).

- 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
  
  - Information Channel
    \[ X \rightarrow Y \frac{p(y|x)}{p(x)} \]
  
  - Conditional Entropy
    \[ H(Y|X) = -\sum_{x,y} p(x,y) \log p(y|x) \]

  - Mutual Information
    \[ I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \]

  - Jensen-Shannon inequality
    \[ JSE(p_1, p_2, \ldots, p_N) = H(\sum_{i=1}^{N} \pi_i p_i) - \sum_{i=1}^{N} \pi_i H(p_i) \geq 0 \]

- Related Work
  
  - Heuristic measure: Plemenos et al. [1996]
  
  - Viewpoint Entropy
    \[ C(v) = \sum_{i=1}^{n} \left( \frac{p_i}{p_{\text{avg}}} \right) + \sum_{i=1}^{n} \frac{P_i}{P_{\text{avg}}} \]

  - Kullback-Leibler distance
    \[ KL(v) = \sum_{i=1}^{N} \frac{a_i}{b_i} \log \frac{a_i}{b_i} \]

- Origins: Rigau et. al [2000], Vázquez et al. [2001-2006], Sbert. Et al [2005]
We formalize the viewpoint selection using an information channel

\[
P(o) = \sum_{v \in V} p(v) p(o|v)
\]

This framework is based on geometric characteristics

Viewpoint Mutual Information evaluation (I)

- **Best View**
- **Worst View**
- **Spheres**

Viewpoint Similarity and Unstability

**Viewpoint Similarity**

Any clustering over \( V \to \hat{V} \) or \( O \to \hat{O} \) reduce \( I(V,O) \)

\[
\Delta I(v_i, v_j) = I(V, O) - I(\hat{V}, \hat{O}) = p(v_i) p(o_i|v_i) - p(\hat{v}_i) p(\hat{o}_i|\hat{v}_i)
\]

\[
\Delta D(v_i, v_j) = D(v_i, v_j) = D(O|v_i, v_j)
\]

Viewpoint Mutual Information evaluation (II)

- **Model**
- **Heuristic**
- **Entropy**
- **VMI**

Viewpoint Similarity and Unstability

**Viewpoint Unstability**

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

\[
U(v_i) = \frac{1}{N_v} \sum_{j=1}^{N_v} D(v_i, v_j)
\]
**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 $JS$

![Image of view selection](image)

**Viewpoint Clustering**

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

![Image of view clustering](image)

**Scene Exploration**

- **Exploratory Tour**

![Image of exploratory tour](image)

- **Guided Tour**

![Image of guided tour](image)

**Mesh Visibility**

- **Reversion of the Channel**
  - Channel is reversed using the Bayes theorem

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

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

![Image of mesh visibility](image)
**Mesh Visibility**

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

**Demo**

- **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 Saliency

\[ S(\omega) = \frac{1}{N_\omega} \sum_{i=1}^{N_\omega} J(\omega, \omega_i) J(\omega, \omega_i) \geq 0. \]

Viewpoint Saliency

\[ S(v) = \sum_{v \in \mathcal{V}} S(\omega)p(v|\omega) \]

Importance-based Viewpoint Mutual Information

\[
I(\omega, \mathcal{O}) = \sum_{v \in \mathcal{O}} p(\omega|v) \log \frac{p(\omega|v)}{p(\omega)} ,
\]

\[
p'(\omega) = \frac{p(\omega|v)}{\sum_{v \in \mathcal{O}} p(\omega|v)}
\]
View-based Object Recognition

**System features**
- VMI Sphere: View-based Shape descriptor
- Rigid registration system: Rotations ($\theta, \phi$)
- 642 viewpoints
- Fixed & Floating Sphere
- Metric
  
  \[ \text{MSE}(A, B) = \sum_{i=1}^{N} (a_i - b_i)^2 \]
- Interpolator: Nearest Neighbour

**Fixed**

**Floating**

View-based Object Recognition

**Results**

VMI Spheres

Models

---

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
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

- Interactive focus of attention
- Object selection by user
- Viewpoint transformation
- Cut-away and level of ghosting
- Object-space distance weight
- Visibility estimation
- Image-space weight
- Characteristic viewpoint estimation
- Interactive view of attention
- Focus discrimination
- Object selection by user
- Up-vector information

Characteristic View Estimation

- Information-theoretic framework for optimal viewpoint estimation
- For every view
- For every object

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

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

Characteristic Viewpoint Estimation
- Sets of views and objects are random variables
  - Views \( V = (v_1, v_2, v_3, \ldots, v_n) \)
  - Objects \( O = (o_1, o_2, o_3, \ldots, o_m) \)
- View rating (visibility, weights)
  - Information channel between \( V \rightarrow O \)
  - Conditional probability \( p(o_j|v_i) \)
- Mutual information between \( V \) and \( O \) expresses degree of dependence

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

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
**Probability Transition Matrix**

- Probability of the viewpoint
- Marginal probability of the object
- View rating of object \( o_j \) from viewpoint \( v_i \)

**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

**Resulting Characteristic Viewpoints**

- Interactive focus of attention
  - Characteristic viewpoint estimation
  - Focus discrimination
  - Cut-away and level of ghosting
  - Object selection by user
  - Importance distribution
  - Viewpoint transformation

**Viewpoint Mutual Information**

- Degree of correlation \( v_i \rightarrow O \)

\[
I(v_i, O) = \sum_j p(o_j | v_i) \log \frac{p(o_j | v_i)}{p(o_j)}
\]

**Incorporating Importance**

- Importance distribution
  \[
  I(v_i, O) = \sum_j p(o_j | v_i) \log \frac{p(o_j | v_i) \operatorname{im}(o_j)}{\sum_k p(o_k) \operatorname{im}(o_k)}
  \]
Emphasis of Focus Object

- Levels of sparseness
  - dense
  - max

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 Diagram](image)

Example - Stag beetle

![Stag beetle Diagram](image)
**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]

**Application-Driven View Selection**

**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.

**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 $v$.

  \[ H_v = - \sum_{i=1}^{Nf} \frac{a_i}{a_0} \log \frac{a_i}{a_0} \]

- **Where:**
  - $N_f$: number of polygons in the scene
  - $a_i$: projected area of polygon $i$ over the sphere
  - $a_{bg}$: projected area of background in open scenes
  - $a_0$: total area of the sphere

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

**Simplification algorithm**

```c
/* Compute $I_v$ for the original mesh $M$ */
Compute $I_v$ where $v=\{1,..,n\}$

/* Build initial heap of edge collapses */
for (v \in M)
    Perform collapse $v$
    Compute $I_v'$ where $v=\{1,..,n\}$
    Compute collapse cost $C_v$
    Insert ($v$, $C_v$) in heap $h$
    Undo collapse $v$
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
```

**Viewpoint Entropy**

- **Error metric**
  - Defined as the sum of variations of viewpoint entropy for all viewpoints $v'$.

  \[ c = \sum_{v} |H_v - H_v'| \]

- **Where:**
  - $H_v$: viewpoint entropy before an edge collapse
  - $H_v'$: viewpoint entropy after an edge collapse

**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 $H_V$

<table>
<thead>
<tr>
<th>Model</th>
<th>Triangles</th>
<th>RMSE Error</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Original</td>
<td>Final</td>
<td>$H_V$</td>
</tr>
<tr>
<td>Fish</td>
<td>815</td>
<td>100</td>
<td>22.83</td>
</tr>
<tr>
<td>Galleon</td>
<td>4,698</td>
<td>500</td>
<td>36.84</td>
</tr>
<tr>
<td>Galo</td>
<td>6,592</td>
<td>500</td>
<td>12.40</td>
</tr>
<tr>
<td>Octopus</td>
<td>8,468</td>
<td>500</td>
<td>25.84</td>
</tr>
<tr>
<td>Porsche</td>
<td>10,474</td>
<td>1,000</td>
<td>8.28</td>
</tr>
<tr>
<td>Unicycle</td>
<td>13,810</td>
<td>1,000</td>
<td>11.06</td>
</tr>
</tbody>
</table>

Mutual Information

Definition

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

$$p(v) = \frac{1}{N} \sum_{o \in O} p(o | v)$$

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 $v$

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

The error metric

Defined as the sum of variations of viewpoint mutual information for all viewpoints $V$

$$c = \sum_{i=1}^{N} [I_i - I'_i]$$

Experiments $H_V$

Comparison at several degrees of simplification of the Galleon model

Mutual Information

The mutual information for a given viewpoint

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

High values mean high degree of dependence “highly coupled view”

Low values correspond to low dependence “more representative view”

Observe that

$$I(v, O) = K L(p(O | v) | p(O))$$
Experiments VMI

Original Shark T=734

QSim T=600

VMI C=20 T=400

Original Galo T=6,592

QSim T=500

VMI C=20 T=500

Original Hammer T=13,380

QSim T=600

VMI C=20 T=900

Original Elephant T=31,548

QSim T=900

VMI C=20 T=900

Geometric Error

Visual Error

Comparison at several degrees of simplification of the Shark model

Experiments VMI

Original Greekship T=9,510

QSim T=600

VMI C=20 T=600

Original Tree T=11,136

QSim T=600

VMI C=20 T=600

Experiments VMI

Kullback-Leibler

Definition

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

\[
KL(p \| q) = \sum_{x} p(x) \log \frac{p(x)}{q(x)}
\]

- Viewpoint Kullback-Leibler distance

\[
KL_v = \sum_{i=1}^{N_v} \log \frac{a_i}{A_i}
\]

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Triangles</th>
<th>RMSE</th>
<th>Metro</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shark</td>
<td>734</td>
<td>33,41</td>
<td>0,20</td>
<td>0,02</td>
</tr>
<tr>
<td>Galo</td>
<td>6592</td>
<td>12,40</td>
<td>0,05</td>
<td>0,08</td>
</tr>
<tr>
<td>Greekship</td>
<td>9510</td>
<td>17,20</td>
<td>0,21</td>
<td>0,11</td>
</tr>
<tr>
<td>Tree</td>
<td>11,136</td>
<td>20,73</td>
<td>0,11</td>
<td>0,20</td>
</tr>
<tr>
<td>Hammer</td>
<td>13,380</td>
<td>8,99</td>
<td>0,03</td>
<td>0,52</td>
</tr>
<tr>
<td>Elephant</td>
<td>31,548</td>
<td>25,32</td>
<td>0,08</td>
<td>0,03</td>
</tr>
</tbody>
</table>

Visual ErrorGeometric Error

Comparison at several degrees of simplification of the Shark model
**Kullback-Leibler**

- The error metric
  - Defined as the sum of variations of Kullback-Leibler distance for all viewpoints $v$
  \[ c = \sum_{v \in V} [KL_v - KL_v'] \]
  - The cost of the algorithm is higher than Entropy or Mutual Information due to the $AT$ computation
  - Hidden polygons will be removed according to their actual area

**Experiments $KL_v$**

<table>
<thead>
<tr>
<th>Model</th>
<th>Triangles</th>
<th>RMSE Original</th>
<th>RMSE Final</th>
<th>RMSE Metro</th>
<th>RMSE QSim</th>
<th>RMSE Hv</th>
<th>RMSE KLV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fish</td>
<td>815</td>
<td>22.83</td>
<td>11.57</td>
<td>12.98</td>
<td>0.09</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>Galo</td>
<td>6,592</td>
<td>12.40</td>
<td>9.34</td>
<td>10.48</td>
<td>0.05</td>
<td>0.03</td>
<td>0.01</td>
</tr>
<tr>
<td>Al Capone</td>
<td>7,124</td>
<td>17.66</td>
<td>11.47</td>
<td>12.07</td>
<td>0.03</td>
<td>0.08</td>
<td>0.03</td>
</tr>
<tr>
<td>Tree</td>
<td>11,136</td>
<td>20.73</td>
<td>16.96</td>
<td>18.04</td>
<td>0.11</td>
<td>0.13</td>
<td>0.04</td>
</tr>
<tr>
<td>Big, etc</td>
<td>13,594</td>
<td>16.50</td>
<td>15.97</td>
<td>15.44</td>
<td>0.08</td>
<td>0.05</td>
<td>0.03</td>
</tr>
<tr>
<td>Elephant</td>
<td>31,548</td>
<td>25.32</td>
<td>13.16</td>
<td>13.40</td>
<td>0.08</td>
<td>0.14</td>
<td>0.05</td>
</tr>
</tbody>
</table>

**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 $e$)

Compute collapse cost $C'_{e'}$

Update ($e'$, $C'_{e'}$) location in heap $h$

end for

end while

**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 $e$)

Compute collapse cost $C'_{e'}$

Update ($e'$, $C'_{e'}$) location in heap $h$

end for

end while

**Experiments $KL_v$**

- Analysis on the number of cameras using Mutual Information

**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 $e$)

Compute collapse cost $C'_{e'}$

Update ($e'$, $C'_{e'}$) location in heap $h$

end for

end while

**Experiments $KL_v$**

- Analysis on the number of cameras using Mutual Information

**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 $e$)

Compute collapse cost $C'_{e'}$

Update ($e'$, $C'_{e'}$) location in heap $h$

end for

end while

**Experiments $KL_v$**

- 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.