MORGAN & CLAYPOOL PUBLISHERS

Information Theory Tools for Computer Graphics

Mateu Sbert
Miquel Feixas
Jaume Rigau
Miguel Chover
Ivan Viola

SYNTHESIS LECTURES ON COMPUTER GRAPHICS AND ANIMATION

Brian A. Barsky, *Series Editor*

Information Theory Tools for Computer Graphics

Information Theory Tools for Computer Graphics

Mateu Sbert, Miquel Feixas, Jaume Rigau, Miguel Chover, and Ivan Viola

www.morganclaypool.com

ISBN: 9781598299298 paperback
ISBN: 9781598299304 ebook

DOI 10.2200/S00208ED1V01Y200909CGR012

A Publication in the Morgan & Claypool Publishers series
SYNTHESIS LECTURES ON COMPUTER GRAPHICS AND ANIMATION

Lecture #12
Series Editor: Brian A. Barsky, *University of California, Berkeley*

Series ISSN
Synthesis Lectures on Computer Graphics and Animation
Print 1933-8996 Electronic 1933-9003

Synthesis Lectures on Computer Graphics and Animation

Information Theory Tools for Computer Graphics

Mateu Sbert
University of Girona, Spain

Miquel Feixas
University of Girona, Spain

Jaume Rigau
University of Girona, Spain

Miguel Chover
Jaime I University, Spain

Ivan Viola
University of Bergen, Norway

SYNTHESIS LECTURES ON COMPUTER GRAPHICS AND ANIMATION #12

MORGAN & CLAYPOOL PUBLISHERS

ABSTRACT

Information theory (IT) tools, widely used in scientific fields such as engineering, physics, genetics, neuroscience, and many others, are also emerging as useful transversal tools in computer graphics. In this book, we present the basic concepts of IT and how they have been applied to the graphics areas of radiosity, adaptive ray-tracing, shape descriptors, viewpoint selection and saliency, scientific visualization, and geometry simplification. Some of the approaches presented, such as the viewpoint techniques, are now the state of the art in visualization. Almost all of the techniques presented in this book have been previously published in peer-reviewed conference proceedings or international journals. Here, we have stressed their common aspects and presented them in an unified way, so the reader can clearly see which problems IT tools can help solve, which specific tools to use, and how to apply them. A basic level of knowledge in computer graphics is required but basic concepts in IT are presented. The intended audiences are both students and practitioners of the fields above and related areas in computer graphics. In addition, IT practitioners will learn about these applications.

KEYWORDS

radiosity, ray-tracing, complexity, shape descriptors, viewpoint selection, scientific visualization, level of detail, geometry simplification, information theory, entropy, mutual information, Jensen–Shannon divergence, information bottleneck method, *f*-divergences

Contents

Preface

Information Theory (IT) tools, widely used in scientific fields such as engineering, physics, genetics, neuroscience, and many others, are emerging as useful transversal tools in computer graphics. In this book, we present the basic concepts of IT and how they have been applied to the graphics areas of radiosity, adaptive ray-tracing, shape descriptors, viewpoint selection and saliency, scientific visualization, and geometry simplification. Throughout the presentation of the applications, we aim to make clear problems that IT tools can help solve, which specific tools to use, and how to apply them. The reader is expected to have a basic background in computer graphics. IT basics are presented and self-contained in this book.

The intended audiences include students and practitioners of the fields above, as well as, related areas in computer graphics and IT. We believe that there will be interest in this book for several reasons. First, IT techniques are pervading more areas and are gradually gaining momentum as a technological tool. Second, since global illumination is increasingly used in production, more efficient techniques will be welcomed. Third, the medical imaging community is already using and relying on existing IT techniques and the growing importance of this community is influencing both computer graphics and image processing views. Finally, some of the approaches presented, such as the viewpoint techniques, are now the state of the art in visualization; receiving increased attention from practitioners.

This document is organized in the following way. After this preface, the first chapter deals with the basics of IT. The concept of information channel is introduced, and the quantities of entropy and mutual information are defined, together with important relationships such as the Jensen-Shannon inequality. The f-divergences are also introduced, and, finally, a basic idea of generalized entropy is given.

The second chapter deals with scene complexity measures and their application to radiosity. Radiosity is a viewpoint independent global illumination technique that discretises the scene into small polygons or patches to solve a transport system of equations. The way the scene is discretized is critical for the efficiency of the result. We define first a scene information channel, which allows us to study the interchange of information between the patches. From the study of this channel, several refinement oracles, i.e., criteria for subdividing the geometry, are obtained, aimed at maximizing the transport of information. Both classic and generalized information-theoretic quantities are used to these means.

Another application of scene complexity measures is presented in the third chapter, where different shape descriptors based on the complexity of the object are defined. Shape descriptors are important when classifying and retrieving objects from databases. Inner and outer complexity,

obtained from mutual information calculation with uniformly distributed lines, will be shown to classify different families of 2D and 3D objects.

The fourth chapter is about adaptive ray-tracing. This technique is aimed at tracing more rays only where they are needed. For instance, smoothly illuminated regions of the scene with low variation do not need as much effort as rapidly varying illumination or geometric discontinuities. IT quantities will be used again to define adaptive refinement criteria. In this chapter, new oracles are defined for radiosity, and adaptive ray-tracing using the Kullback–Leibler, Chi-square, and Hellinger f-divergences are discussed.

In the fifth chapter, we define a viewpoint information channel between the points of view around an object and the polygons of the object. Several quantities associated to this channel, such as mutual information and entropy, are interpreted in terms of viewpoint quality measures. Viewpoint similarity and stability are defined, as well as methods for the selection of best n-views and exploring the object. Mesh saliency is interpreted in terms of the viewpoint channel and polygonal mutual information as an ambient occlusion quantity. Importance is introduced into the scheme and saliency is used as an importance to guide the viewpoint selection.

The sixth chapter deals with view selection in scientific visualization. The problem and context of the visualization of volumetric data sets is presented, together with the different viewpoint quality measures that are used. The framework presented in Chapter five is then applied to this context. Guided navigation, using higher-level semantics, is also studied.

Finally, the seventh chapter is about viewpoint-driven simplification. Several simplification algorithms are based on the variations perceived in image space, measured in our case through information-theoretic metrics: entropy, Kullback-Leibler distance, and mutual information. These techniques are shown to give a better simplified mesh than object-based approaches, although at the cost of an increased processing time.

ACKNOWLEDGEMENTS

The authors wish to thank coauthors Esteve del Acebo, Anton Bardera, Philippe Bekaert, Imma Boada, Stefan Bruckner, Pascual Castelló, Fran González, Eddie Groeller, Dimitri Plemenos, Marc Ruiz, Pere-Pau Vázquez, and Qing Xu for discussions and producing some of the images used in this book. Finally, special thanks goes to Laszlo Szirmay-Kalos and Vlastimil Havran for their most useful comments on an early draft of this book. Thanks go also to the authors who have allowed the use of their images in this book.

Authors acknowledge support from grants TIN2007-68066-C04-01, TIN2007-68066-C04-02, from the Spanish government, and by the Norwegian Research Council (Project Nr: 193170/S10) and by the MedViz Initiative in Bergen (medviz.uib.no).

CHAPTER 1

Information Theory Basics

In 1948, Claude Shannon published a paper entitled "A mathematical theory of communication" [113] which marks the beginning of information theory. In this paper, Shannon defined measures such as entropy and mutual information[1], and introduced the fundamental laws of data compression and transmission. Information theory deals with the transmission, storage, and processing of information and is used in fields such as physics, computer science, mathematics, statistics, economics, biology, linguistics, neurology, learning, image processing, and computer graphics.

In information theory, *information* is simply the outcome of a selection among a finite number of possibilities, and an information source is modeled as a random variable or a random process. The classical measure of information, Shannon entropy, expresses the information content or the uncertainty of a single random variable. It is also a measure of the dispersion or diversity of a probability distribution of observed events. For two random variables, their mutual information is a measure of the dependence between them. Mutual information plays an important role in the study of a *communication channel*, a system in which the output depends probabilistically on its input [25, 139, 146].

This chapter presents Shannon's information measures (entropy, conditional entropy, and mutual information) and their most basic properties. The information bottleneck method, f-divergences, and generalized entropies are also introduced. Good references of information theory are the books by Cover and Thomas [25], and Yeung [146].

1.1 ENTROPY

After representing a discrete information source as a random process, Shannon asks himself: "Can we define a quantity which will measure, in some sense, how much information is produced by such a process, or better, at what rate information is produced?" [113].

In his answer, Shannon supposes that we have a set of possible events whose probabilities of occurrence are $p_1, p_2, \ldots, p_n$ and asks for the possibility of finding a measure, denoted by $H(p_1, p_2, \ldots, p_n)$, of how much "choice" is involved in the selection of the event or of how uncertain we are of the outcome. If this uncertainty measure exists, Shannon considers reasonable to require of it the following properties:

1. H would be continuous in the p_i.

[1] In Shannon's paper, the mutual information is called rate of transmission.

2. If all the p_i are equal (i.e., $p_i = 1/n$), then H should be a monotonic increasing function of n. With equally likely events, there is more choice, or uncertainty when there are more possible events.

3. If a choice is broken down into two successive choices, the original H should be the weighted sum of the individual values of H. The meaning of this property, called grouping property, is illustrated in Fig. 1.1[2].

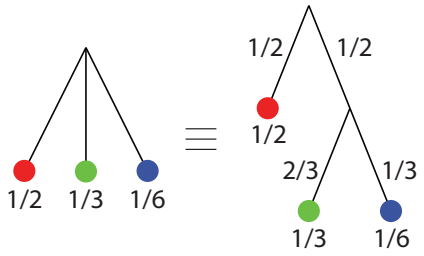


Figure 1.1: Grouping property of the entropy. On the left, we have three possibilities with probabilities $p_1 = 1/2$, $p_2 = 1/3$, $p_3 = 1/6$. On the right, we first choose between two possibilities, each with probability $1/2$, and if the second occurs, we make another choice with probabilities $2/3$, $1/3$. The final results have the same probabilities as before. In this example, it is required that $H(1/2, 1/3, 1/6) = H(1/2, 1/2) + (1/2)H(2/3, 1/3)$. The coefficient $1/2$ is because the second choice occurs with this probability.

After these requirements, Shannon proved the following theorem:

Theorem 1.1. *The only measure H satisfying the three above assumptions is of the form*

$$H = -K \sum_{i=1}^{n} p_i \log p_i, \tag{1.1}$$

where K is a positive constant[3]. To prove this theorem, Shannon assumed $H(1/n, 1/n, \ldots, 1/n) =$

$f(n)$ and decomposed a choice from s^m equally likely possibilities into a series of m choices from s equally likely possibilities. Thus, from the previous required property (3), $f(s^m) = mf(s)$. In essence, this expression contains the intuition that the uncertainty of m choices should be m times the uncertainty of only one choice. One function that fulfills this requirement is the logarithm function (see the complete proof in [113]).

[2]This example has been used by Shannon in [113]. Another example showing the recursive character of this property is in Fig. 4.1.
[3]This constant will be taken equal to 1 in the definition of entropy (Equ. 1.2).

There are other axiomatic formulations which involve the same definition of uncertainty [25, 89]. Shannon called this quantity entropy[4], as it can be identified with the entropy used in thermodynamics and statistical mechanics.

Let X be a discrete random variable[5] with alphabet $\mathcal{X}$ and probability distribution $\{p(x)\}$, where $p(x) = \Pr\{X = x\}$ and $x \in \mathcal{X}$. In this book, $\{p(x)\}$ will be also denoted by $p(X)$ or simply p. This notation will be extended to two or more random variables. As an example, a discrete random variable can be used to describe the toss of a fair coin, with alphabet $\mathcal{X} = \{head, tail\}$ and probability distribution $p(X) = \{1/2, 1/2\}$.

Definition 1.2. The entropy $H(X)$ of a discrete random variable X is defined by

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x), \tag{1.2}$$

where the summation is over the corresponding alphabet and the convention $0 \log 0 = 0$ is taken.

In this book, logarithms are taken in base 2 and, as a consequence, entropy is expressed in bits. The convention $0 \log 0 = 0$ is justified[6] by continuity since $x \log x \to 0$ as $x \to 0$. The term $-\log p(x)$ represents the information content (or uncertainty) associated with the result x. Thus, the entropy gives us the average amount of information (or uncertainty) of a random variable. Information and uncertainty are opposite. Uncertainty is considered before the event, information after. So, information reduces uncertainty. Note that the entropy depends only on the probabilities. We will use, interchangeably, the notation $H(X)$ or $H(p)$ for the entropy, where p stands for the probability distribution $p(X)$.

For example, the entropy of a fair coin toss is $H(X) = -(1/2)\log(1/2) - (1/2)\log(1/2) = \log 2 = 1$ bit. For the toss of a fair die with alphabet $\mathcal{X} = \{1, 2, 3, 4, 5, 6\}$ and probability distribution $p(X) = \{1/6, 1/6, 1/6, 1/6, 1/6, 1/6\}$, the entropy is $H(X) = \log 6 = 2.58$ bits.

Some relevant properties of the entropy [113] are

- $0 \leq H(X) \leq \log |\mathcal{X}|$
  - $H(X) = 0$ when all the probabilities are zero except one with unit value.
  - $H(X) = \log |\mathcal{X}|$ when all the probabilities are equal.
- If the probabilities are equalized, entropy increases.

[4] In the Eighteenth Century, R. Clausius introduced the term entropy in thermodynamics and L. Boltzmann gave its probabilistic interpretation in the context of statistical mechanics. The relationship between the Boltzmann entropy and Shannon entropy was developed in a series of papers by E. Jaynes [59]. The link between the second law of thermodynamics ("The entropy of an isolated system is non-decreasing") and the Shannon entropy is analyzed in [25].

[5] We assume that all random variables used are discrete unless otherwise specified.

[6] See in Yeung's book [146] the discussion on probability distributions which are not strictly positive.

The binary entropy (Fig. 1.2) of a random variable X with alphabet $\{x_1, x_2\}$ and probability distribution $\{p, 1-p\}$ is given by

$$H(X) = -p \log p - (1-p) \log(1-p). \tag{1.3}$$

Note that the maximum entropy is $H(X) = 1$ bit when $p = 1/2$.

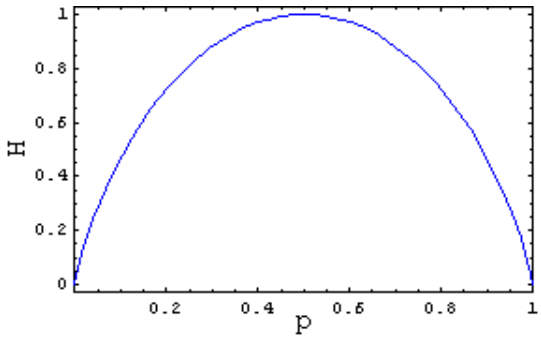


Figure 1.2: Plot of binary entropy.

The definition of entropy is now extended to a pair of random variables.

Definition 1.3. The joint entropy $H(X, Y)$ of a pair of discrete random variables X and Y with a joint probability distribution $p(X, Y) = \{p(x, y)\}$ is defined by

$$H(X, Y) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(x, y), \tag{1.4}$$

where $p(x, y) = \Pr[X = x, Y = y]$ is the joint probability of x and y. The conditional entropy of a random variable given another is defined as the expected value of the entropies of the conditional distributions.

Definition 1.4. The conditional entropy $H(Y|X)$ of a random variable Y given a random variable X is defined by

$$\begin{aligned} H(Y|X) &= \sum_{x \in \mathcal{X}} p(x) H(Y|X = x) = \sum_{x \in \mathcal{X}} p(x) \Big(-\sum_{y \in \mathcal{Y}} p(y|x) \log p(y|x) \Big) \\ &= -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(y|x), \end{aligned} \tag{1.5}$$

where $p(y|x) = \Pr[Y = y|X = x]$ is the conditional probability of y given x[7]. The conditional

[7]The Bayes theorem relates marginal probabilities $p(x)$ and $p(y)$, conditional probabilities $p(y|x)$ and $p(x|y)$, and joint probabilities $p(x, y)$:

$$p(x, y) = p(x)p(y|x) = p(y)p(x|y). \tag{1.6}$$

entropy can be thought of in terms of a communication or *information channel* $X \to Y$ whose output Y depends probabilistically on its input X. This information channel is characterized by a transition probability matrix which determines the conditional distribution of the output given the input [25]. Hence, $H(Y|X)$ corresponds to the uncertainty in the channel output from the sender's point of view, and vice versa for $H(X|Y)$. Note that in general $H(Y|X) \neq H(X|Y)$. In this book, the conditional probability distribution of Y given x will be denoted by $p(Y|x)$ and the transition probability matrix (i.e., the matrix whose rows are given by $p(Y|x)$) will be denoted by $p(Y|X)$.

The following properties hold:

- $H(X, Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$
- $H(X, Y) \leq H(X) + H(Y)$
- $H(X) \geq H(X|Y) \geq 0$
- If X and Y are independent, then $H(Y|X) = H(Y)$ since $p(y|x) = p(y)$ and, consequently, $H(X, Y) = H(X) + H(Y)$ (i.e., entropy is additive for independent random variables).

As an example, we consider the joint distribution $p(X, Y)$ represented in Fig. 1.3.*left*. The

$p(X,Y)$		$\mathcal{Y}$ y_1	y_2	$p(X)$
$\mathcal{X}$	x_1	0.125	0.125	0.25
	x_2	0.25	0	0.25
	x_3	0	0.5	0.5
$p(Y)$		0.375	0.625	
$H(X,Y) = 1.75$				

$p(Y\|X)$		$\mathcal{Y}$ y_1	y_2	$H(Y\|x \in \mathcal{X})$
$\mathcal{X}$	x_1	0.5	0.5	$H(Y\|x_1) = 1$
	x_2	1	0	$H(Y\|x_2) = 0$
	x_3	0	1	$H(Y\|x_3) = 0$
$H(Y\|X) = 0.25$				

Figure 1.3: Example of joint, marginal, and conditional probability distributions of random variables X and Y. On the left, joint distribution $p(X, Y)$, marginal distributions $p(X)$ and $p(Y)$, and joint entropy $H(X, Y)$. On the right, transition probability matrix $p(Y|X)$ and conditional entropy $H(Y|X)$.

marginal probability distributions of X and Y are given by $p(X) = \{0.25, 0.25, 0.5\}$ and $p(Y) = \{0.375, 0.625\}$, respectively. Thus, $H(X) = -0.25 \log 0.25 - 0.25 \log 0.25 - 0.5 \log 0.5 = 1.5$ bits, $H(Y) = -0.375 \log 0.375 - 0.625 \log 0.625 = 0.954$ bits, and $H(X, Y) = -0.125 \log 0.125 - 0.125 \log 0.125 - 0.25 \log 0.25 - 0 \log 0 - 0 \log 0 - 0.5 \log 0.5 = 1.75$ bits.

If X and Y are independent, then $p(x, y) = p(x)p(y)$. Marginal probabilities can be obtained from $p(x, y)$ by summation: $p(x) = \sum_{y \in \mathcal{Y}} p(x, y)$ and $p(y) = \sum_{x \in \mathcal{X}} p(x, y)$.

From the transition probability matrix $p(Y|X)$ represented in Fig. 1.3.*right*, we can compute $H(Y|X)$ as follows:

$$\begin{aligned} H(Y|X) &= \sum_{i=1}^{3} p(x_i) H(Y|X = x_i) \\ &= 0.25\ H(Y|X = x_1) + 0.25\ H(Y|X = x_2) + 0.5\ H(Y|X = x_3) \\ &= 0.25 \times 1 + 0.25 \times 0 + 0.5 \times 0 = 0.25 \text{ bits.} \end{aligned}$$

1.2 RELATIVE ENTROPY AND MUTUAL INFORMATION

We now introduce two new measures, relative entropy and mutual information, which quantify the distance between two probability distributions and the shared information between two random variables, respectively.

Definition 1.5. The relative entropy or Kullback-Leibler distance $D_{KL}(p, q)$ between two probability distributions p and q, that are defined over the alphabet $\mathcal{X}$, is defined by

$$D_{KL}(p, q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}. \tag{1.7}$$

The conventions that $0 \log(0/0) = 0$ and $a \log(a/0) = \infty$ if $a > 0$ are adopted. The relative entropy satisfies the divergence or information inequality

$$D_{KL}(p, q) \geq 0, \tag{1.8}$$

with equality if and only if $p = q$. The relative entropy is also called information divergence [27] or informational divergence [146], and it is not strictly a metric[8] since it is not symmetric and does not satisfy the triangle inequality.

Definition 1.6. The mutual information $I(X; Y)$ between two random variables X and Y is defined by

$$\begin{aligned} I(X; Y) &= H(X) - H(X|Y) = H(Y) - H(Y|X) \\ &= \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p(x)p(y)} = \sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y|x) \log \frac{p(y|x)}{p(y)}. \end{aligned} \tag{1.9}$$

Mutual information represents the amount of information that one random variable, the input of the channel, contains about a second random variable, the output of the channel, and vice versa. That

[8] A metric between x and y is defined as a function $d(x, y)$ that fulfills the following properties: (1) non-negativity: $d(x, y) \geq 0$, (2) identity: $d(x, y) = 0$ if and only if $x = y$, (3) symmetry: $d(x, y) = d(y, x)$, and (4) triangle inequality: $d(x, y) + d(y, z) \geq d(x, z)$.

is, mutual information expresses how much the knowledge of Y decreases the uncertainty of X, and vice versa. $I(X;Y)$ is a measure of the shared information or dependence between X and Y. Thus, if X and Y are independent, then $I(X;Y) = 0$. Note that the mutual information can be expressed as the relative entropy between the joint distribution and the product of marginal distributions:

$$I(X;Y) = D_{KL}(p(X,Y), p(X)p(Y)). \tag{1.10}$$

Mutual information $I(X;Y)$ fulfills the following properties:

- $I(X;Y) \geq 0$ with equality if and only if X and Y are independent
- $I(X;Y) = I(Y;X)$
- $I(X;Y) = H(X) + H(Y) - H(X,Y)$
- $I(X;Y) \leq \min\{H(X), H(Y)\}$
- $I(X;X) = H(X)$.

The relationship between Shannon's information measures can be expressed by a Venn diagram, as shown in Fig. 1.4[9]. The correspondence between Shannon's information measures and set theory is discussed in [146].

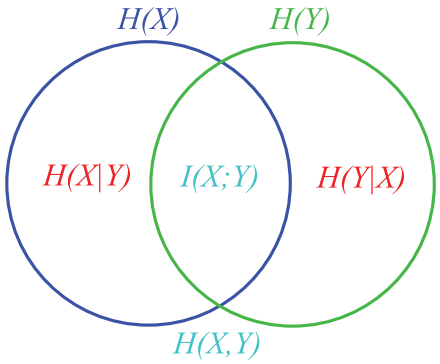


Figure 1.4: The information diagram represents the relationship between Shannon's information measures. Observe that $I(X;Y)$ and $H(X,Y)$ are represented, respectively, by the intersection and the union of the information in X (represented by $H(X)$) with the information in Y (represented by $H(Y)$). $H(X|Y)$ is represented by the difference between the information in X and the information in Y, and vice versa for $H(Y|X)$.

For the example presented in Fig. 1.3, the mutual information can be easily computed: $I(X;Y) = H(Y) - H(Y|X) = 0.954 - 0.25 = 0.704$ bits.

[9] The information diagram does not include the universal set as in a usual Venn diagram.

1.3 INEQUALITIES

In this section, we introduce a group of inequalities that are essential in the study of information theory and for the development of the concepts presented in this book [25, 146], and, in particular, to derive most of the refinement criteria.

1.3.1 JENSEN'S INEQUALITY

In this section, we introduce the concepts of convexity and concavity. Many important inequalities and results in information theory are obtained from the concavity of the logarithmic function.

Definition 1.7. A function $f(x)$ is convex over an interval $[a, b]$ (the graph of the function lies below any chord) if for every $x_1, x_2 \in [a, b]$ and $0 \leq \lambda \leq 1$,

$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2). \tag{1.11}$$

A function is strictly convex if equality holds only if $\lambda = 0$ or $\lambda = 1$.

Definition 1.8. A function $f(x)$ is concave (the graph of the function lies above any chord) if $-f(x)$ is convex.

For instance, x^2 and $x \log x$ (for $x > 0$) are strictly convex functions, and $\log x$ (for $x > 0$) is a strictly concave function. Fig. 1.5 plots $x \log x$ and $\log x$.

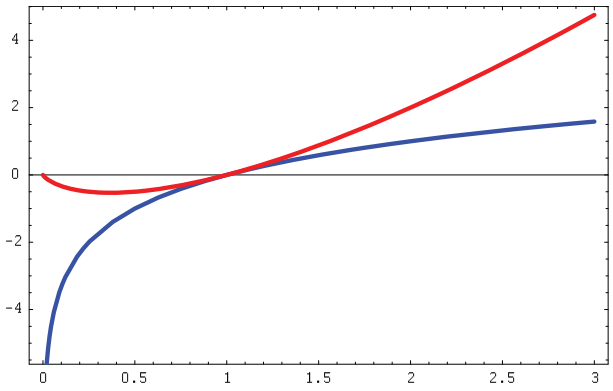


Figure 1.5: Plots of the strictly convex function $x \log x$ (red) and the strictly concave function $\log x$ (blue) for $x \in (0, 3]$.

Jensen's inequality can be expressed as follows. If f is a convex function on the interval $[a, b]$, then

$$\sum_{i=1}^{n} \lambda_i f(x_i) - f\left(\sum_{i=1}^{n} \lambda_i x_i\right) \geq 0, \tag{1.12}$$

where $0 \leq \lambda \leq 1, \sum_{i=1}^{n} \lambda_i = 1$, and $x_i \in [a, b]$. If f is a concave function, the inequality is reversed. A special case of this inequality is when $\lambda_i = 1/n$ because then

$$\frac{1}{n} \sum_{i=1}^{n} f(x_i) - f\left(\frac{1}{n} \sum_{i=1}^{n} x_i\right) \geq 0, \tag{1.13}$$

that is, the value of the function at the mean of the x_i is less or equal than the mean of the values of the function at each x_i.

Jensen's inequality can also be expressed in the following way: if f is convex on the range of a random variable X, then

$$f(E[X]) \leq E[f(X)], \tag{1.14}$$

where E denotes expectation (i.e., $E[f(X)] = \sum_{x \in \mathcal{X}} p(x) f(x)$). Observe that if $f(x) = x^2$ (convex function), then $E[X^2] - (E[X])^2 \geq 0$. Thus, the variance is always positive.

One of the most important consequences of Jensen's inequality is the divergence inequality $D_{KL}(p, q) \geq 0$ (Equ. 1.8). Some properties of Shannon's information measures presented in Sec. 1.1 and Sec. 1.2 can be derived from this inequality.

1.3.2 LOG-SUM INEQUALITY

The log-sum inequality can be obtained from Jensen's inequality (Equ. 1.12). For non-negative numbers $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$, the log-sum inequality is expressed as

$$\sum_{i=1}^{n} a_i \log \frac{a_i}{b_i} - \left(\sum_{i=1}^{n} a_i\right) \log \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i} \geq 0, \tag{1.15}$$

with equality if and only if a_i / b_i is constant for all i. The conventions that $0 \log 0 = 0, 0 \log(0/0) = 0$, and $a \log(a/0) = \infty$ if $a > 0$ are again adopted.

From this inequality, the following properties can be proved [25]:

- $D_{KL}(p, q)$ is convex in the pair (p, q).
- $H(X)$ is a concave function of p.
- If X and Y have the joint distribution $p(x, y) = p(x)p(y|x)$, then $I(X; Y)$ is a concave function of $p(x)$ for fixed $p(y|x)$ and a convex function of $p(y|x)$ for fixed $p(x)$.

1.3.3 JENSEN-SHANNON INEQUALITY

The Jensen-Shannon divergence, derived from the concavity of entropy, is used to measure the dissimilarity between two probability distributions and has the important feature that a different weight can be assigned to each probability distribution.

Definition 1.9. The Jensen-Shannon (JS) divergence is defined by

$$JS(\pi_1, \pi_2, \ldots, \pi_n; p_1, p_2, \ldots, p_n) = H\left(\sum_{i=1}^{n} \pi_i p_i\right) - \sum_{i=1}^{n} \pi_i H(p_i), \tag{1.16}$$

where $p_1, p_2, \ldots, p_n$ are a set of probability distributions defined over the same alphabet with prior probabilities or weights $\pi_1, \pi_2, \ldots, \pi_n$, fulfilling $\sum_{i=1}^{n} \pi_i = 1$, and $\sum_{i=1}^{n} \pi_i p_i$ is the probability distribution obtained from the weighted sum of the probability distributions $p_1, p_2, \ldots, p_n$. From the concavity of entropy (Sec. 1.3.2), the Jensen-Shannon inequality [10] is obtained:

$$JS(\pi_1, \pi_2, \ldots, \pi_n; p_1, p_2, \ldots, p_n) \geq 0. \tag{1.17}$$

The JS-divergence measures how far the probabilities p_i are from their mixing distribution $\sum_{i=1}^{n} \pi_i p_i$, and equals zero if and only if all the p_i are equal. It is important to note that the JS-divergence is identical to the mutual information $I(X; Y)$ when $\pi_i = p(x_i)$ (i.e., $\{\pi_i\}$ corresponds to the marginal distribution $p(X)$), $p_i = p(Y|x_i)$ for all $x_i \in \mathcal{X}$ (i.e., p_i corresponds to the conditional distribution of Y given x_i), and $n = |\mathcal{X}|$ [10, 120].

1.3.4 DATA PROCESSING INEQUALITY

The data processing inequality is expressed as follows. If $X \to Y \to Z$ is a Markov chain[10], then

$$I(X; Y) \geq I(X; Z). \tag{1.18}$$

This result proves that no processing of Y, deterministic or random, can increase the information that Y contains about X. In particular, if $Z = f(Y)$, then $X \to Y \to f(Y)$ and, consequently, $I(X; Y) \geq I(X; f(Y))$ [25].

1.4 ENTROPY RATE

Using the property $H(X_1, X_2) = H(X_1) + H(X_2|X_1)$ (Sec. 1.1) and the induction on n [146], it can be proved that the joint entropy of a collection of n random variables $X_1, \ldots, X_n$ is given by

$$H(X_1, \ldots, X_n) = \sum_{i=1}^{n} H(X_i|X_1, \ldots, X_{i-1}). \tag{1.19}$$

We now introduce the entropy rate that quantifies how the entropy of a sequence of n random variable increases with n.

[10] For random variables X, Y, and Z, $X \to Y \to Z$ forms a Markov chain if $p(x, y, z) = p(x)p(y|x)p(z|y)$. That is, the probability of the future state depends on the current state only and is independent of what happened before the current state. See a more general definition of Markov chain in Sec. 1.4.

Definition 1.10. The entropy rate or entropy density H_X of a stochastic process[11] $\{X_i\}$ is defined by

$$H_X = \lim_{n\to\infty} \frac{1}{n} H(X_1, X_2, \ldots, X_n) \tag{1.20}$$

when the limit exists. The entropy rate represents the average information content per symbol in a stochastic process. For a stationary stochastic process[12], the entropy rate exists and is equal to

$$H_X = \lim_{n\to\infty} H_X(n), \tag{1.21}$$

where $H_X(n) = H(X_1, \ldots, X_n) - H(X_1, \ldots, X_{n-1}) = H(X_n|X_{n-1}, \ldots, X_1)$. Entropy rate can be seen as the uncertainty associated with a given symbol if all the preceding symbols are known. It can be also interpreted as the irreducible randomness in sequences produced by an information source [36].

If $\{X_i\}$ is a Markov chain[13], then X_n is called the state at time n. A stationary Markov chain is characterized by its initial state and a transition probability matrix $\mathbf{P} = \{P_{ij}\}$, where $P_{ij} = \Pr\{X_{n+1} = j|X_n = i\}$ is called transition probability. A distribution on the states such that the distribution $\mathbf{w} = \{w_i\}$ at time $n + 1$ is the same as the distribution at time n is called a *stationary distribution*. A Markov chain is called *irreducible* if it is possible to go from every state to every state in a finite number of steps, that is, there is always a path between any two states. A Markov chain is said to be *aperiodic* if it has no periodic state. A periodic state is a state that can be visited back by a path starting from it only at multiples of a given period [21]. An irreducible and aperiodic Markov chain is called *ergodic*. For an ergodic Markov chain, the stationary distribution $\mathbf{w}$ exists and is unique, and $w_j = \lim_{n\to\infty}(\mathbf{P}^n)_{ij}$. The stationary distribution satisfies the left eigenvector equation $\mathbf{wP} = \mathbf{w}$. Thus, we can also think of the stationary distribution as a left eigenvector of the transition probability matrix $\mathbf{P}$.

The entropy rate of a stationary Markov chain, with stationary distribution $\mathbf{w}$ and transition probability matrix $\mathbf{P}$, is given by

$$\begin{aligned} H_X &= \lim_{n\to\infty} \frac{1}{n} H(X_n|X_n - 1, \ldots, X_1) = \lim_{n\to\infty} H(X_n|X_{n-1}) \\ &= H(X_2|X_1) = -\sum_{i=1}^{n} w_i \sum_{j=1}^{n} P_{ij} \log P_{ij}. \end{aligned} \tag{1.22}$$

[11] A stochastic process or a discrete-time information source $\{X_i\}$ is an indexed sequence of random variables characterized by the joint probability distribution $p(x_1, x_2, \ldots, x_n) = \Pr\{(X_1, X_2, \ldots, X_n) = (x_1, x_2, \ldots, x_n)\}$ with $(x_1, x_2, \ldots, x_n) \in \mathcal{X}^n$ for $n \geq 1$ [25, 146].

[12] A stochastic process $\{X_i\}$ is stationary if two subsets of the sequence, $\{X_1, X_2, \ldots, X_n\}$ and $\{X_{1+l}, X_{2+l}, \ldots, X_{n+l}\}$, have the same joint probability distribution for any $n, l \geq 1$: $\Pr\{(X_1, \ldots, X_n) = (x_1, x_2, \ldots, x_n)\} = \Pr\{(X_{1+l}, X_{2+l}, \ldots, X_{n+l}) = (x_1, x_2, \ldots, x_n)\}$. That is, the statistical properties of the process are invariant to a shift in time. At least, H_X exists for all stationary stochastic processes.

[13] A stochastic process $\{X_i\}$ is a Markov chain if $\Pr(X_{n+1} = x_{n+1}|X_n = x_n, \ldots, X_1 = x_1) = \Pr(X_{n+1} = x_{n+1}|X_n = x_n)$, for $n = 1, 2, \ldots$ and for all $x_i \in \mathcal{X}$.

1.5 ENTROPY AND CODING

In this section, we present different interpretations of the Shannon entropy:

- As we have seen in Sec. 1.1, $-\log p(x)$ represents the information associated with the result x. The value $-\log p(x)$ can also be interpreted as the surprise associated with the outcome x. If $p(x)$ is small, the surprise is large; if $p(x)$ is large, the surprise is small. Thus, entropy (Equ. 1.2) can be seen as the expectation value of the surprise [35].

- A fundamental result of information theory is the Shannon source coding theorem, which deals with the encoding of information in order to store or transmit it efficiently. This theorem can be formulated in the following ways [25, 35]:

  - Given a random variable X, $H(X)$ fulfills

$$H(X) \leq \overline{\ell} < H(X) + 1, \tag{1.23}$$

    where $\overline{\ell}$ is the expected length of an optimal binary code for X. An example of an optimal binary code is the Huffman instantaneous coding[14].

  - If we optimally encode n identically distributed random variables X with a binary code, the Shannon source coding theorem can be enunciated in the following way:

$$H(X) \leq \overline{\ell}_n < H(X) + \frac{1}{n}, \tag{1.24}$$

    where $\overline{\ell}_n$ is the expected codeword length per unit symbol. Thus, by using large block lengths, we can achieve an expected codelength per symbol arbitrarily close to the entropy [25].

  - For a stationary stochastic process, we have

$$\frac{H(X_1, X_2, \ldots, X_n)}{n} \leq \overline{\ell}_n < \frac{H(X_1, X_2, \ldots, X_n)}{n} + 1 \tag{1.25}$$

    and, from the definition of entropy rate H_X (Equ. 1.20),

$$\lim_{n\to\infty} \overline{\ell}_n \to H_X. \tag{1.26}$$

    Thus, the entropy rate is the expected number of bits per symbol required to describe the stochastic process.

[14] A code is called a prefix or instantaneous code if no codeword is a prefix of any other codeword. Huffman coding uses a specific algorithm to obtain the representation for each symbol. The main characteristic of this code is that the most common symbols use shorter strings of bits than the ones used by the less common symbols.

- From the previous Shannon theorem, it can be proved that entropy is related to the difficulty in guessing the outcome of a random variable [25, 35] since

$$H(X) \leq \overline{q} < H(X) + 1, \tag{1.27}$$

where $\overline{q}$ is the average minimum number of binary questions to determine X. This idea agrees with the interpretation of entropy as a measure of uncertainty.

1.6 CONTINUOUS CHANNEL

In this section, entropy and mutual information are defined for continuous random variables. Let X be a continuous random variable with continuous cumulative distribution function $F(x) = \Pr\{X \leq x\}$. When the derivative $F'(x) = f(x)$ is defined and $\int_{-\infty}^{\infty} f(x)\mathrm{d}x = 1$, then $f(x)$ is called the probability density function (pdf) of X. The support of X is given by $\mathcal{S}_X = \{x : f(x) > 0\}$, that is, the set of points where the function is non-zero. The statement "if it exists" should be included in the following definitions involving integrals and probability density functions. See a more detailed presentation in Cover and Thomas [25], and Yeung [146].

The differential entropy of a continuous random variable X is defined similarly to the entropy of a discrete random variable (see Equ. 1.2).

Definition 1.11. The continuous or differential entropy $h(X)$ of a continuous random variable X with a pdf $f(x)$ is defined by

$$h(X) = -\int_{\mathcal{S}_X} f(x) \log f(x)\mathrm{d}x. \tag{1.28}$$

Definition 1.12. For two continuous random variables X and Y with joint pdf $f(x, y)$, the continuous conditional entropy $h(Y|X)$ is defined as

$$h(Y|X) = -\int_{\mathcal{S}_X} \int_{\mathcal{S}_Y(x)} f(x, y) \log f(y|x)\mathrm{d}x\mathrm{d}y, \tag{1.29}$$

where $f(y|x)$ is the conditional pdf and $\mathcal{S}_Y(x) = \{y : f(y|x) > 0\}$.

Definition 1.13. For two continuous random variables X and Y with joint pdf $f(x, y)$, the continuous mutual information $I^c(X; Y)$ is defined as

$$I^c(X; Y) = h(X) - h(X|Y) = \int_{\mathcal{S}_X} \int_{\mathcal{S}_Y(x)} f(x, y) \log \frac{f(x, y)}{f(x)f(y)}\mathrm{d}x\mathrm{d}y. \tag{1.30}$$

Following the exposition in [25], we divide the range of the continuous random variable X into discrete bins of length Δ (Fig. 1.6.*a*). Then, assuming the continuity of $f(x)$ within the bins and using the mean value theorem, for each bin there exists a value x_i such that

$$f(x_i)\Delta = \int_{i\Delta}^{(i+1)\Delta} f(x)\mathrm{d}x. \tag{1.31}$$

The discretised version of X is defined by

$$X_\Delta = x_i, \qquad \text{if } i\Delta \leq X < (i+1)\Delta, \tag{1.32}$$

with probability distribution $p(x_i) = \Pr\{X_\Delta = x_i\} = f(x_i)\Delta$. Thus, the entropy of X_Δ is given by

$$\begin{aligned} H(X_\Delta) &= -\sum_i p(x_i)\log p(x_i) = -\sum_i f(x_i)\Delta\log(f(x_i)\Delta) \\ &= -\sum_i f(x_i)\Delta\log f(x_i) - \log\Delta. \end{aligned} \tag{1.33}$$

If $f(x)\log f(x)$ is Riemann integrable, we obtain that

$$\lim_{\Delta\to 0}(H(X_\Delta) - \log\Delta) = h(X), \tag{1.34}$$

since $h(X) = \lim_{\Delta\to 0}(-\sum_i f(x_i)\Delta\log f(x_i))$. Thus, in general, the entropy of a continuous random variable does not equal the entropy of the discretized random variable in the limit of a finer discretisation. We can also see that, due to the fact that $-\lim_{\Delta\to 0}\log\Delta = \infty$, the entropy $H(X_\Delta)$ goes to infinity when the bin size goes to zero:

$$\lim_{\Delta\to 0} H(X_\Delta) = \infty. \tag{1.35}$$

For instance, if $f(x) = 1/k$ in the interval $(0, k)$ (Fig. 1.6.*b*), then $h(X) = -\int_0^k (1/k)\log(1/k)\mathrm{d}x = \log k$. Observe that the differential entropy is negative when $k < 1$.

In contrast with the behavior of the differential entropy, the mutual information between two continuous random variables X and Y is the limit of the mutual information between their discretised versions. Thus, in the limit of a finer discretisation we get

$$I^c(X;Y) = \lim_{\Delta\to 0} I(X_\Delta; Y_\Delta). \tag{1.36}$$

Kolmogorov [68] and Pinsker [92] defined mutual information as $I^c(X;Y) = sup_{P,Q} I([X]_P;[Y]_Q)$, where the supremum ($sup$) is over all finite partitions P and Q of X and Y, respectively. From this definition and Equ. 1.36, two important properties can be derived: the continuous mutual information is the least upper bound for the discrete mutual information, and refinement can never decrease the discrete mutual information. This last property can also be deduced from the data processing inequality (Equ. 1.18) [51].

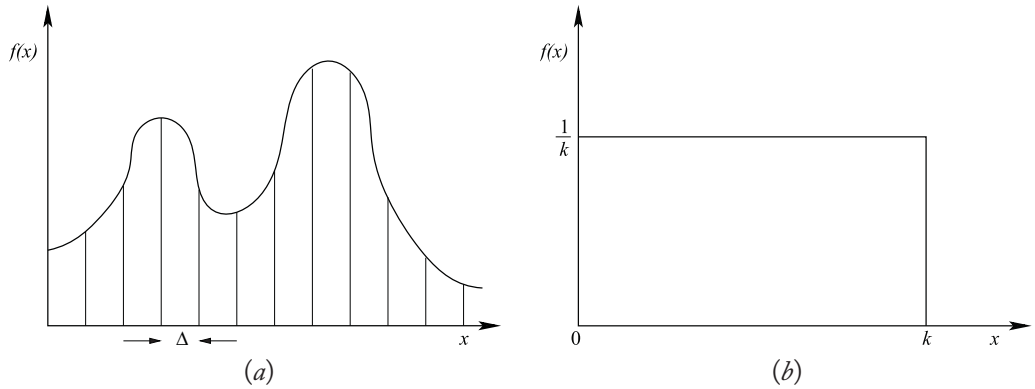


Figure 1.6: (*a*) Probability density function of X. The range of the continuous random variable X is divided into discrete bins of length Δ. (*b*) Constant probability density function $f(x) = 1/k$.

1.7 INFORMATION BOTTLENECK METHOD

The information bottleneck method, introduced by Tishby et al. [129], is a technique that extracts a compact representation of the variable X, denoted by $\widehat{X}$, with minimal loss of mutual information with respect to another variable Y (i.e., $\widehat{X}$ preserves as much information as possible about the control variable Y). Thus, given an information channel between X and Y, the information bottleneck method tries to find the optimal tradeoff between accuracy and compression of X when the bins of this variable are clustered.

Soft [129] and hard [119] partitions of X can be adopted. In the first case, every $x \in \mathcal{X}$ can be assigned to a cluster $\hat{x} \in \widehat{\mathcal{X}}$ with some conditional probability $p(\hat{x}|x)$ (soft clustering). In the second case, every $x \in \mathcal{X}$ is assigned to only one cluster $\hat{x} \in \widehat{\mathcal{X}}$ (hard clustering).

In this book, we consider hard partitions, and we focus our attention on the agglomerative information bottleneck method [119]. Given a cluster $\hat{x}$ defined by $\hat{x} = \{x_1, \ldots, x_l\}$, where $x_k \in \mathcal{X}$ for all $k \in \{1, \ldots, l\}$, and the probabilities $p(\hat{x})$ and $p(y|\hat{x})$ defined by

$$p(\hat{x}) = \sum_{k=1}^{l} p(x_k), \tag{1.37}$$

$$p(y|\hat{x}) = \frac{1}{p(\hat{x})} \sum_{k=1}^{l} p(x_k, y) \quad \forall y \in \mathcal{Y}, \tag{1.38}$$

the following properties are fulfilled:

- The decrease in the mutual information $I(X;Y)$ due to the merge of $x_1, \ldots, x_l$ is given by

$$\delta I_{\hat{x}} = p(\hat{x}) JS(\pi_1, \ldots, \pi_l; p_1, \ldots, p_l) \geq 0, \tag{1.39}$$

where the weights and probability distributions of the JS-divergence are given by $\pi_k = p(x_k)/p(\hat{x})$ and $p_k = p(Y|x_k)$ for all $k \in \{1, \ldots, l\}$, respectively. An optimal clustering algorithm should minimize $\delta I_{\hat{x}}$.

- An optimal merge of l components can be obtained by $l - 1$ consecutive optimal merges of pairs of components.

1.8 f-DIVERGENCES

Many different measures quantifying the divergence between two probability distributions have been studied in the past. They are frequently called "distances", although some of them are not strictly metrics. Some particular examples of divergences play an important role in different fields such as statistics and information theory [90].

Next, we present a measure of divergence between two probability distributions called f-divergence. This measure was independently introduced by Csiszár [26] and Ali and Silvey [1]. The following definition is taken from Csiszár and Shields [27].

Definition 1.14. Let $f(t)$ be a convex function defined for $t > 0$, with $f(1) = 0$. The f-divergence of a distribution p from q is defined by

$$D_f(p, q) = \sum_{x \in \mathcal{X}} q(x) f\left(\frac{p(x)}{q(x)}\right), \tag{1.40}$$

where the conventions $0 f(0/0) = 0$, $f(0) = \lim_{t \to 0} f(t)$, $0 f(a/0) = \lim_{t \to 0} t f(a/t) = a \lim_{u \to \infty} (f(u)/u)$ are adopted.

For the purposes of this book, we present three of the most important f-divergences: Kullback-Leibler, Chi-square, and Hellinger distances. These can be obtained from different convex functions f (see Fig. 1.7):

- Kullback-Leibler distance or information divergence [69]:
  If $f(t) = t \log t$, the Kullback-Leibler distance is given by

$$D_{KL}(p, q) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)}. \tag{1.41}$$

- Chi-square distance [91]:
  If $f(t) = (t - 1)^2$, the Chi-square distance is given by

$$D_{\chi^2}(p, q) = \sum_{x \in \mathcal{X}} \frac{(p(x) - q(x))^2}{q(x)}. \tag{1.42}$$

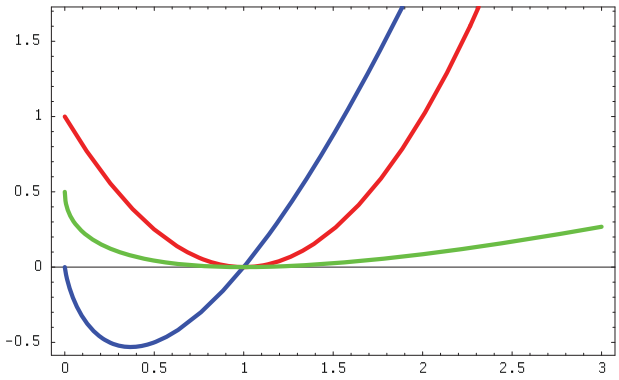


Figure 1.7: Plots for three strictly convex functions: $t \log t$ (blue), $(t-1)^2$ (red), and $1/2(1-\sqrt{t})^2$ (green). From these functions, the Kullback-Leibler, Chi-square, and Hellinger distances are obtained, respectively.

- Hellinger distance [54]:
  If $f(t) = 1/2(1-\sqrt{t})^2$, the Hellinger distance is given by

$$D_{h^2}(p,q) = \frac{1}{2}\sum_{x\in\mathcal{X}}(\sqrt{p(x)} - \sqrt{q(x)})^2. \tag{1.43}$$

Note that none of the above distances fulfills all the properties of a metric. However, the square root of the Hellinger distance is a true metric [31].

According to Csiszár and Shields [27], f-divergences generalize the Kullback-Leibler distance. Using the analogue of the log-sum inequality (Sec. 1.3.2), given by

$$\sum_{i=1}^{n} b_i f\left(\frac{a_i}{b_i}\right) - \left(\sum_{i=1}^{n} b_i\right) f\left(\frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i}\right) \geq 0, \tag{1.44}$$

many of the properties of the information divergence extend to general f-divergences. If f is strictly convex the equality in Equ. 1.44 holds if and only if a_i/b_i is constant for all i.

1.9 GENERALIZED ENTROPIES

Rényi [98] proposed a generalized entropy which recovers the Shannon entropy as a special case, and Harvda and Charvát [53] introduced a new generalized definition of entropy which also includes the Shannon entropy as a particular case. Sharma and Mittal [114], and Sharma and Taneja [115] introduced two-parameter entropies where Rényi and Harvda-Charvát entropies are particular cases. Tsallis [131] used the Harvda-Charvát entropy in order to generalize the Boltzmann entropy in statistical mechanics. The introduction of this entropy responds to the objective of generalizing

the statistical mechanics to non-extensive systems[15]. For the objectives of this book, we review the so-called Harvda-Charvát-Tsallis entropy.

Definition 1.15. The Harvda-Charvát-Tsallis entropy $H_\alpha^T(X)$ of a discrete random variable X is defined by

$$H_\alpha^T(X) = k\frac{1-\sum_{x\in\mathcal{X}} p(x)^\alpha}{\alpha-1}, \tag{1.45}$$

where k is a positive constant (by default $k=1$) and $\alpha \in \mathbb{R}\backslash\{1\}$ is called entropic index. This entropy recovers the Shannon discrete entropy when $\alpha \to 1$ and fulfills the properties of non-negativity and concavity (for $\alpha > 0$). If X and Y are independent, then the Harvda-Charvát-Tsallis entropy fulfills the non-additivity property:

$$H_\alpha^T(X,Y) = H_\alpha^T(X) + H_\alpha^T(Y) + (1-\alpha)H_\alpha^T(X)H_\alpha^T(Y), \tag{1.46}$$

hence, superextensivity, extensivity or subextensivity occurs when $\alpha < 1$, $\alpha = 1$ or $\alpha > 1$, respectively [133].

Taneja [126] and Tsallis [132] also introduced the generalized mutual information.

Definition 1.16. The Harvda-Charvát-Tsallis mutual information $I_\alpha^T(X,Y)$ between two discrete random variables X and Y is defined by

$$I_\alpha^T(X,Y) = \frac{1}{1-\alpha}\left(1-\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}}\frac{p(x,y)^\alpha}{p(x)^{\alpha-1}p(y)^{\alpha-1}}\right). \tag{1.47}$$

Shannon mutual information is recovered when $\alpha \to 1$. The transition of $I_\alpha^T(X,Y)$ to the continuous generalized mutual information is straightforward. Some alternative forms for the generalized mutual information can be seen in Taneja [127].

[15] An extensive system fulfills that quantities like energy and entropy are proportional to the system size. Similarly, to Shannon entropy, a fundamental property of the Boltzmann entropy is its additivity. That is, if we consider a system composed by two probabilistically independent subsystems X and Y (i.e., $p(x,y) = p(x)p(y)$), then H(X,Y) = H(X) + H(Y). This property ensures the extensivity of the entropy but strongly correlated systems present non-extensive properties that require another type of entropy fulfilling non-additivity. Tsallis proposed the Harvda-Charvát entropy in order to deal with these "pathological" systems.

CHAPTER 2

Scene Complexity and Refinement Criteria for Radiosity

One of the most important topics in computer graphics is the accurate computation of the *global illumination* in a scene (i.e., the computation of the intensities of light taking into account all the bounces over the surfaces of a scene). This type of simulation is called global illumination and is represented by the rendering equation [61], which is a Fredholm integral equation of the second kind. However, obtaining an exact representation of the illumination is an intractable problem. Many different techniques are used to obtain an approximate quantification of it [20, 117, 44]. In this chapter, we deal with global illumination using the radiosity method, which only considers diffuse surfaces, where reflected light does not depend on the incoming direction. Radiosity is a viewpoint independent global illumination technique that discretises the scene into small polygons or patches to solve a transport equation system.

In the radiosity setting, the difficulty in obtaining an accurate solution of the global illumination in a scene mainly depends on the degree of dependence between all the surfaces. In this chapter, this dependence, called *scene complexity*, is quantified by the mutual information, which is a measure of the information transfer between the different parts of a scene [33, 32]. The scene complexity will be used in obtaining a mutual information-based refinement criterion (oracle) [32] for the hierarchical radiosity algorithm [52]. Finally, by analogy with this oracle, refinement criteria based on f-divergences are also introduced [103].

2.1 BACKGROUND

In this section, the radiosity method is shortly reviewed and some basic hints on form factor computation and scene random walk are given.

2.1.1 RADIOSITY METHOD

The radiosity method, introduced by Goral et al. [47], Nishita and Nakamae [82], and Cohen and Greenberg [19], solves the problem of illumination in a virtual environment (or scene) of diffuse surfaces. The radiosity of a patch surface is the light energy leaving this patch per discrete time interval and can be seen as the combination of emitted and reflected energy for the patch. Thus, the

radiosity algorithm computes the amount of light energy transferred among the surfaces of a scene, assuming that the scattering at all surfaces is perfectly diffuse.

The radiosity equation can be written in the following form:

$$B(x) = E(x) + \rho(x) \int_{\mathcal{S}} B(y) F(x, y) \mathrm{d}A_y, \tag{2.1}$$

where $\mathcal{S}$ represents the surface of the environment, x and y are points on $\mathcal{S}$, $\mathrm{d}A_y$ is a differential area at point y, $B(x)$ and $B(y)$ are, respectively, the radiosities (W/m^2) at points x and y, $E(x)$ is the emittance or emitted flux of energy per unit area (W/m^2) at point x, $\rho(x)$ is the diffuse reflectance (dimensionless) at point x, and

$$F(x, y) = \frac{\cos\theta_x \cos\theta_y}{\pi r_{xy}^2} V(x, y), \tag{2.2}$$

is the point-to-point form factor , where θ_x and θ_y are the angles that the line joining x and y form with the normals at x and y, respectively, r_{xy} is the distance between x and y, and $V(x, y)$ is a visibility function which is equal to 1 if x and y are mutually visible and 0 if not (Fig. 2.1.*a*). Equ. 2.1 expresses that the radiosity at point x is equal to the emitted energy at x plus the reflected energy, given by the reflectance at x multiplied by the arriving energy at x from all points y in a scene.

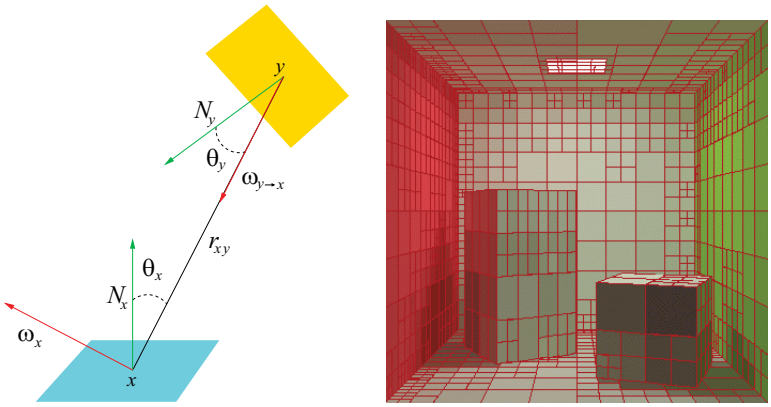


Figure 2.1: (*a*) Outgoing and incoming directions at point x. (*b*) The Cornell box with a mesh of 1, 995 patches.

To solve the radiosity equation, we can use a finite element approach, discretising the environment into N_p patches and considering the radiosities, emissivities, and reflectances constant over the patches (Fig. 2.1.*b*). With these assumptions, the integral Equ. 2.1 becomes the system of radiosity

equations [47]:

$$B_i = E_i + \rho_i \sum_{j=1}^{N_p} F_{ij} B_j, \tag{2.3}$$

where B_i, E_i, and ρ_i are, respectively, the radiosity, emittance (or emissivity), and reflectance of patch i, B_j is the radiosity of patch j, and F_{ij} is the patch-to-patch form factor, only dependent on the geometry of the scene and defined by

$$F_{ij} = \frac{1}{A_i} \int_{\mathcal{S}_i} \int_{\mathcal{S}_j} F(x, y) \mathrm{d}A_y \mathrm{d}A_x = \frac{1}{A_i} \int_{\mathcal{S}_i} \int_{\Omega_x} \frac{\cos\theta_x}{\pi} V(x, y) \mathrm{d}\omega_x \mathrm{d}A_x, \tag{2.4}$$

where A_i is the area of patch i, $\mathcal{S}_i$ and $\mathcal{S}_j$ represent, respectively, the surfaces of patches i and j, and x and y are, respectively, points on $\mathcal{S}_i$ and $\mathcal{S}_j$, Ω_x represents the hemisphere of directions around $\mathrm{d}A_x$, $\mathrm{d}\omega_x = (\cos\theta_y / r_{xy}^2)\mathrm{d}A_y$ is the differential solid angle from $\mathrm{d}A_x$ to $\mathrm{d}A_y$, and $V(x, y)$ expresses the visibility of $\mathrm{d}A_y$ from $\mathrm{d}A_x$ in direction w_x. Form factor F_{ij} between patches i and j expresses the fraction of energy leaving patch i which goes directly to patch j.

Form factors fulfill the following properties:

- Reciprocity

$$A_i F_{ij} = A_j F_{ji} \quad \forall i, j \in \{1, \ldots, N_p\}. \tag{2.5}$$

- Energy conservation

$$\sum_{j=1}^{N_p} F_{ij} = 1 \quad \forall i \in \{1, \ldots, N_p\}. \tag{2.6}$$

In brief, the radiosity method consists of the following steps: discretisation of the surfaces of the scene into a mesh of polygons called patches, computation of form factors (see Sec. 2.1.2), solution of the system of linear equations, and visualization of the solution [20, 117, 44]. The main problems of the radiosity method are meshing and form factor computation. The form factor computation is the most costly step. More specifically, its cost is mainly due to the presence of the visibility term in the geometric kernel. Scene meshing not only has to accurately represent illumination variations, but it also has to avoid unnecessary subdivisions of the surfaces that would increase the number of form factors to be computed, and consequently, the computational time. The best strategy tries to balance accuracy and computational cost.

To manage the complexity of the radiosity computation, different strategies can be used to reduce the number of form factors that need to be computed: progressive refinement, substructuring, adaptive refinement, hierarchical refinement, etc. Other strategies try to reduce the number of form factors arriving at a solution within a given error bound [117].

For the purposes of this chapter, we focus our attention on the hierarchical refinement algorithm, introduced by Hanrahan and Salzman [52]. Additional information can be found in [117, 44].

This algorithm is based on the objective of reducing the number of form factors needed to propagate the light through the environment. In hierarchical radiosity, the patches are subdivided into smaller elements, if necessary, in order to achieve an accurate light transport between them. The main objective is to obtain an accurate piecewise constant approximation of the radiosity on all the elements. To do this, the mesh is generated adaptively: when a constant radiosity assumption on patch i is not valid for the radiosity due to another patch, the refinement algorithm will refine i in a set of subpatches or elements. Finally, a multiresolution element mesh will enable us to accurately represent the energy transport between patches [44]. A refinement criterion (or oracle) based on an error estimation informs us if a subdivision of the surfaces is needed. The oracle takes geometrical and visibility information about the patches and also the source radiosity and receiver reflectance, and returns whether or not the interaction is valid. Some of patches will need further refinement until a certain level where no further refinement is needed or a previously imposed bound on the area of the patches is reached.

2.1.2 FORM FACTOR COMPUTATION

Due to the importance of form factor computation for the objectives of this chapter, we show how the form factor integral (Equ. 2.4) can be evaluated by the Monte Carlo method. Monte Carlo integration enables us to estimate numerically an integral by converting it to an expected value of a random variable. Suppose we want to solve the integral of a function $g(x)$, which can be factored into the product $(g(x)/f(x))f(x)$ with $f(x) \neq 0$ when $g(x) \neq 0$. This can be written as

$$I = \int_D g(x)dx = \int_D \frac{g(x)}{f(x)} f(x)\mathrm{d}x. \tag{2.7}$$

If $f(x) \geq 0$ $(\forall x \in D)$ and $\int_D f(x)\mathrm{d}x = 1$, then $f(x)$ can be considered as a *probability density function* of a random variable X, and the integral of Equ. 2.7 can be read as the expected value of the random variable $g(X)/f(X)$ with respect to the pdf $f(x)$:

$$I = E_f\left[\frac{g(X)}{f(X)}\right]. \tag{2.8}$$

Then, I can be estimated by sampling N independent values $x_1, x_2, \ldots, x_N$ from $f(x)$:

$$\widehat{I}_N = \frac{1}{N}\sum_{k=1}^{N}\frac{g(x_k)}{f(x_k)}. \tag{2.9}$$

This estimator is unbiased, that is, the expected value of this estimator is the value of the integral: $E[\widehat{I}_N] = I$ (for a more detailed description, see [62]).

Uniform area sampling and local and global lines to estimate a form factor are reviewed here (for a brief survey, see [109, 4]). The uniform area sampling takes random points x and y on patches i and j, respectively, (Fig. 2.2.*a*). This means taking as pdf $f(x, y) = 1/(A_i A_j)$. For N_s pairs (x, y)

of samples, the form factor integral (Equ. 2.4) is approximated by the following estimator:

$$\widehat{F}_{ij} = A_j \frac{1}{N_s} \sum_{k=1}^{N_s} F(x_k, y_k). \tag{2.10}$$

We can also consider a set of rays, which we call *local lines*[1], with origin uniformly distributed over the surface of patch i and directions distributed according to the cosine with respect to the surface normal on i. So, we estimate the integral of Equ. 2.4 taking as pdf $f(x, w_x) = (1/A_i)\cos\theta_x/\pi$. The form factor F_{ij} can be interpreted as the fraction of local lines with origin on patch i that have j as the nearest patch intersected (Fig. 2.2.*b*). An estimator for F_{ij} is given by

$$\widehat{F}_{ij} = \frac{N_{ij}}{N_i}, \tag{2.11}$$

where N_i is the number of local lines with origin on patch i and N_{ij} is the number of local lines with origin on patch i that hit patch j.

The same estimator can be obtained using *uniformly distributed lines*[2]. Uniformly distributed random lines can be generated by putting the scene within a sphere and selecting pairs of random points on the surface of this sphere. The lines connecting each pair of points are uniformly distributed throughout the scene. These lines are also called *global lines* [108]. So, the form factor F_{ij} can also be considered as the probability of a global line that, crossing patch i, hits patch j as the first intersected patch (Fig. 2.2.*c*). From integral geometry [107, 109] we know that, for a planar patch, the probability that a global line intersects patch i is proportional to A_i. To sample with global lines is equivalent to casting, for each patch, a number of local lines proportional to its area. If we identify the lines connecting two patches with visibility, the form factor will give us the visibility between patches [108, 109].

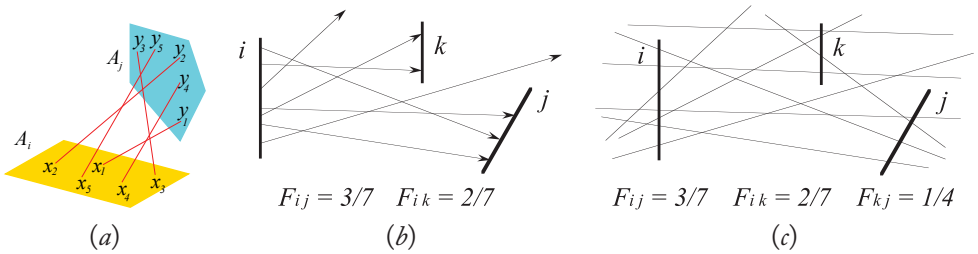


Figure 2.2: Form factors are computed using (*a*) uniform area sampling, (*b*) local lines, and (*c*) global lines.

[1] *Local* in the sense that the lines depend on patch i; in contraposition to the global lines, independent of the patches.
[2] The study of uniformly distributed lines will be extended in Sec. 3.1.

2.1.3 SCENE RANDOM WALK

A discrete random walk [62, 106] is a Monte Carlo technique that can be used to solve the radiosity equation (2.3). A *random walk* in a scene can be considered as a stationary Markov chain[3] [21, 25, 80]. As we have seen in Sec. 1.4, a stationary Markov chain is a discrete stochastic process defined over a set of states $\mathcal{S} = \{1, 2, \ldots, m\}$ which is described by a transition probability matrix $\mathbf{P} = \{P_{ij}\}$. In each step, an imaginary particle (or ray) makes a transition from its current state i to a new state j with *transition probability* P_{ij}. This transition probability only depends on the current state. If we are in state i, the probability of being in state j after n steps is $(\mathbf{P}^n)_{ij}$. For an ergodic Markov chain, the probabilities of finding the particle in each state i converge to a stationary distribution $w = \{w_1, \ldots, w_m\}$ after a number of steps.

It is easy to see that a random walk in a scene[4] is an ergodic Markov chain (i.e., irreducible and aperiodic), where the transition probability matrix is given by the form factor matrix $\mathbf{F} = \{Fi_{ij}\}$ and the states correspond to the patches of the scene. Using the left eigenvector equation, we can see that the stationary or equilibrium distribution for a scene random walk is given by $\mathbf{a} = \{a_i\}$. Thus, the form factor matrix $\mathbf{F}$ fulfills

$$\lim_{m\to\infty} (\mathbf{F}^m)_{ij} \to \frac{A_j}{A_T} = a_j \quad \forall i, j \in \{1, \ldots, N_\mathrm{p}\}, \tag{2.12}$$

where A_j is the area of patch j, $A_T = \sum_{i=1}^{N_\mathrm{p}} A_i$, and a_j is the relative area of patch j, and, when the length of a scene random walk grows to infinity, the number of hits on any patch i becomes proportional to a_i, independently of where the random walk started its trajectory. A scene random walk is also reversible since, for every pair of patches, $w_i P_{ij} = w_j P_{ji}$.

When the states form a finite set, as stated before, the Markov chain is called a *discrete* chain. When the states are not countable, the chain is called *continuous*. For instance, when the states are taken as infinitesimal areas $\mathrm{d}A_x$ at each point x on the surfaces S of the scene as the states and transition probabilities are given by the differential form factors $F(x, y)$, with $x, y \in S$, a continuous Markov chain with stationary distribution $w(x) = 1/A_T$ results.

The previous results for a 3D scene can be extended to 2D[5]. It can be shown that in 2D the stationary probabilities of the resulting discrete Markov chain are given by $w_i = L_i/L_T = l_i$, where L_T is the total length of all segments of the scene, L_i is the length of segment i and l_i is the relative length of segment i. When the sates are taken as infinitesimal lengths $\mathrm{d}L_x$ at each point x on the set of segments L of the scene and transition probabilities are given by the differential form factors $F(x, y)$, with $x, y \in L$, a continuous Markov chain with stationary distribution $w(x) = 1/L_T$ is obtained.

[3] The notion of Markov chain introduced in Sec. 1.4 is now extended in the context of a scene.
[4] We consider that in a scene always exists a path which communicates all patches (irreducibility property).
[5] The notion of continuous Markov chain for a 2D scene will be used to define 2D shape descriptors in Sec. 3.2.3.

2.2 SCENE INFORMATION CHANNEL

In this section, a scene is interpreted as a discrete information channel. This fact enables us to introduce the notions of entropy and mutual information to study the visibility and radiosity of a scene [33, 32].

2.2.1 BASIC DEFINITIONS

From the visibility point of view, the scene can be modelled in two equivalent ways, a random walk and an information channel:

- A discrete *random walk* in a discretised scene (Fig. 2.3.*a*) is a discrete stationary Markov chain (see Sec. 2.1.3) where the states correspond to the patches of a scene, the transition probabilities P_{ij} are the form factors F_{ij}, and the stationary distribution $\{w_i\}$ is given by the distribution of relative areas $\{a_i\}$ of patches, where $a_i = A_i/A_T$, being A_i the area of patch i and A_T the total area of the scene [109].

- A scene can be interpreted as a *discrete information channel* $X \rightarrow Y$ where the input and output variables take values over the set $\mathcal{S} = \{1, 2, \ldots, N_p\}$ of patches, both with the same marginal probability distribution $\{a_i\}$, and the conditional probabilities are the form factors F_{ij}. Thus, with respect to the notation introduced in Sec. 1.1, X and Y are discrete random variables with alphabet $\mathcal{S}$, $N_p = |\mathcal{S}|$, $p(x)$ and $p(y)$ are, respectively, given by the relative areas a_i and a_j (where $a_i = \Pr\{X = i\}$ and $a_j = \Pr\{Y = j\}$ with $i, j \in \mathcal{S}$), and $p(y|x)$ and $p(x, y)$ are, respectively, given by F_{ij} and $a_i F_{ij}$.

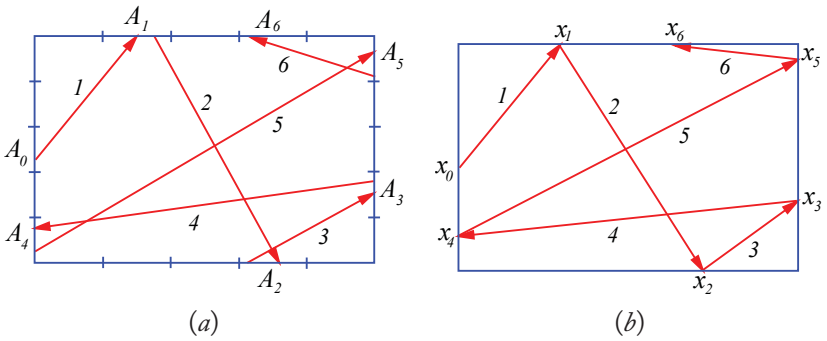


Figure 2.3: (*a*) Discrete and (*b*) continuous random walk in a scene.

From these assumptions and Equations 1.22, 1.5, 1.2, 1.4, and 1.9, Shannon's information measures can be defined for a scene.

Definition 2.1. The discrete scene visibility entropy rate (or conditional entropy), called scene visibility entropy, is defined by

$$H_S = H(Y|X) = -\sum_{i\in\mathcal{S}} a_i \sum_{j\in\mathcal{S}} F_{ij} \log F_{ij}. \tag{2.13}$$

The scene entropy can be interpreted as the average uncertainty that remains about the destination patch of a random walk (or ray) when the source patch is known. It expresses the average information content of a random walk in a scene and can be seen as the intrinsic unpredictability or the irreducible randomness associated with the chain. H_S is also the expected minimum number of bits per symbol required to code a random walk in a scene (see Equ. 1.26). Note that the Bayes theorem is expressed by the reciprocity property of the form factors (Equ. 2.5): $a_i F_{ij} = a_j F_{ji}$. From this property we obtain that $H_S = H(Y|X) = H(X|Y)$.

Definition 2.2. The discrete scene visibility positional entropy is defined by

$$H_P = H(X) = H(Y) = -\sum_{i\in\mathcal{S}} a_i \log a_i. \tag{2.14}$$

H_P expresses the average uncertainty on the position (patch) of a ray traveling an infinite random walk. It is the Shannon entropy of the stationary distribution.

Definition 2.3. The discrete scene visibility joint entropy is given by

$$H_J = H(X,Y) = -\sum_{i\in\mathcal{S}} \sum_{j\in\mathcal{S}} a_i F_{ij} \log(a_i F_{ij}). \tag{2.15}$$

H_J can be interpreted as the average uncertainty of the transition $i \to j$ of a ray in an infinite random walk. It is the Shannon entropy of a random variable with probability distribution $\{a_i F_{ij}\}$.

Definition 2.4. The discrete scene visibility mutual information is defined by

$$I_S \quad = \quad I(X;Y) = H(Y) - H(Y|X) = \sum_{i\in\mathcal{S}} \sum_{j\in\mathcal{S}} a_i F_{ij} \log \frac{F_{ij}}{a_j}. \tag{2.16}$$

I_S expresses the amount of information that the destination patch conveys about the source patch, and vice versa. It is a measure of the average information transfer or dependence between the different parts of a scene.

It is especially interesting to ask about the extremal cases of maximum and minimum scene visibility entropy, which correspond, respectively, to the maximum and minimum unpredictability in the ray path. Both cases can be illustrated with the following two examples:

- The maximum entropy is exemplified by the interior of an empty sphere[6] divided into equal area patches. In this case, all the form factors are equal ($F_{ij} = a_j$) and the uncertainty on the destination patch of a random walk is maximum: $H_S = H_P = \log |\mathcal{S}|$ (i.e., no visibility direction is privileged). Observe also that the information transfer I_S is zero for any discretisation of the sphere since this can be represented by a channel where the variables X and Y are independent (i.e., $a_i F_{ij} = a_i a_j$).

- The minimum entropy can be represented by a scene with almost touching objects. In this case, there are strongly privileged visibility directions. This system is highly correlated and the information transfer is high.

The behaviour of the entropy and mutual information is illustrated with the scenes of Fig. 2.4. In these experiments, form factors have been computed using 10^7 global lines (see Sec. 2.1.2).

In scenes with the same discretisation (Figs. 2.4.*a*-*b*), where we have a cubical enclosure with 512 interior cubes and the same H_P, observe (Table 2.1) that the increase of entropy is compensated by a mutual information decrease, and vice versa.

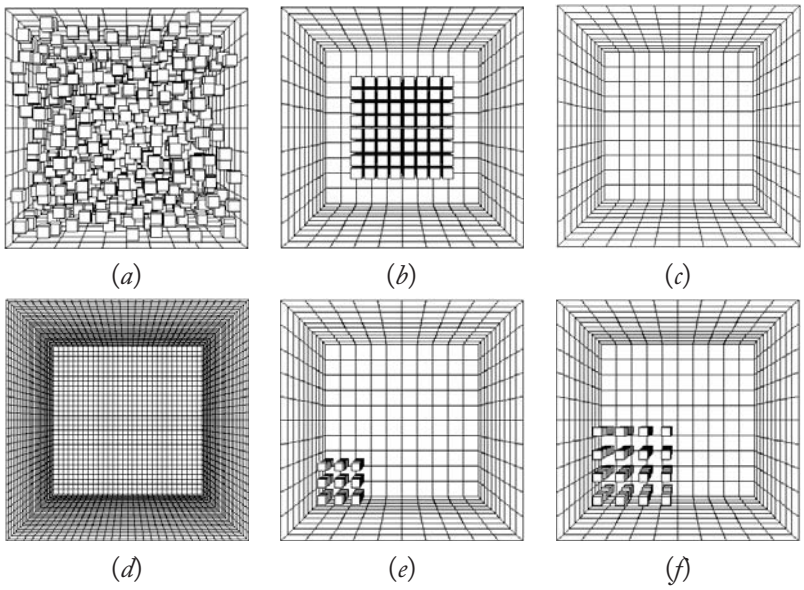


Figure 2.4: Different scene configurations with entropy and mutual information values in Table 2.1.

How does scene entropy behave with an increase in the number of patches? According to Equ. 1.35, the scene entropy goes to infinity when the number of patches also goes to infinity. On the other hand, in this case, the scene mutual information tends to a finite value. The increase of entropy and mutual information is illustrated in Figs. 2.4.*c*-*d* and Table 2.1, where we have a

[6]From Equ. 2.4, it can be seen that in a sphere $F_{ij} = a_j$.

Table 2.1: Entropy and mutual information values for scenes in Fig. 2.4.

Scene	H_S	I_S	H_P	$\overline{H}_S$
Fig. 2.4.*a*	6.761	4.779	11.541	0.586
Fig. 2.4.*b*	5.271	6.270	11.541	0.457
Fig. 2.4.*c*	7.838	1.391	9.229	0.849
Fig. 2.4.*d*	10.852	1.547	12.399	0.875
Fig. 2.4.*e*	7.589	1.861	9.450	0.803
Fig. 2.4.*f*	7.606	2.112	9.718	0.783

cubical enclosure with two different regular discretisations of their surfaces (600 and 5, 400 patches, respectively). For these two scenes, $H_P = \log N_p$, as all the patches have the same area.

In order to account for changes in the proportion of randomness and correlation in a scene, H_S and I_S can be normalized by dividing them by the positional entropy H_P.

Definition 2.5. The normalized scene visibility entropy is defined by

$$\overline{H}_S = \frac{H_S}{H_P}. \tag{2.17}$$

Definition 2.6. The normalized scene visibility mutual information is defined by

$$\overline{I}_S = \frac{I_S}{H_P} = 1 - \overline{H}_S. \tag{2.18}$$

In the literature, the normalized mutual information is considered as a measure of correlation [25]. In Table 2.1, the behavior of the normalized entropy is shown for the scenes of Fig. 2.4. Observe how $\overline{H}_S$ decreases when we introduce more cubes in the scene. This fact increases the correlation in the scene to the detriment of its randomness, in spite of the fact that H_P also increases.

2.2.2 FROM VISIBILITY TO RADIOSITY

The definitions of scene entropy and scene mutual information in Sec. 2.2.1 have been based on the existence of a Markov chain and the knowledge of its stationary distribution. Thus, to study the entropy and mutual information of a scene with illumination, we need to find an analog of the form factor matrix for the radiosity setting. This analog appears when we take the transition probabilities in a random walk solution of radiosity that lead to null variance estimators [110]:

$$P_{ij} = p_{j|i} = \frac{\rho_i F_{ij} B_j}{B_i - E_i}. \tag{2.19}$$

Then, the left eigenvector equation $\mathbf{wP} = \mathbf{w}$ (see Sec. 1.4) is used to obtain (without normalization) the stationary distribution

$$A_i \frac{B_i - E_i}{\rho_i} B_i = A_i B_i^{in} B_i^{out}, \tag{2.20}$$

where $B_i^{in} = (B_i - E_i)/\rho_i$ is the incoming radiosity and $B_i^{out} = B_i$ is the outgoing radiosity. It is easy to check that these probabilities fulfill the Bayes theorem:

$$A_i B_i F_{ij} B_j = A_j B_j F_{ji} B_i. \tag{2.21}$$

This expression is an extended reciprocity relation (without normalization). Thus, the analogy is complete.

From the above assumptions, the entropy and mutual information can be defined straightforwardly for the radiosity setting using the following substitutions:

- $A_i \to \mathsf{A}_i = A_i \frac{B_i - E_i}{\rho_i} B_i$
- $A_T \to \mathsf{A}_T = \sum_i \mathsf{A}_i$
- $a_i = \frac{A_i}{A_T} \to \mathsf{a}_i = \frac{\mathsf{A}_i}{\mathsf{A}_T}$
- $F_{ij} \to \mathsf{F}_{ij} = \frac{\rho_i F_{ij} B_j}{B_i - E_i}$.

This analogy can be interpreted as a mapping of a given scene into a new (imaginary) scene, transforming the areas and the transition probabilities according to the above formulae.

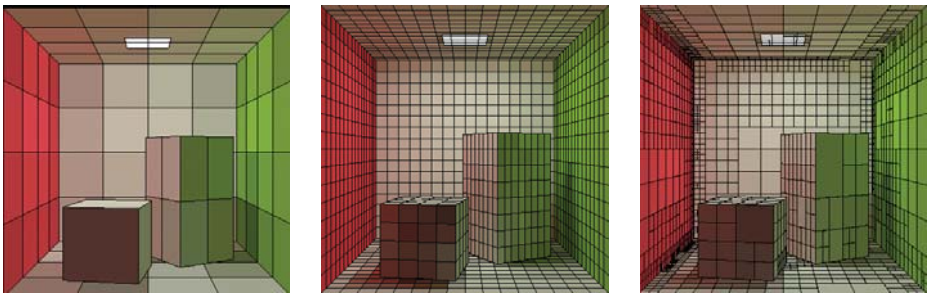
Definition 2.7. The discrete scene radiosity entropy is defined by

$$\mathsf{H}_S = -\sum_{i \in \mathcal{S}} \mathsf{a}_i \sum_{j \in \mathcal{S}} \mathsf{F}_{ij} \log \mathsf{F}_{ij}. \tag{2.22}$$

Definition 2.8. The discrete scene radiosity mutual information is defined by

$$\mathsf{I}_S = \sum_{i \in \mathcal{S}} \sum_{j \in \mathcal{S}} \mathsf{a}_i \mathsf{F}_{ij} \log \frac{\mathsf{F}_{ij}}{\mathsf{a}_j}. \tag{2.23}$$

Entropy H_S and mutual information I_S have been computed for three different discretisations of the Cornell box scene (Fig. 2.5) and for each RGB color component. In Table 2.2, it can be seen that the finer the mesh, the higher the entropy and mutual information. Observe that I_S of the scene shown in Fig. 2.5.*c* is higher than the one of Fig. 2.5.*b*. As we will see in the next sections, a higher mutual information corresponds to a more accurate mesh. Note also that, in these scenes, mutual information is very similar for all the channels, whereas entropy in the red channel is clearly the highest.



(*a*) (*b*) (*c*)

Figure 2.5: Three different discretisations for the Cornell box scene with 121, 1924, and 1924 patches, respectively. See mutual information and entropy values in Table 2.2.

Table 2.2: I_S and H_S for the Cornell box scene (Fig. 2.5).

Scene	$I_S(R, G, B)$	$H_S(R, G, B)$
Fig. 2.5.*a*	(0.935, 0.965, 0.960)	(3.138, 2.636, 2.285)
Fig. 2.5.*b*	(1.004, 1.014, 0.993)	(6.170, 5.616, 5.243)
Fig. 2.5.*c*	(1.052, 1.046, 1.020)	(6.849, 6.339, 6.059)

2.3 SCENE COMPLEXITY

This section is dedicated to scene complexity, which is interpreted as the difficulty in achieving a precise discretisation of the scene. We begin by defining the continuous scene mutual information, independent of any discretisation, as the measure for the scene complexity. Then, the relationship between the discrete and continuous mutual information is analysed and the difference between them is taken as the scene discretisation error [33, 32].

The study of complexity has multiple directions and objectives, and also many fields of application (automata, computer science, physics, biology, etc.) [3], which reflect the great activity in this area. But, what is complexity? According to W. Li [73], the meaning of this quantity should be very close to certain measures of difficulty concerning the object or the system in question: the difficulty in constructing an object, the difficulty in describing a system, the difficulty in reaching a goal, the difficulty in performing a task, and so on. Many definitions of complexity, corresponding to the different ways of quantifying these difficulties, can be found. In the two last decades, diverse complexity measures, as for instance the mutual information, have been proposed to quantify the degree of structure, dependence, or correlation of a system [50, 73, 36]. Feldman and Crutchfield [37] proposed to call them measures of statistical complexity.

Scene complexity has often been expressed as the number of patches into which a scene is subdivided. But, what do we really expect scene complexity to measure? In our context, scene complexity has to answer the question of how difficult it is to compute the visibility and radiosity of

a scene with sufficient accuracy. The difficulty in obtaining a precise illumination solution depends on the degree of dependence between all the surfaces, how the interaction between these surfaces changes in dependence when the system is subdivided, and the degree of unpredictability. The two first considerations can be represented by a statistical complexity measure, which quantifies correlation, structure, or interdependence between the parts of a system, and the third one by the entropy, which measures randomness or unpredictability. In this book, the scene complexity is associated with the mutual information and the scene randomness with the entropy.

2.3.1 CONTINUOUS SCENE VISIBILITY MUTUAL INFORMATION

A scene is a continuous system. Thus, by discretising a scene into patches, some information is lost and, consequently, a distortion or error is introduced. Obviously, the maximum accuracy of the discretisation is accomplished when the number of patches tends to infinity. Since the continuous mutual information expresses with maximum precision the information transfer or correlation in a scene, it will be considered as the measure of scene complexity.

Similarly, to the previous section, the scene is now modeled by a continuous random walk (Fig. 2.3.*b*) or by a continuous information channel. From Equ. 1.36, we know that the mutual information between two continuous random variables X and Y is the limit of the mutual information between their discretised versions, while, from Equ. 1.34, we find that the entropy of a continuous random variable does not equal the entropy of its discretised version in the limit of a finer discretisation. Thus, in a scene, discrete mutual information converges to continuous mutual information when the number of patches tends to infinity (and the size of all the patches tends to zero), while scene visibility entropy tends to infinity when the number of patches tends to infinity: $\lim_{N_p \to \infty} H_S = \infty$ (see Equ. 1.35). From now on, in this chapter, we focus attention on mutual information.

As we have seen in Sec. 2.1.3, when the states form an uncountable set, we deal with a continuous Markov chain. The continuous mutual information of a scene can be obtained from the discrete mutual information (Equ. 2.16) using the following substitutions:

- Each state is substituted by an infinitesimal area and each summatory by an integral.
- $w_i = A_i / A_T \to 1/A_T$. This means substituting the discrete probability of taking patch i by the continuous probability of selecting any point.
- $F_{ij} \to F(x, y)$. This means substituting the patch-to-patch form factor by the point-to-point form factor.

The scene continuous mutual information can also be obtained from the continuous mutual information (Equ. 1.30) by using the following changes:

- $dx \to \mathrm{d}A_x, dy \to \mathrm{d}A_y$
- $f(x) \to \frac{1}{A_T}$

- $f(y|x) \rightarrow F(x, y)$
- $f(x, y) \rightarrow \frac{1}{A_T} F(x, y)$.

Thus, the continuous mutual information for a scene can be defined as follows:

Definition 2.9. The continuous scene visibility mutual information is defined by

$$I_S^c = \int_{\mathcal{S}} \int_{\mathcal{S}} \frac{1}{A_T} F(x, y) \log(A_T F(x, y)) \mathrm{d}A_x \mathrm{d}A_y, \tag{2.24}$$

where $\mathcal{S}$ represents the set of surfaces of the scene. For instance, in the interior of an empty sphere,

where $F(x, y) = \cos\theta_x \cos\theta_y / (\pi r_{xy}^2) = 1/A_T$ (see Equ. 2.2), the result obtained is, as expected, $I_S^c = 0$. Remember that, in a sphere, $I_S = 0$ and, thus, $I_S^c = I_S = 0$.

Like continuous visibility mutual information I_S^c, that has been derived from discrete visibility mutual information I_S, continuous radiosity mutual information could also be obtained from discrete radiosity mutual information I_S(Equ. 2.23). For more details, see [32].

2.3.2 COMPUTATION OF SCENE VISIBILITY COMPLEXITY

In this section, we show how the continuous mutual information can be computed by Monte Carlo using local or global lines (see Sec. 2.1.2). Reparametrizing the continuous mutual information integral, we obtain

$$\begin{aligned} I_S^c &= \int_{\mathcal{S}} \int_{\mathcal{S}} \frac{1}{A_T} F(x, y) \log(A_T F(x, y)) \mathrm{d}A_x \mathrm{d}A_y \\ &= \int_{\mathcal{S}} \int_{\Omega_x} \frac{1}{A_T} \frac{\cos\theta_x}{\pi} \log(A_T F(x, y(x, \omega_x))) \mathrm{d}A_x \mathrm{d}\omega_x, \end{aligned} \tag{2.25}$$

where $\mathrm{d}\omega_x = (\cos\theta_y / r_{xy}^2)\mathrm{d}A_y$ and $y(x, \omega_x)$ is the point visible from x in the direction ω_x. We can now take $\cos\theta_x/(\pi A_T)$ as probability density function ($\int_{\mathcal{S}} \int_{\Omega_x} \cos\theta_x/(\pi A_T)\mathrm{d}A_x \mathrm{d}\omega_x = 1$). Drawing samples according to this distribution means simply selecting first a random point in the scene upon the area and a direction upon the form factor distribution. Alternatively, we can directly use the density $F(x, y)/A_T$, which is achieved by casting global lines. The result obtained is

$$I_S^c \approx \frac{1}{N} \sum_{k=1}^{N} \log(A_T F(x_k, y_k(x_k, \omega_{x_k})) = \frac{1}{N} \sum_{k=1}^{N} \log\left(\frac{A_T \cos\theta_{x_k} \cos\theta_{y_k}}{\pi r_{x_k y_k}^2}\right). \tag{2.26}$$

In the case of using global lines, N stands for the total number of segments of the global lines or the number of pairs of points considered, which is the total number of intersections divided by two (see Fig. 2.6). In the case of using local lines, N represents the total number of local lines used in a scene. The quantity N_i of local lines cast from patch i is proportional to its area: $N_i = (A_i/A_T)N$.

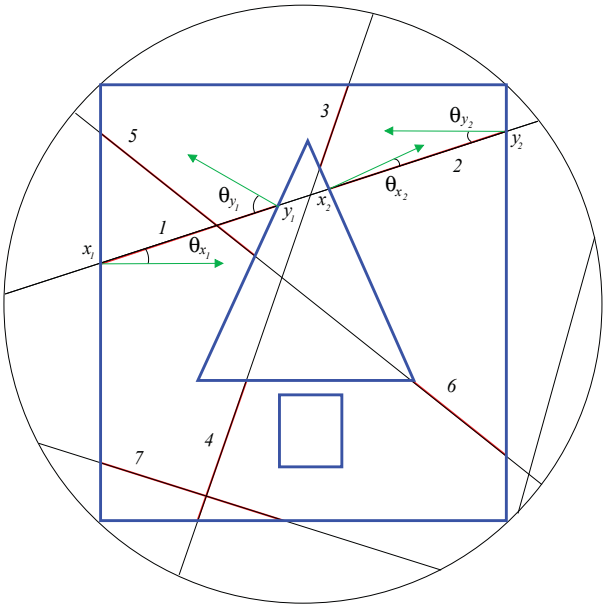


Figure 2.6: 2D representation of a scene where the segments of some global lines used to compute the continuous mutual information are shown.

The complexity of platonic solids and Cornell box[7] (Fig. 2.5) has been computed using global lines. In Table 2.3, we can observe that, among the platonic solids, the minimum complexity corresponds to a sphere and the maximum complexity to a tetrahedron. As we expected, the polyhedra that are nearer to the sphere are less complex, that is, they have less correlation. The complexity of the Cornell box is clearly greater than the one of the empty cube, since the introduction of objects in the interior of an empty scene increases its complexity.

2.3.3 COMPLEXITY AND DISCRETISATION

The relationship between the discrete and continuous mutual information is now analyzed and the scene complexity I_S^c is shown to be closely related to the difficulty in obtaining an accurate discretisation. In a way, to discretise a scene is to model it. "A system is not complex by some abstract criterion but because it is intrinsically hard to model" [3]. This point of view is compatible with W. Li's comment: "An intuitively satisfactory definition of complexity should measure the amount of effort put in that generates correlations in a sequence. Of course, one cannot be sure that all the effort is spent on generating correlations. As a result, a measure of correlation typically provides a lower bound of a measure of complexity, and might be a reasonable estimate of the complexity" [73].

[7] In our version of the Cornell box scene, prism and cube are slightly separated from the floor. This fact increases the scene visibility complexity since in the narrow spaces the correlation is higher.

Table 2.3: Complexity of sphere, platonic solids, and Cornell box.

Scene	I_S^c	Figure
Sphere	0	
Icosahedron	0.543	Fig. 3.2.*e*
Dodecahedron	0.825	Fig. 3.2.*d*
Octahedron	1.258	Fig. 3.2.*c*
Cube	1.609	Fig. 3.2.*b*
Tetrahedron	2.626	Fig. 3.2.*a*
Cornell box	3.274	Fig. 2.7.*a*

In Sec. 1.6, we have presented the following properties: the mutual information between two continuous random variables is the limit of the mutual information between their discretised versions, refinement can never decrease the discrete mutual information, and the continuous mutual information is the least upper bound for the discrete mutual information. If we now apply these properties to scene visibility, we find that

- If any patch is divided into two or more patches, the discrete mutual information I_S of the new scene increases or remains the same.
- The continuous scene visibility mutual information is the least upper bound to the discrete scene visibility mutual information.

Thus, a scene fulfills

$$I_S^c - I_S \geq 0. \tag{2.27}$$

This difference expresses the loss of information transfer due to the discretisation. Thus, between different discretisations of the same scene, we can consider that the most precise will be the one that has a higher discrete mutual information I_S, that is, the one that best captures the information transfer. With this in mind, the discretisation error can be defined.

Definition 2.10. The scene discretisation error δ^v is defined[8] by

$$\delta^v = I_S^c - I_S, \tag{2.28}$$

and the relative scene discretisation error as the quotient

$$\overline{\delta}^v = \frac{I_S^c - I_S}{I_S^c}. \tag{2.29}$$

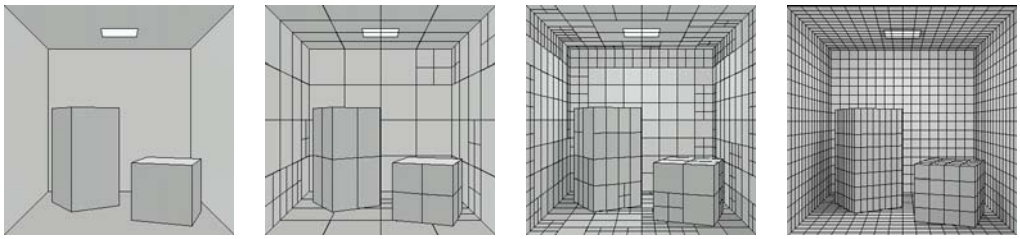
[8]The superindex v stands for *visibility*, meaning that only visibility criteria are taken into account.

In a complementary way, the relative discretisation accuracy is given by I_S/I_S^c.

From the following experiments, we can see that continuous mutual information I_S^c is closely related with the difficulty in obtaining an accurate discretisation: the higher the I_S^c (i.e., when there is more information transfer in a scene), the more difficult it is to obtain an accurate discretisation, and probably more refinements will be necessary to achieve a given precision. From this point of view, the difficulty in discretising the interior of an empty sphere is null (the discretisation error is always equal to zero).

Table 2.4: Results for the cubical enclosure of Figs. 2.4.*c*-*d* with two different discretisations of their surfaces. For each scene, 10^5, 10^7, and 10^9 global lines have been cast.

Scene	Lines	H_S	I_S	I_S^c
	10^5	6.482	2.747	1.610
Fig. 2.4.*c*	10^7	7.821	1.408	1.612
	10^9	7.838	1.391	1.610
	10^5	4.313	8.086	1.610
Fig. 2.4.*d*	10^7	9.684	2.715	1.611
	10^9	10.852	1.547	1.610



(*a*) (*b*) (*c*) (*d*)

Figure 2.7: Four different discretisations for the Cornell box. The total number of patches is, respectively, 19, 196, 826, and 1, 924. Mutual information results are shown in Table 2.5.

The above properties and assumptions are tested on different scene configurations (Fig. 2.7, Fig. 2.8, and Fig. 2.9). As we can see in Table 2.4, corresponding to Figs. 2.4.*c*-*d*, the computational cost of I_S^c is much lower than the cost of computing I_S. That is, I_S^c can be computed with enough precision using a relatively small number of lines, while the computation of I_S requires much more lines to get a precise measurement. Observe that I_S increases with the number of patches but has to be always less than I_S^c when enough precision for I_S is obtained. We can also see that, due to

Table 2.5: I_S^c and I_S for the scenes in Fig. 2.7.

Scene	Patches	Lines	I_S	I_S^c
Fig. 2.7.*a*	19	10^7	0.690	3.273
Fig. 2.7.*b*	196	10^7	2.199	3.273
Fig. 2.7.*c*	826	10^7	2.558	3.273
Fig. 2.7.*d*	1924	10^8	2.752	3.273

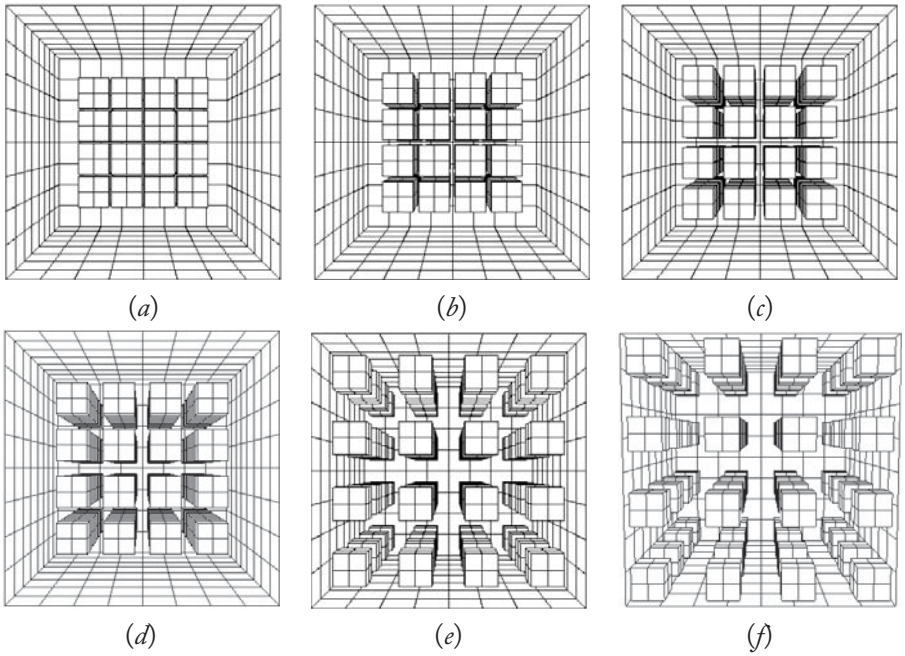


Figure 2.8: 64 cubes are grouped very closely together in the center of the cubical enclosure and then are separated and moved outwards until they almost touch the walls. The discretisation of the cubes (1, 536 patches) is finer than the discretisation of the walls (384 patches).

the Monte Carlo integration error, the value of the discrete mutual information decreases (until convergence is achieved) with the increase in the number of lines cast. In Table 2.5, corresponding to Fig. 2.7, we also show how discrete mutual information I_S increases with the mesh refinement.

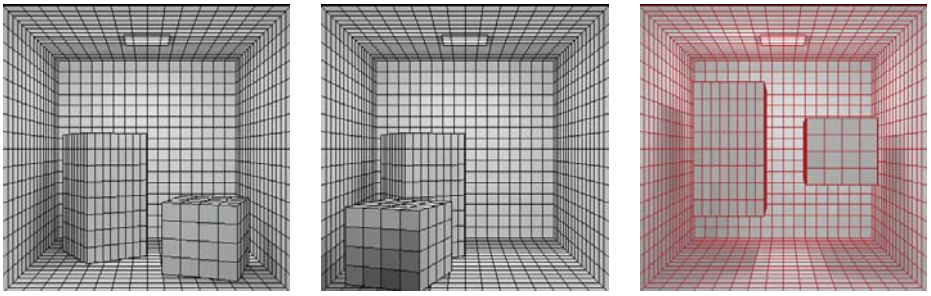
In the following experiments, we find that, for a regular discretisation, the relative discretisation error is lower in the less complex scenes. This hypothesis is analysed from the results shown in Table 2.6, which have been obtained from the scenes of Fig. 2.8. Initially, 64 cubes are grouped very closely together in the center of the cubical enclosure and, little by little, they are separated and moved outwards until they almost touch the walls. In this sequence of scenes, the discretisation of the cubes is finer than the discretisation of the walls. The relative discretisation accuracy appears to

Table 2.6: Results for the scenes of Fig. 2.8. For each scene, 10^8 global lines have been cast.

Scene	(*a*)	(*b*)	(*c*)	(*d*)	(*e*)	(*f*)
I_S	5.492	5.054	4.672	4.395	4.356	4.775
I_S^c	6.430	5.678	5.177	4.867	5.015	6.055
I_S/I_S^c	0.854	0.890	0.902	0.903	0.869	0.789

be higher in the "middle" scenes (Figs. 2.8.*b*–*e*), and lower in the "extremal" scenes (Fig. 2.8.*a* and Fig. 2.8.*f*). These last two scenes are the most complex scenes and, consequently, should have a finer discretisation in order to obtain greater accuracy. In other words, these scenes are the ones most difficult to discretise. Note also that the relative discretisation accuracy of the scene of Fig. 2.8.*a* is higher than the one of Fig. 2.8.*f*. This is due to the fact that the discretisation is finer in the narrow spaces between the cubes. In contrast, when the cubes are near the walls, a greater precision would be obtained when the discretisation of the walls were finer.

In Table 2.7, corresponding to Fig. 2.9, we also observe that the relative discretisation error is higher for more complex scenes. All these experiments suggest that the discretisation error may be used to choose the best discretisation from several alternatives.



(*a*) (*b*) (*c*)

Figure 2.9: Three different scenes with a regular discretisation of their surfaces and the same number of patches (1, 924).

Table 2.7: I_S, I_S^c, and relative discretisation error $\overline{\delta}^v$ for the scenes in Fig. 2.9. For each scene, 10^8 local lines have been cast.

Scene	Fig. 2.9a	Fig. 2.9b	Fig. 2.9c
I_S	2.752	2.823	2.459
I_S^c	3.274	3.375	2.613
$\overline{\delta}^v$	0.159	0.164	0.059

2.4 REFINEMENT CRITERION BASED ON MUTUAL INFORMATION

As we have mentioned in Sec. 2.1.1, scene discretisation has to accurately represent illumination variations, but it has to avoid unnecessary refinements that would increase the computational cost. To achieve this objective, in this section, we introduce a mutual-information-based oracle for hierarchical radiosity, which is based on the loss of visibility information transfer between two patches due to the discretisation [32].

2.4.1 LOSS OF INFORMATION TRANSFER DUE TO DISCRETISATION

To obtain a refinement criterion for hierarchical radiosity, we calculate the difference between both, continuous and discrete, patch-to-patch visibility information transfers. From Equ. 2.16, the term

$$I_{ij} = a_i F_{ij} \log\left(\frac{F_{ij}}{a_j}\right), \tag{2.30}$$

can be considered as an element of a mutual information matrix, and it is easy to see that $I_{ij} = I_{ji}$. Each element represents the information transfer between patches i and j. We can also consider that

$$I_i = a_i \sum_{j\in\mathcal{S}} F_{ij} \log\left(\frac{F_{ij}}{a_j}\right), \tag{2.31}$$

expresses the information transfer from patch i. Taking into account that the summation is the Kullback-Leibler distance between $\{F_{ij}\}$ and $\{a_j\}$ with j ranging in the set of patches, it is easy to see that $I_i \geq 0$. From Equ. 2.16, we can write

$$I_S = \sum_{i\in\mathcal{S}} I_i = \sum_{i\in\mathcal{S}} \sum_{j\in\mathcal{S}} I_{ij}. \tag{2.32}$$

The information transfer between two patches can be accurately calculated if we consider the continuous mutual information between them. Thus, from the continuous visibility mutual information (Equ. 2.24), we get

$$I^c_{ij} = \int_{S_i}\int_{S_j} \frac{1}{A_T} F(x,y) \log(A_T F(x,y)) \mathrm{d}A_x \mathrm{d}A_y. \tag{2.33}$$

This continuous measure expresses with maximum precision the visibility information transfer between two elements, and it can be computed with an area-to-area sampling (see Sec. 2.1.2), that is, using random lines joining both elements i and j (the pdf is $1/(A_i A_j)$). For N_{ij} lines, we have

$$I^c_{ij} \approx \frac{A_i A_j}{A_T} \frac{1}{N_{ij}} \sum_{k=1}^{N_{ij}} F(x_k, y_k) \log\left(F(x_k, y_k) A_T\right), \tag{2.34}$$

where x_k and y_k are, respectively, the end-points on patches i and j of the k-th line.

From Equ. 2.30, I_{ij} can be rewritten as

$$I_{ij} = \frac{A_i F_{ij}}{A_T} \log\left(\frac{F_{ij} A_T}{A_j}\right) = \frac{A_i A_j}{A_T} \frac{F_{ij}}{A_j} \log\left(\frac{F_{ij}}{A_j} A_T\right). \tag{2.35}$$

Thus, taking $F_{ij}/A_j \approx (1/N_{ij}) \sum_{k=1}^{N_{ij}} F(x_k, y_k)$ (Equ. 2.10), we obtain the visibility discretisation error between patches i and j:

$$\begin{aligned} \delta^v_{ij} = I^c_{ij} - I_{ij} \approx & \frac{A_i A_j}{A_T}\left(\frac{1}{N_{ij}}\left(\sum_{k=1}^{N_{ij}} F(x_k, y_k) \log(F(x_k, y_k))\right)\right. \\ & \left. - \left(\frac{1}{N_{ij}} \sum_{k=1}^{N_{ij}} F(x_k, y_k)\right) \log\left(\frac{1}{N_{ij}} \sum_{k=1}^{N_{ij}} F(x_k, y_k)\right)\right) \geq 0. \end{aligned} \tag{2.36}$$

The positivity of δ^v_{ij} is obtained from the log-sum inequality (Equ. 1.15). This difference gives us the discretisation error between two elements, and it is used as the basis for the mutual-information-based oracle. Observe also that δ^v_{ij} is symmetric: $\delta^v_{ij} = \delta^v_{ji}$. As we expected, it is easy to see that the discretisation error between two spherical patches is equal to zero.

2.4.2 MUTUAL-INFORMATION-BASED ORACLE FOR HIERARCHICAL RADIOSITY

As the refinement strategy in hierarchical radiosity deals with one pair of elements at a time, an oracle based on the discretisation error between two patches or elements (Equ. 2.36) is introduced.

The fundamental idea is that the difference between continuous and discrete patch-to-patch (or element-to-element) mutual information gives us the loss of information transfer or, equivalently, the maximum potential gain of information transfer between two elements. Hence, this difference can be interpreted as the benefit to be gained by refining and can be used as a decision criterion.

To create the mutual-information-based oracle (MI oracle), we take a similar approach to the classic smoothness-based oracles, which multiply $\rho_i B_j$ (from the radiosity equation 2.3) by an expression of the visibility gradient between the two patches involved. In our case, the visibility gradient is substituted by the discretisation error $\delta^v_{ij} = I^c_{ij} - I_{ij}$, which, in a way, also represents the variation of the radiosity kernel. Thus, the MI oracle is based on the following two considerations:

- In the radiosity equation (Equ. 2.3)

$$B_i = E_i + \rho_i \sum_{j=1}^{N_p} F_{ij} B_j,$$

the contribution of patch j to the radiosity of patch i is given by $\rho_i F_{ij} B_j$. That is, the geometric factor (radiosity kernel) is weighted by $\rho_i B_j$.

- The kernel-smoothness-based oracles, such as [121]

$$\rho_i (F_{ij}^{max} - F_{ij}^{min}) A_j B_j < \epsilon \tag{2.37}$$

and [48]

$$\rho_i \max(F_{ij}^{max} - F_{ij}^{av}, F_{ij}^{av} - F_{ij}^{min}) A_j B_j < \epsilon, \tag{2.38}$$

try to capture the variation of the radiosity kernel using the maximum and minimum kernel values.

Taking these two facts on board, that is, weighting the variation of the radiosity kernel (expressed by the visibility discretisation error δ_{ij}^v between two patches) by $\rho_i B_j$ and using the same oracle scheme, the MI oracle is given by

$$\rho_i \delta_{ij}^v B_j < \epsilon, \tag{2.39}$$

which can be computed with N_{ij} element-to-element random lines between elements i and j:

$$\frac{\rho_i A_i A_j B_j}{A_T} \left(\frac{1}{N_{ij}} \left(\sum_{k=1}^{N_{ij}} F(x_k, y_k) \log(F(x_k, y_k)) \right) - \left(\frac{1}{N_{ij}} \sum_{k=1}^{N_{ij}} F(x_k, y_k) \right) \log\left(\frac{1}{N_{ij}} \sum_{k=1}^{N_{ij}} F(x_k, y_k) \right) \right) < \epsilon. \tag{2.40}$$

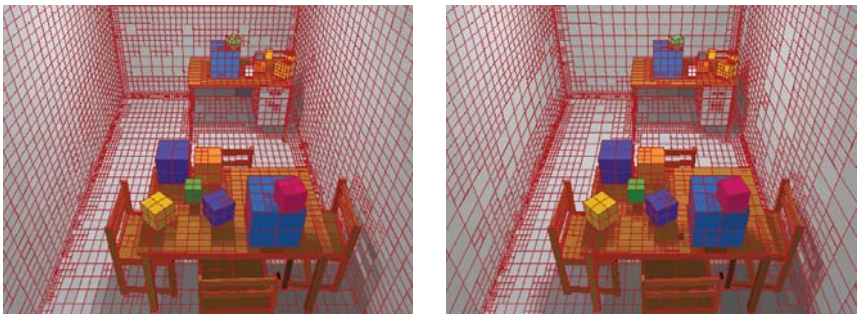
Observe that in this expression the receiver area appears weighting the oracle and thus avoiding an excessively small receiver subdivision.

The performance of the MI oracle is compared with a classic kernel-smoothness-based (KS) oracle (Equ. 2.38) using the hierarchical Monte Carlo radiosity method [5][9]. It has to be noted that these oracles can be used with any hierarchical radiosity method.

Fig. 2.10 and Fig. 2.11 show the behaviour of the KS and MI oracles for a given scene. Observe the accurate representation of the shadow of the chair near the right wall (Fig. 2.10.*ii*) and front wall (Fig. 2.11.*ii*) obtained by the MI oracle. Observe also the much better discrimination in the mesh, seen for instance on the floor and walls, and how the shadows on the table are represented more accurately in Fig. 2.11.*ii*. Fig. 2.12 shows a more accurate solution obtained with the MI oracle and computed with 10 element-to-element random lines for each oracle evaluation and 2, 684, 260 rays for radiosity computation.

In Rigau et al. [99, 105], the MI oracle has been generalized using the Harvda-Charvát-Tsallis (HCT) mutual information (Equ. 1.47), so that the discretisation error has been computed as the difference between the continuous and discrete Harvda-Charvát-Tsallis mutual information. It is interesting to note that, for entropic indexes α less than one, the HCT oracle produces more refinements than the MI oracle in the most shadow areas and less refinements in the corners of the scene, and vice versa for entropic indexes greater than one (see Fig. 2.13). Note that, when $\alpha = 1$, the HCT oracle is equivalent to the MI oracle. More details can be seen in [99, 105].

[9]The hierarchical Monte Carlo radiosity algorithm has been implemented in the RenderPark system [23].



(*a.i*) KS (*a.ii*) MI

(*b.i*) KS (*b.ii*) MI

Figure 2.10: Results obtained by (*i*) KS and (*ii*) MI oracles with the test scene showing the (*a*) mesh and (*b*) Gouraud shaded solution. The number of patches is (*i*) 13, 902 and (*ii*) 13, 878, respectively, and 402, 650 rays have been used for radiosity computation and 10 rays for each oracle evaluation between two elements.

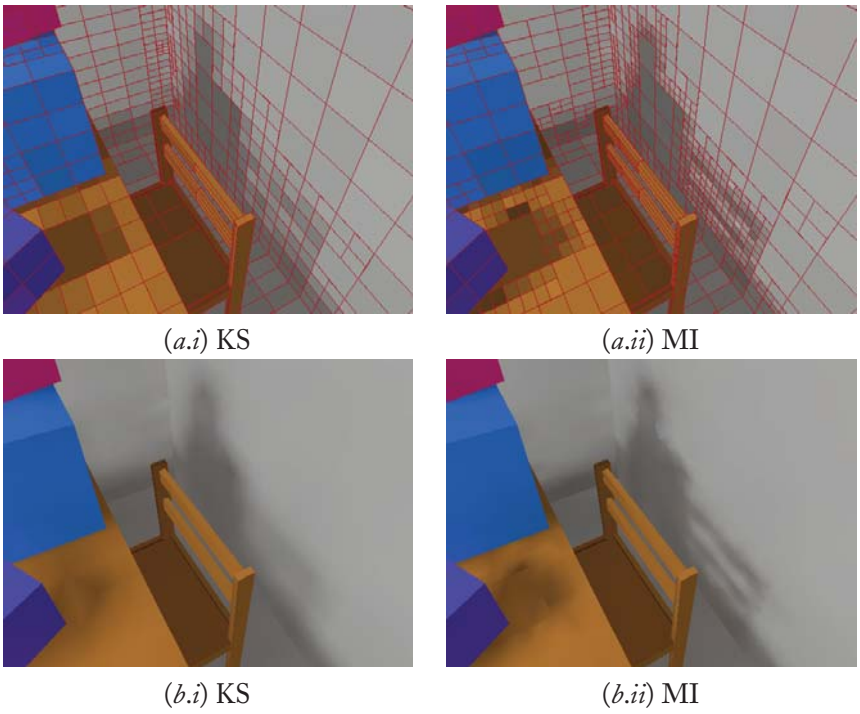
2.5 REFINEMENT CRITERIA BASED ON *f*-DIVERGENCES

In this section, *f*-divergence-based oracles for hierarchical radiosity refinement are presented [103]. The introduction of these refinement criteria was motivated by the observation that the MI oracle presented in Sec. 2.4 could be rewritten as an *f*-divergence.

The discretisation error (Equ. 2.36) can be written in the following way:

$$\begin{aligned}\delta_{ij}^{v} &\approx \frac{A_i A_j}{A_T}\widehat{F}\Big(\frac{1}{N_s}\sum_{k=1}^{N_s} p_k \log p_k - \frac{1}{N_s}\sum_{k=1}^{N_s} p_k \log(\frac{1}{N_s}\sum_{k=1}^{N_s} p_k)\Big) \\ &= \frac{A_i A_j}{A_T}\widehat{F}\Big(\frac{1}{N_s}\sum_{k=1}^{N_s} p_k \log p_k - \frac{1}{N_s}\log\frac{1}{N_s}\Big) \geq 0 \ ,\end{aligned} \tag{2.41}$$

where $\widehat{F} = \sum_{k=1}^{N_s} F(x_k, y_k)$, $p_k = F(x_k, y_k)/\widehat{F}$ for all $1 \leq k \leq N_s$, and $1/N_s \sum_{k=1}^{N_s} p_k = 1/N_s$.



(*a.i*) KS (*a.ii*) MI

(*b.i*) KS (*b.ii*) MI

Figure 2.11: (*a*) Mesh and (*b*) Gouraud shaded solution for a different view of the test scene of Fig. 2.10 obtained by (*i*) KS and (*ii*) MI oracles.

It can be seen that the expression between parentheses in Equ. 2.41, except for a constant factor $1/N_s$, is the Kullback-Leibler distance between the distributions $p_k = F(x_k, y_k)/\widehat{F}$ and $q_k = 1/N_s$. Thus, we get

$$\delta_{ij}^{v} \approx \frac{A_i A_j}{A_T} \frac{1}{N_s} \widehat{F}\, D_{KL}(p, q) \quad . \tag{2.42}$$

This fact suggests to try other *f*-divergences in the kernel of the refinement oracle (Equ. 2.39). These measures will give us the variability of the distribution $\{F(x_1, y_1)/\widehat{F}, \ldots, F(x_{N_s}, y_{N_s})/\widehat{F}\}$ with respect to the uniform distribution $\{1/N_s, \ldots, 1/N_s\}$.

The Kullback-Leibler (Equ. 1.7), Chi-square (Equ. 1.42), and Hellinger (Equ. 1.43) distances have been tested. Note that the Kullback-Leibler-based oracle coincides with the MI oracle (Equ. 2.39).

The oracles used in the test are the following:

- Kullback-Leibler (KL)

$$\rho_i A_i A_j \widehat{F}\, D_{KL}(p, q) B_j < \epsilon. \tag{2.43}$$

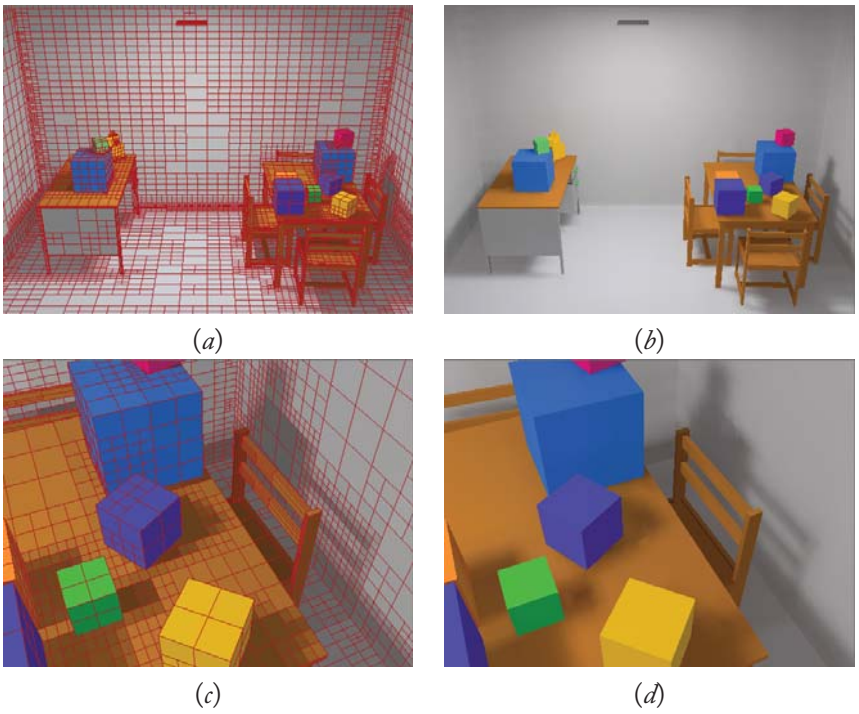


Figure 2.12: Results obtained with the MI method for the scene shown in Fig. 2.10. (*a*) and (*c*) show the grids corresponding to (*b*) and (*d*) Gouraud shaded solutions. The number of patches is 18, 338 and 2, 684, 260 rays have been used for radiosity computation and 10 rays for each oracle evaluation.

- Chi-square (CS)

$$\rho_i A_i A_j \widehat{F} D_{\chi^2}(p,q) B_j < \epsilon. \tag{2.44}$$

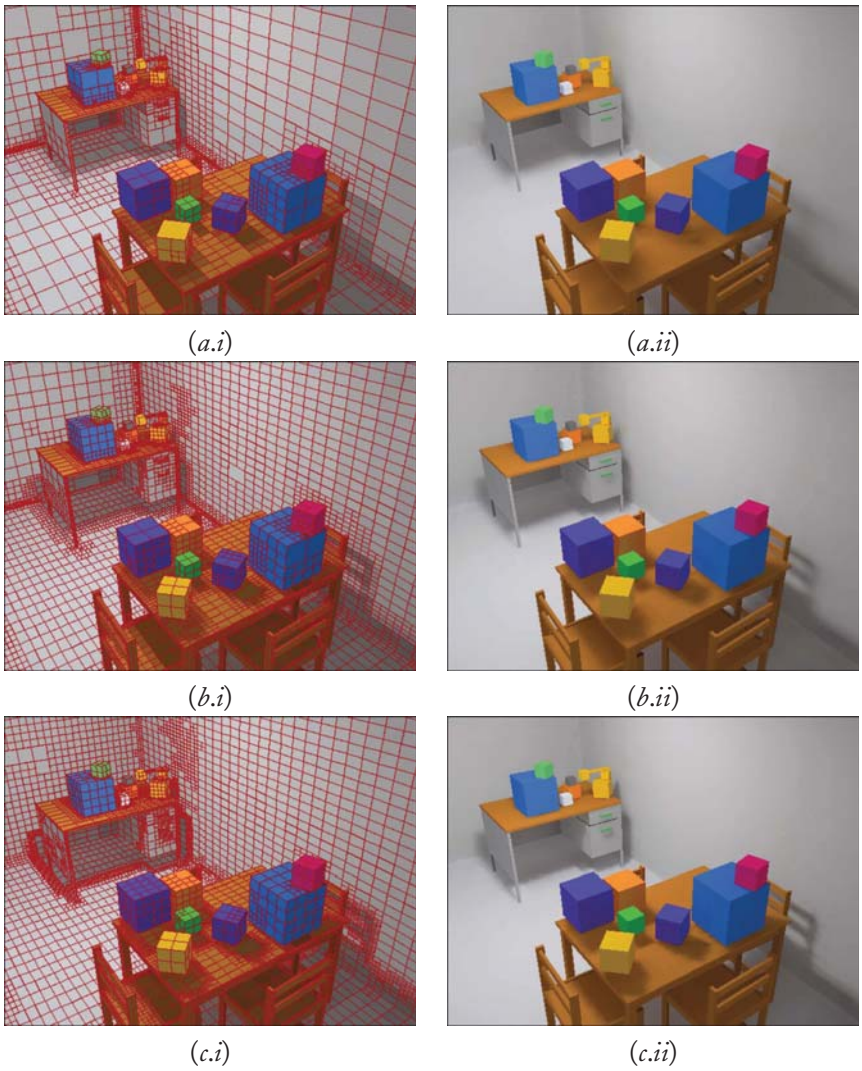
- Hellinger (HE)

$$\rho_i A_i A_j \widehat{F} D_{h^2}(p,q) B_j < \epsilon \ , \tag{2.45}$$

based all on their respective distances. Observe that the constants $1/A_T$ and $1/N_s$ have been removed.

It is important to note that the expression between parentheses in Equ. 2.41 is equal to the first term in Jensen's inequality (Equ. 1.13) with $f(x) = x \log x$ and $x = F(x, y)/\widehat{F}$. Moreover, we can also see that this expression is equal to the first term in the log-sum inequality (Equ. 1.15), taking $b_i = 1$ and $a_i = F(x_i, y_i)/\widehat{F}$.

The KS oracle and the f-divergence-based oracles have been implemented in the hierarchical Monte Carlo radiosity method [5]. In Fig. 2.14, we present the results obtained with the KS oracle (Fig. 2.14.*a*) and the f-divergence-based ones (Figs. 2.14.*b-d*) for a view of the test scene. Left column (*i*) shows the subdivision obtained, while the right one (*ii*) corresponds to the Gouraud



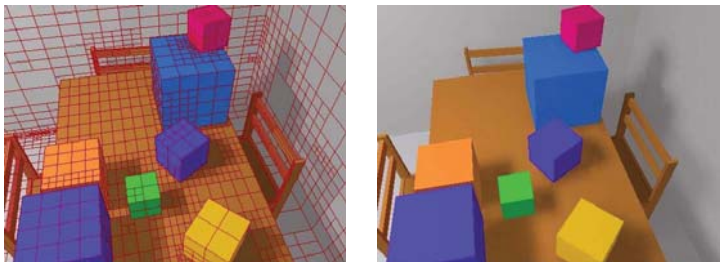
(*a.i*) (*a.ii*) (*b.i*) (*b.ii*) (*c.i*) (*c.ii*)

Figure 2.13: (*a*) Mesh and (*b*) Gouraud shaded solution for the test scene of Fig. 2.10 obtained by the Harvda-Charvát-Tsallis mutual information oracle using entropic indexes (*a*) α=1.5, (*b*) α=1, and (*c*) α=0.5.

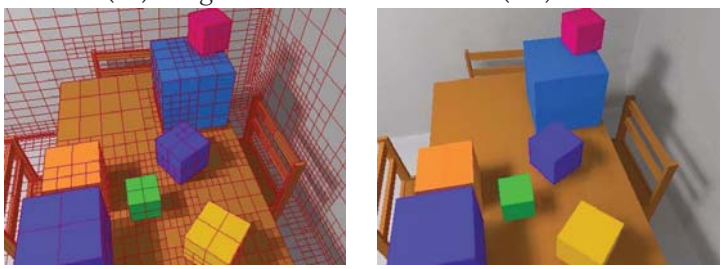
shaded solution. Each oracle has been evaluated with 10 random lines between the corresponding pair of elements and a total of 2, 685, 000 rays have been cast for the radiosity computation. The ϵ parameter has been tuned so that the grids obtained have approximately 19, 000 patches in all the methods.

In Figs. 2.14.*b-d*, we can see how the *f*-divergence-based oracles outperform the KS one (Fig. 2.14.*a*), especially in the much more-defined shadow of the chair and the cubes on the right wall. Observe also the superior quality of the grid created on top of the table, and in the corner between the walls.

On the other hand, comparing the three *f*-divergence oracles, we conclude that, although they exhibit a similar quality, the KL (or MI) one is slightly better. For instance, observe how the shadows on the table are more defined. A possible explanation for this better behaviour could be that the KL oracle, unlike the other ones, meets Jensen's inequality (Equ. 1.13). This might confer a distinctive theoretical advantage to the Kullback-Leibler oracle.



(*a.i*) KS grid (*a.ii*) KS



(*b.i*) KL grid (*b.ii*) KL

(*c.i*) CS grid (*c.ii*) CS

(*d.i*) HE grid (*d.ii*) HE

Figure 2.14: A view of the test scene for comparison of (*a*) kernel-smoothness-based (KS) and *f*-divergence-based oracles: (*b*) Kullback-Leibler (KL), (*c*) Chi-square (CS), and (*d*) Hellinger (HE). Column (*i*) shows the grid obtained in the refinement process and column (*ii*) shows the Gouraud shaded solution. In all methods, the oracles have been evaluated with 10 random lines between two elements. In each case, a total of 2, 685, 000 rays are cast for the radiosity computation, obtaining approximately 19, 000 patches. ([103] © The Eurographics Association, 2003.)

CHAPTER 3

Shape Descriptors

In the last years, shape complexity has been analysed from different fields, such as computer vision [86] and psychology [38]. The benefits of a shape complexity theory (with its corresponding measures) would range from object classification for posterior database retrieval to improvements in cognitive science. As we have seen in Sec. 2.3, mutual information has been used to quantify the complexity of a scene and, consequently, can also be used to measure the shape complexity of an object.

In this chapter, we propose two shape complexity measures based on the continuous mutual information which are independent of the discretisation and appropriate to describe the shape of any object [104]. Thus, shape complexity will be analysed from two different perspectives. First, from the inside of the object, its degree of structure (interdependence between its parts) is quantified. In this case, we measure the information shared by the interior surfaces of the object. A differential of surface will be related to another differential of surface by the uniformly distributed lines [108] that join them, that is, make them visible to each other. Second, from the outside of the object, the degree of interaction between the object and its circumscribing sphere is calculated using the same uniformly distributed lines. These complexity measures can be used as shape descriptors in fields such as object recognition and classification.

3.1 BACKGROUND

In Sec. 2.1.2 and Sec. 2.3.2, we have seen that the density of uniformly distributed lines crossing two differential areas $\mathrm{d}A_x$, $\mathrm{d}A_y$, centered at inner points x and y, is given by $\mathrm{d}G = F(x, y)\mathrm{d}A_x\mathrm{d}A_y$, where $F(x, y)$ is the point-to-point form factor defined in Sec. 2.1.1 and equal to $\cos\theta_x \cos\theta_y/(\pi r_{xy}^2)$ for mutually visible points, or zero otherwise (Equ. 2.2), where θ_x and θ_y are the angles which the normals at x and y form with the segment joining them, and r_{xy} is the distance between x and y (Fig. 3.1).

Global lines intersect an object forming random chords, that can be used to measure the visibility within a body (Fig. 3.1). Visibility directions must be homogeneous and isotropic over the body, a quality fulfilled by the uniform line density. It can be considered that the form factor measures this visibility. In Sec. 2.3, the scene visibility complexity has been computed using global lines. Interestingly, the global line density is related to the curvature function through an integral relation [107]. For an object K, we have that

$$\int_{K\cap G\neq\emptyset} (\sum_{i=1}^{n} \kappa_i)\mathrm{d}G = Cc, \tag{3.1}$$

where κ_i is the curvature (2D) or Gauss curvature (3D) at the ith intersection point (out of n) of a line G and the object K, and $C = 2$ for a planar object and $C = \pi$ for a 3D-object, being c the total curvature[1]. That is, the integral of the sum of curvatures at the intersection points along a global line is equal to C times total curvature.

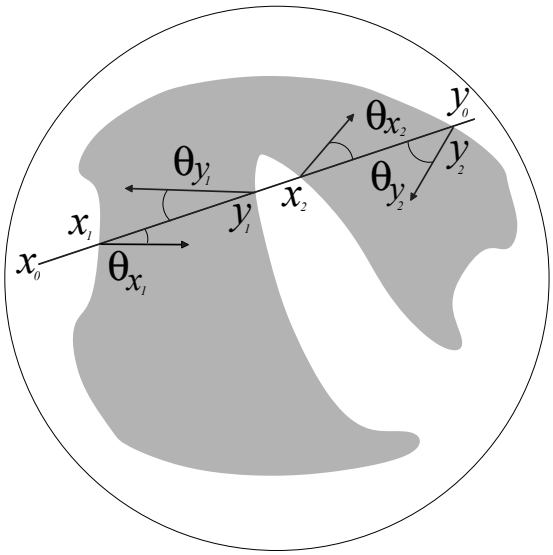


Figure 3.1: Geometry of one global line (x_0, y_0) that generates five random chords in a 3D-shape: (x_0, x_1), (x_1, y_1), (y_1, x_2), (x_2, y_2), and (y_2, y_0).

3.2 INNER SHAPE COMPLEXITY

In this section, the shape complexity of an object is analysed from its interior. In Sec. 2.3, continuous mutual information has been introduced as a complexity measure in order to evaluate the difficulty of discretising a scene and to obtain a refinement criterion for hierarchical radiosity (Sec. 2.1.1). Here, continuous mutual information is proposed to measure the shape complexity (Equ. 1.30).

3.2.1 COMPLEXITY MEASURE

The approaches taken by Page [86] and Feldman [38] consider the information or entropy contained in the curvature of an object. In [86], the quantity measured depends on both the size and positioning of the (regular) discretisation of the curve or surface. A discretisation that misses some corner would give an incorrect measurement. Moreover, the entropy is a function that diverges with the number of discretisation bins (see Equ. 1.35). On the other hand, in [38], the curvature distribution is a

[1] The total curvature is related to Gauss curvature by $c = \int_{\partial K} \kappa_x \mathrm{d}A_x$.

generic distribution that is apt for a generic study of local complexity of a region of the curve but not to compute the whole complexity of the object.

We need a measure able to compute the shape complexity of an object independently of the discretisation. As we have seen in Chapter 2, continuous mutual information fulfills this requirement. Remember that the basic idea of this approach is that a distribution of global lines crossing an object defines a continuous information channel $X \to Y$, where X and Y represent the set of surface points (or contour points for 2D shapes). Mutual information I^c between two continuous random variables X and Y of ∂K is given by

$$I^c = \int_{\partial K} \int_{\partial K} \frac{F(x, y)}{\mu(\partial K)} \log(\mu(\partial K) F(x, y)) \mathrm{d}A_x \mathrm{d}A_y, \tag{3.2}$$

where ∂K represents the surface (3D) or the contour (2D) and $\mu(\partial K)$ its measure (i.e., A_T for 3D objects and L_T for 2D objects). Thus, for 3D-objects, the next descriptor can be defined.

Definition 3.1. The inner 3D-shape complexity is given by

$$I^c = \int_{\mathcal{S}} \int_{\mathcal{S}} \frac{F(x, y)}{A_T} \log(A_T F(x, y)) \mathrm{d}A_x \mathrm{d}A_y, \tag{3.3}$$

where $\mathcal{S}$ represents the internal surface of the object.

As we have seen in Sec. 2.3.2, the continuous mutual information can be efficiently computed with Monte Carlo integration by sampling global lines because global lines crossing $\mathrm{d}A_x$ and $\mathrm{d}A_y$ are distributed according to $F(x, y)\mathrm{d}A_x\mathrm{d}A_y$. Thus, sampling the density $f(x, y) = F(x, y)/A_T$ in Equ. 3.3 with global lines (see Equ. 2.26), we obtain

$$I^c \approx \frac{1}{N} \sum_{k=1}^{N} \log\left(\frac{A_T \cos\theta_{x_k} \cos\theta_{y_k}}{\pi r^2_{x_k y_k}}\right), \tag{3.4}$$

where N is the total number of segments of the global lines or the number of pairs of points considered, which is the total number of intersections divided by two (see Fig. 3.1). The term of the summatory is the contribution of each chord to the complexity, and we call it chord complexity.

Continuous mutual information is invariant to translations, rotations and a change of scale. As we have seen in Sec. 3.1, point-to-point form factor gives the density of uniformly distributed lines crossing differential areas with centre at these points, and by definition, this density is invariant under translations and rotations. In addition, scale invariance is easily seen from Equ. 3.4 where a scaling of the distances is compensated by the corresponding scaling of the total area.

Observe that chord complexity is bigger for small chord lengths and for angles near to zero. Thus, regions corresponding to corners or narrow spaces will contribute more to mutual information (shape complexity). As we have previously seen (Sec. 2.3), for the interior of an empty sphere, the result obtained is $I^c = 0$ since any pair (x, y) fulfills $F(x, y) = 1/A_T$. Analogously to [83, 143, 85], from chord complexities, we could obtain a shape complexity distribution of the object to be applied in object recognition, classification, clustering, and retrieval.

Table 3.1: Mutual information I^c and dual-mutual information I^{c*} for the objects in Fig. 3.2 (10^5 global lines has been used). The dual object of a sphere is the empty set and its I^{c*} is not defined.

Figure	Object	I^c	I^{c*}
	Sphere	0	
Fig. 3.2.*a*	Tetrahedron	2.626	1.800
Fig. 3.2.*b*	Hexahedron	1.610	3.650
Fig. 3.2.*c*	Octahedron	1.262	3.165
Fig. 3.2.*d*	Dodecahedron	0.813	6.161
Fig. 3.2.*e*	Icosahedron	0.545	5.412
Fig. 3.2.*f*	In-Hexahedron	7.438	2.952
Fig. 3.2.*g*	Out-Hexahedron	0.571	5.599
Fig. 3.2.*h*	Out-Icosahedron	5.914	3.198
Fig. 3.2.*i*	Torus	2.859	4.154
Fig. 3.2.*j*	Cylinder-I	1.027	5.337
Fig. 3.2.*k*	Cone-I	1.648	3.164
Fig. 3.2.*l*	Pencil-I	2.104	0.730
Fig. 3.2.*m*	Plate-I	5.601	1.523
Fig. 3.2.*n*	Glass-I	11.392	3.397
Fig. 3.2.*o*	Cylinder-II	1.125	3.508
Fig. 3.2.*p*	Cone-II	1.475	1.985
Fig. 3.2.*q*	Pencil-II	2.908	0.377
Fig. 3.2.*r*	Plate-II	6.482	1.222
Fig. 3.2.*s*	Glass-II	11.344	2.518

3.2.2 INNER 3D-SHAPE COMPLEXITY RESULTS

In this section, we show the inner shape complexity (column I^c of Table 3.1) of a set of 3D objects (Fig. 3.2). As we have seen, the sphere has the minimum complexity ($I^c = 0$), and among the platonic solids, the minimum and maximum complexity correspond, respectively, to icosahedron and tetrahedron since at the corners of a tetrahedron the dependence between the parts is greater than at the corners of an icosahedron. As expected, the polyhedra that are nearer to the sphere are less complex, that is, they have less correlation. In fact, it can be considered that mutual information I^c quantifies how "distant" is any object from the sphere. Observe also that the complexity of in-stellated hexahedron is bigger than the one of the out-stellated icosahedron because the hexahedron has folded faces that leave very narrow spaces between them.

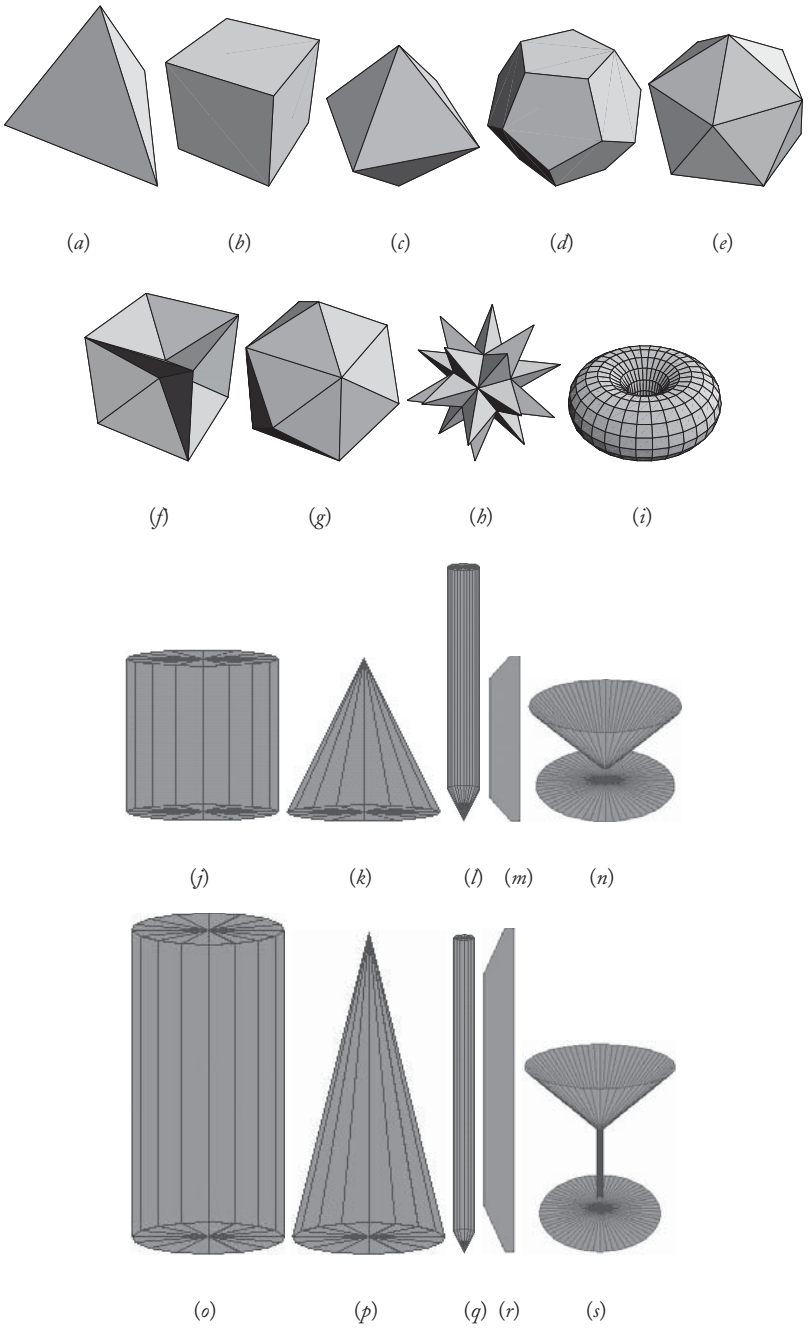


Figure 3.2: Collection of test objects of Table 3.1: platonic solids and other geometrical shapes. ([104] © IEEE, 2005.)

Table 3.2: Exact I^c values for a circle and three regular polygons compared with results obtained by Monte Carlo simulation (MC, 10^5 global lines have been used).

Scene	Exact value			MC
Circle	$\log \frac{\pi}{e}$	$\simeq$	0.209	0.209
Hexagon	$\log \frac{e^{\sqrt{3}-4} 324(7+4\sqrt{3})}{168+97\sqrt{3}}$	$\simeq$	0.475	0.475
Tetragon	$\log \frac{8(1+\sqrt{2})}{e^{1+\sqrt{2}}}$	$\simeq$	0.789	0.788
Trigon	$\log \frac{18}{e^2}$	$\simeq$	1.285	1.284

On the other hand, in Fig. 3.2, we have a collection of common objects with some modifications. Similar shapes should have similar mutual information values (cylinders, cones, pencils, etc.). Observe that mutual information ranges from a minimal value for the Cylinder-I (Fig. 3.2.*j*) to a maximum value for the Glass-I (Fig. 3.2.*n*). Note that the Plate objects (Fig. 3.2.*m* and Fig. 3.2.*r*) are very simple but have high mutual information values. These values are explained from the fact that two very near surfaces contribute with high values to the computation of its mutual information.

Note also that Plate-II has a higher mutual information value that Plate-I because it is a proportionally thinner object. Note that the same kind of behaviour can be observed for both Pencil objects (Fig. 3.2.*l* and Fig. 3.2.*q*).

3.2.3 INNER 2D-SHAPE COMPLEXITY RESULTS

Similarly, to Equ. 3.3, 2D-shape complexity can be defined as follows:

Definition 3.2. The inner 2D-shape complexity is given by

$$I^c = \int_{\mathcal{L}} \int_{\mathcal{L}} \frac{F(x,y)}{L_T} \log(L_T F(x,y)) \mathrm{d}L_x \mathrm{d}L_y, \tag{3.5}$$

where $\mathcal{L}$ is the set of segments that form the environment, L_T is the total length of the contour, and $F(x,y) = \cos\theta_x \cos\theta_y / (2r_{xy}) V(x,y)$ is the point-to-point 2D-form factor between x and y. As in Equ. 3.4, this integral can be solved by Monte Carlo integration and the computation can be done efficiently by casting uniformly distributed global lines upon segments [15]. Hence, continuous mutual information can be approximated by

$$I^c \approx \frac{1}{N} \sum_{k=1}^{N} \log\left(\frac{L_T \cos\theta_{x_k} \cos\theta_{y_k}}{2r_{x_k y_k}}\right), \tag{3.6}$$

where N is the total number of pairs of points considered, which is the total number of intersections divided by two.

The closed-form solution of the continuous mutual information integral for the circle and some regular polygons (hexagon, square, and equilateral triangle) is shown in Table 3.2. The circle requires special attention. Observe that its complexity is different from zero: $I^c = \log(\pi/e)$. Since a sphere has zero complexity ($I^c = 0$), we could expect the same for a circle. But null complexity for the sphere can be explained by the fact that a global line can be generated by selecting two random points on its surface. However, in the case of a circle, selecting pairs of random points on its perimeter will not yield a uniform density [15]. In 2D, we can not imagine a scene with less complexity than a circle. In this sense, there is a significant difference between the 3D and 2D worlds.

We have also computed the complexity of a sequence representing the formation of a 12-pointed star and the von Koch fractal. If we start with a polygon of 24 edges, with a complexity very similar to the one of a circle, and we continue closing the edges as shown in Fig. 3.3.*a*, the complexity increases noticeably, due to the growth of the interaction within the edges. In the von Koch fractal (Fig. 3.3.*b*), a similar thing happens: by increasing the number of corners, the correlation increases.

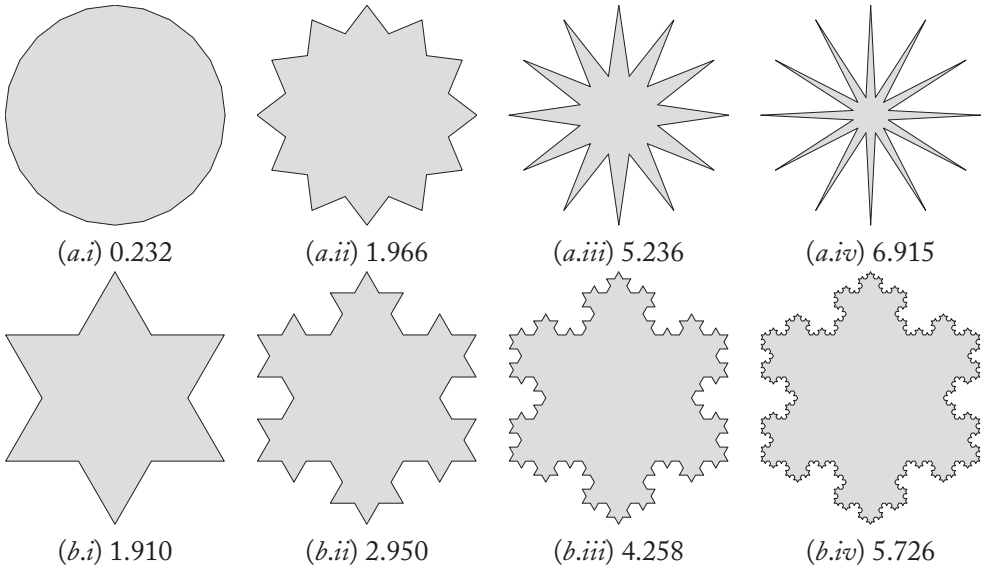


Figure 3.3: (a) I^c value for a 24-sided regular polygon and three 12-pointed stars and (b) for von Koch fractals. ([104] © IEEE, 2005.)

3.3 OUTER SHAPE COMPLEXITY

In addition to the mutual information of an object, we introduce a secondary shape complexity measure given by the mutual information between the object and its minimum circumscribing sphere (Fig. 3.4). The mutual information of this "new" object (dual-object) is called dual mutual information (I^{c^*}). This value can be seen as the increase in mutual information induced by the

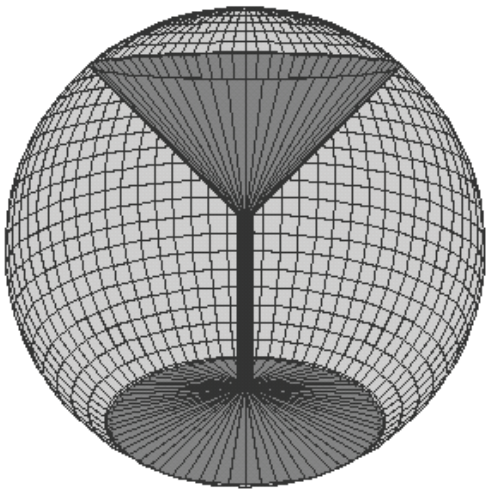


Figure 3.4: The dual-object is formed by the space between the object and its circumscribing sphere. ([104] © IEEE, 2005.)

introduction of the object within an spherical environment. The choice of this environment is coherent with the fact that the mutual information of a sphere is zero.

The sphere of the smallest radius that contains an object exists and is unique. Following Gärtner [42], we compute this in linear time with respect to the number of vertices of the object. The dual mutual information, similarly to mutual information, is also computed from a set of global lines. Note that the same set of global lines can be used to compute both shape complexities. The chords that contribute to the dual mutual information are complementary of the ones that are used for mutual information computation. Note that the sphere-sphere chords do not contribute to the outer shape complexity since the term within the logarithm in Equ. 3.4 is equal to 1. If the object is a sphere, a singularity is obtained.

From Table 3.1, we can analyse the behaviour of our secondary descriptor I^{c^*}. We can see that the bigger the dual mutual information the bigger the interaction between the object and the circumscribing sphere. For instance, we observe that, somehow against intuition, hexahedron and dodecahedron interchange their ordering with octahedron and icosahedron when considering I^{c^*}. This is so because the relative volume within the circumscribing sphere is higher for the hexahedron and dodecahedron than for the octahedron and icosahedron, respectively, and consequently, the interaction within the dual object becomes stronger. Also, for two similar objects, the more complex the contour of an object (concavities, rugosities), the higher the I^{c^*} value.

It can be seen that the dual mutual information clearly discriminates between the Cone-I ($I^{c^*} = 3.164$) and the Cone-II ($I^{c^*} = 1.985$). The same happens for Cylinder-I and II with I^{c^*}

values of 5.337 and 3.508, respectively. Maybe, the most remarkable example are the objects Glass-I and Glass-II (Fig. 3.2.*n* and Fig. 3.2.*s*, respectively), since with very similar I^c value, I^{c^*} clearly separates both objects.

CHAPTER 4

Refinement Criteria for Ray-Tracing

In ray-tracing, we trace rays from the camera or eye through the pixels in the image plane, and we obtain the color of the hit point of the scene in a recursive way where a new ray is cast from the hit point and so on. This allows us to obtain the global illumination in a scene. Although ray-tracing is a straightforward and powerful image synthesis technique [43, 44, 123], it usually requires many rays per pixel to eliminate the aliasing or noise in the final image. However, not all the pixels in the image require the same number of rays. The edge of an object, the contour of a shadow, and a high illumination gradient will require a much better treatment than a region with almost uniform illumination. To this effect, many pixel supersampling, refinement criteria have been defined in the literature (a good overview is [44]). The refinement criteria are mainly based on the homogeneity encountered in the samples. Heterogeneity should lead to further sampling, possibly with an adaptive subdivision of the domain. Oracles are then built based on these criteria.

The data obtained by sampling the scene through the pixel can be used to calculate a pixel homogeneity measure from two different points of view: radiance (color) and visibility (geometry). Shannon entropy (Sec. 1.1) will be interpreted as a measure of the degree of homogeneity of a pixel in the sense that the more heterogeneous the pixel, the more difficult it is to obtain an accurate value [101, 102]. We associate homogeneity with quality, so that the need for pixel refinement is proportional to the lack of quality (i.e., heterogeneity of the samples). The idea behind this scheme is to obtain sufficient information in the refinement algorithm in order to find out the sampling needs. We also introduce refinement criteria based on *f-divergences* [103] (Sec. 1.8). The introduction of these measures is motivated by the observation that in the radiosity setting the mutual information-based oracle can be rewritten as an *f*-divergence (Sec. 2.5).

4.1 BACKGROUND

Three principal subproblems make up the process of obtaining a good quality image in ray tracing: efficient sample generation, adaptive control of the sampling rate, and filtering for image reconstruction [87]. Many approaches deal with them:

1. Different pixel sampling methods have been introduced, among them: jittered sampling [24, 30], Poisson disk sampling [30, 79, 77], hierarchical sampling [61], complete stratification at each refinement level [112], importance sampling [116], and quasi-Monte Carlo sampling [64, 84].

2. Diverse refinement criteria for adaptive sampling, based on color intensities and/or scene geometry, can be found to control the sampling rate: Dippé and Wold [30] present an error estimator based on the RMS signal to noise ratio and also consider its variance as a function of the number of samples; Mitchell [79] proposes a contrast [11] based on the characteristics of the human eye; Lee et al. [72], Purgathofer [96], and Tamstorf and Jensen [125] develop different methods based on the variance of the samples with their respective confidence intervals.

3. Samples are filtered to produce the final pixel values. Different filter shapes have been used in image reconstruction: box filter, triangular filter, Gaussian filter, multi-stage filter, etc. (see [44]).

For the purpose of this chapter, we review three commonly used refinement criteria: contrast, depth difference, and variance of the samples.

Mitchell [79] uses a contrast measure [11] for each RGB channel defined by

$$C = \frac{I_{\max} - I_{\min}}{I_{\max} + I_{\min}}, \tag{4.1}$$

where $I_{\min}$ and $I_{\max}$ are, respectively, the minimum and maximum light intensities of the channel. Supersampling is done if any contrast is higher than a given threshold. Mitchell proposes RGB threshold values (0.4, 0.3, and 0.6, respectively) based on the relative sensitivity of the visual system.

In [118], within an interactive rendering context, Simmons uses a priority value p_c based on the above concepts (contrast and perception) [79, 44] defined by

$$p_c = 0.4\frac{r_{\max} - r_{\min}}{r_{\max} + r_{\min}}\overline{r} + 0.3\frac{g_{\max} - g_{\min}}{g_{\max} + g_{\min}}\overline{g} + 0.6\frac{b_{\max} - b_{\min}}{b_{\max} + b_{\min}}\overline{b}, \tag{4.2}$$

where max, min, and the overline represent, respectively, the maximum, minimum, and average values for r, g, and b color channels.

On the other hand, a useful and simple geometric measure for refinement is the depth difference, used in image based rendering [28, 29, 95] and interactive rendering [118]. Depth difference is given by

$$p_d = 1 - \frac{d_{\min}}{d_{\max}}, \tag{4.3}$$

where $d_{\max}$ and $d_{\min}$ represent maximum and minimum distance (i.e., the bigger the gap between the nearest and farthest object, the more priority is required in the refinement in order to capture more details inside this scene). In [118], p_c and p_d measures are combined in

$$p_v = \delta p_c + (1 - \delta) p_d, \tag{4.4}$$

with a weight $\delta = 0.9$.

The basic idea of variance-based methods [72, 96, 125] is to continue sampling until the confidence level or probability that the true value L is within a given tolerance t of the estimated value $\widehat{L}$ is $1 - \alpha$:

$$\Pr[L \in (\widehat{L} - t, \widehat{L} + t)] = 1 - \alpha. \tag{4.5}$$

Mitchell considers that variance is a poor measure of visual perception of local variation [79]. Kirk and Arvo showed that these methods are biased and proposed a simple correction scheme [66].

Refinement criteria have also been applied in the image-based rendering field to weight pixel color for reconstruction purposes [95] and adaptive sampling strategies [28, 29]. Also, Bolin and Meyer [7] have developed a perceptually-based approach using statistical and vision models.

4.2 PIXEL QUALITY

In this section, we introduce a pixel quality measure, the pixel entropy. This measure will be defined from the information provided by a set of samples on the image plane. We use the following sets:

- $\mathcal{P}$ is the set of $N_p > 0$ pixels of the image plane.
- S_p is the set of $N_s^p > 1$ samples of pixel p.
- $S_\mathcal{P}$ is the set of $N_s^\mathcal{P} = \sum_{p \in \mathcal{P}} N_s^p$ samples of the image plane.

The implementation of a sample consists in casting a ray $r(v, \Theta)$ from a scene viewpoint v with direction Θ through the image plane and, in particular, through a pixel. Let us consider that each sample $s \in S_\mathcal{P}$ that hits a scene surface gives us information about the color, distance and orientation of the hit point with respect to the viewpoint.

The entropy (Equ. 1.2) measures the expectation of the surprise of a probability distribution and it can be considered also to be a measure of its homogeneity (Sec. 1.5). From the sample set and from the entropy, two different quality measures are defined, pixel color entropy and pixel geometry entropy, based on the color and geometry, respectively.

4.2.1 PIXEL COLOR ENTROPY

Our first objective is to define the pixel color entropy. We start with a definition of entropy concerning all the samples passing through the image plane. We consider that the color belongs to a color system $\mathbf{c}$ structured in components called color channels. Without loss of generality, in the majority of cases the color measures will refer to a single channel $c \in \mathbf{c}$, $c(s)$ being the color channel data of a sample $s \in S$ (e.g., radiance, luminance, and RGB values).

Let us consider the probability of each image plane sample as its color channel contribution relative to the whole of the image plane sample set (i.e., the ratio between the color channel value of each sample and the sum of the color channel values sampled in the plane).

Definition 4.1. Given a channel c, the image plane channel entropy is defined by

$$H^c(\mathcal{P}) = -\sum_{i=1}^{N_s^\mathcal{P}} r_i \log r_i \qquad r_i = \frac{c(s_i)}{\sum_{j=1}^{N_s^\mathcal{P}} c(s_j)}, \tag{4.6}$$

where r_i represents the normalized color of sample $s_i \in S_\mathcal{P}$. This measure can be interpreted as the

color channel homogeneity of the samples passing through the image plane (Sec. 1.5). Analogously, at the pixel level, we consider the probability of each pixel sample as its color channel contribution relative to the whole of the pixel sample set.

Definition 4.2. Given a pixel p and a channel c, the pixel channel entropy is defined by

$$H^c(\mathrm{p}) = -\sum_{i=1}^{N_\mathrm{s}^\mathrm{p}} p_i \log p_i \qquad p_i = \frac{c(s_i)}{\sum_{j=1}^{N_\mathrm{s}^\mathrm{p}} c(s_j)}, \tag{4.7}$$

where p_i represents the normalized color of sample $s_i \in S_\mathrm{p}$.

From the properties of the entropy (Sec. 1.1), the image plane channel entropy ranges from 0 to $\log N_\mathrm{s}^\mathcal{P}$ and the pixel entropy from 0 to $\log N_\mathrm{s}^\mathrm{p}$. The maximum values are obtained when the channel color of all the samples is the same (i.e., we have a uniform probability distribution). Using the grouping property of entropy (see Sec. 1.1), it is easy to see that the image plane and pixel channel entropies are related in the following way:

$$H^c(\mathcal{P}) = \sum_{i=1}^{N_\mathrm{p}} q_i H^c(\mathrm{p}_i) - \sum_{i=1}^{N_\mathrm{p}} q_i \log q_i = \sum_{i=1}^{N_\mathrm{p}} q_i H^c(\mathrm{p}_i) + H_\mathrm{I}^c(\mathcal{P}), \tag{4.8}$$

where $q_i = \sum_{j=1}^{N_\mathrm{s}^{\mathrm{p}i}} r_j$ is the importance (sum of probabilities) of pixel p_i, $H^c(\mathrm{p}_i)$ is the channel entropy of pixel p_i, and $H_\mathrm{I}^c(\mathcal{P}) = -\sum_{i=1}^{N_\mathrm{p}} q_i \log q_i$ is the entropy of the image plane calculated from the importance of each pixel. That is, the entropy of the image plane is the sum of all the pixel entropies, weighted by the importance of each pixel, plus the importance entropy obtained from the importance of each pixel. Observe that Equ. 4.8 can be considered as a recursive formula if each pixel is divided into subpixels and the same decomposition is applied in turn to the entropy $H^c(\mathrm{p}_i)$ of each pixel (see Fig. 4.1). Note also that $H^c(\mathcal{P}) - H_\mathrm{I}^c(\mathcal{P})$ is the entropy, or information, that will be captured in the refinement process.

The image plane and pixel entropies can be interpreted as the color homogeneity or uniformity measured by its sample set and thus can be considered measures of the quality of the color channel. We can also observe that the entropy increases with the number of samples. In order to give a normalized pixel quality measure, the pixel channel entropy is divided by $\log N_\mathrm{s}^\mathrm{p}$.

Definition 4.3. Given a channel c, the pixel channel quality is defined by

$$Q^c(\mathrm{p}) = \frac{H^c(\mathrm{p})}{\log N_\mathrm{s}^\mathrm{p}}. \tag{4.9}$$

If we want to consider the global quality of a pixel, we mix its set of channels.

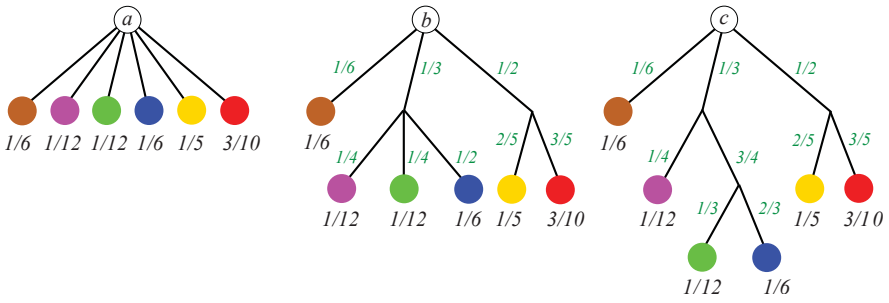


Figure 4.1: Recursiveness of the grouping property of entropy. The entropy of probability distributions of ⓐ is $H(\{\frac{1}{6}, \frac{1}{12}, \frac{1}{12}, \frac{1}{6}, \frac{1}{5}, \frac{3}{10}\})$, of ⓑ is $H(\{\frac{1}{6}, \frac{1}{3}, \frac{1}{2}\}) + \frac{1}{3}H(\{\frac{1}{4}, \frac{1}{4}, \frac{1}{2}\}) + \frac{1}{2}H(\{\frac{2}{5}, \frac{3}{5}\})$, and of ⓒ is $H(\{\frac{1}{6}, \frac{1}{3}, \frac{1}{2}\}) + \frac{1}{3}(H(\{\frac{1}{4}, \frac{3}{4}\}) + \frac{3}{4}H(\{\frac{1}{3}, \frac{2}{3}\})) + \frac{1}{2}H(\{\frac{2}{5}, \frac{3}{5}\})$. Observe that in all these cases the entropy is equal to 2.445. ([102] © Canadian Information Processing Society, 2003.)

Definition 4.4. Given a color system **c**, the pixel color quality is defined by the weighting of its pixel channel qualities:

$$\mathbf{Q}^{\mathbf{c}}(\mathrm{p}) = \frac{\sum_{c \in \mathbf{c}} w^c Q^c(\mathrm{p})}{\sum_{c \in \mathbf{c}} w^c}, \tag{4.10}$$

where w^c is the weight of channel c. The weights depend on each color system. Without a priori information, the same weight per channel can be considered; otherwise, we can use a weight based on the relative luminance of a color[1]. This measure will enable us to define a color contrast measure for pixel sampling (Sec. 4.3.1). Note that the larger the number of samples the more accurate the quality measure.

In Fig. 4.2.*b*, we present a color quality map of the scene shown in Fig. 4.2.*a* using an sRGB color system with the same weight per channel. In the thermal scale used, the minimum quality corresponds to the blue and the maximum to the red[2]. A low quality in shadow areas and edges can be observed.

4.2.2 PIXEL GEOMETRY ENTROPY

Similar concepts introduced in the above section can be defined with respect to a geometric measure. If h is the hitpoint of a sample ray $s = r(v, \Theta)$[3], the geometric information of each sample is given by $\theta_{n_h}^{-\Theta}$ (i.e., the angle of the normal at the hit point) and by the distance r_{vh} between this point

[1] For an sRGB system, $w^{\mathrm{R}} = 0.2126$, $w^{\mathrm{G}} = 0.7152$, and $w^{\mathrm{B}} = 0.0722$ [22].

[2] In order to observe more details in the color quality maps, the outliers (for instance, a sample corresponding to a very low probability, which when substituted in the denominator of the Monte Carlo estimator, Equ. 2.9, causes a very high luminance value) are clamped to the borders of the interval $[-k\sigma, k\sigma]$ where σ is the standard deviation of the results and k is a parameter that modulates the width of the interval.

[3] That is, a ray with origin at viewpoint v and direction Θ.

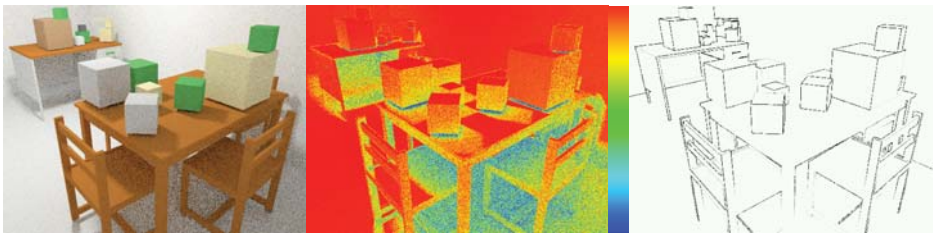
and the origin of the ray (i.e., ray length). We take

$$g(s) = \frac{\cos \theta_{n_h}^{-\Theta}}{r_{vh}^2}, \tag{4.11}$$

as the geometry factor of the sample. This value provides a quality measure of the visibility of a scene point from the observer's point of view.

Replacing the sample color values by geometry values (i.e., $c(\cdot)$ by $g(\cdot)$) in Equ. 4.6 and Equ. 4.7, we obtain the definitions of image plane geometry entropy $H^{\mathrm{g}}(\mathcal{P})$ and pixel geometry entropy $H^{\mathrm{g}}(\mathrm{p})$, respectively [102]. An identical relation to Equ. 4.8 can be established between the geometric entropies of the image plane and the pixel. Given a pixel p, the pixel geometry quality is given by the normalization of its pixel geometry entropy: $Q^{\mathrm{g}}(\mathrm{p}) = H^{\mathrm{g}}(\mathrm{p}) / \log N_s^{\mathrm{p}}$.

In Fig. 4.2.*c*, we show the geometry quality map of Fig. 4.2.*a* using a grey scale. The lowest entropy corresponds to the darkest part, the highest entropy to the lightest[4]. Observe that the edges have a very low entropy and are very clearly emphasised.



(*a*) Reference (*b*) $\mathbf{Q}^{\mathbf{c}}$ (*c*) Q^{g}

Figure 4.2: Color quality map (thermal scale) and geometry quality map (grey scale). (*a*) Reference image obtained with $N_s^{\mathrm{p}} = 8$. (*b*) Pixel color quality $\mathbf{Q}^{\mathbf{sRGB}}$ with the same weight per channel. (*c*) Pixel geometry quality Q^{g}. ([101] © Springer-Verlag, 2002.)

4.3 PIXEL CONTRAST

In this section, we present the pixel contrast measures based on pixel entropy (Sec. 4.2). As the entropy represents the homogeneity of the information brought back by the samples (rays crossing a pixel), we can define a simple measure which expresses the diversity or contrast of a pixel.

4.3.1 PIXEL COLOR CONTRAST

From the entropy $H^c(\mathrm{p})$, we define the pixel contrast as follows:

[4] The outliers have the same treatment as in the color quality maps.

Definition 4.5. Given a pixel p and a channel c, the pixel channel contrast is defined by

$$C^c(\mathrm{p}) = 1 - Q^c(\mathrm{p}) = 1 - \frac{H^c(\mathrm{p})}{\log N_\mathrm{s}^\mathrm{p}}. \tag{4.12}$$

It represents the color channel heterogeneity or contrast of a pixel and takes a value in $[0, 1]$. We can also introduce the pixel binary contrast from minimum and maximum color channel probabilities captured by this pixel. This measure is obtained from the binary entropy of these values (Sec. 1.1).

Definition 4.6. Given a pixel p and a channel c, the pixel channel binary contrast is defined by

$$C_b^c(\mathrm{p}) = 1 - H_b^c(\mathrm{p}) \qquad H_b^c(\mathrm{p}) = H\Big(\Big\{\frac{p_\mathrm{min}}{p_\mathrm{min} + p_\mathrm{max}}, \frac{p_\mathrm{max}}{p_\mathrm{min} + p_\mathrm{max}}\Big\}\Big), \tag{4.13}$$

where $H_b^c(\mathrm{p})$ is the binary entropy of the minimum and maximum channel color probabilities, p_min and p_max, respectively. Both measures, $H_\mathrm{b}^c(\mathrm{p})$ and $C_\mathrm{b}^c(\mathrm{p})$, range also between 0 and 1 due to the fact that, in this case, only two values are taken into account. This binary measure yields more radical contrast than $C^c(\mathrm{p})$.

Similarly, to previous works [79, 44, 118], we can obtain the color contrast of a pixel by averaging all the color channel contrasts weighted by their respective importances (color channel average). This avoids oversampling on the areas with low color values.

Definition 4.7. Given a pixel p and a color system $\mathbf{c}$, the pixel color contrast is defined by the weighting of its pixel channel contrasts:

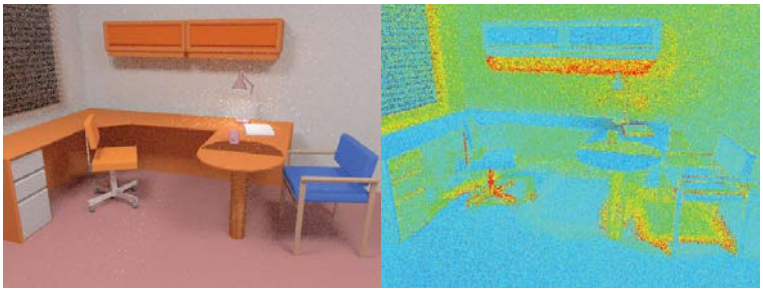
$$\mathbf{C^c}(\mathrm{p}) = \frac{\sum_{c\in\mathbf{c}} w^c \overline{c}\, C^c(\mathrm{p})}{\sum_{c\in\mathbf{c}} w^c \overline{c}} \qquad \overline{c} = \frac{1}{N_\mathrm{s}^\mathrm{p}} \sum_{i=1}^{N_\mathrm{s}^\mathrm{p}} c(s_i), \tag{4.14}$$

where the channel contrasts are weighted by perceptual coefficients w^c, and $\overline{c}$ is the color average in channel c of all $s \in S_\mathrm{p}$. We can define in a similar way the pixel color binary contrast:

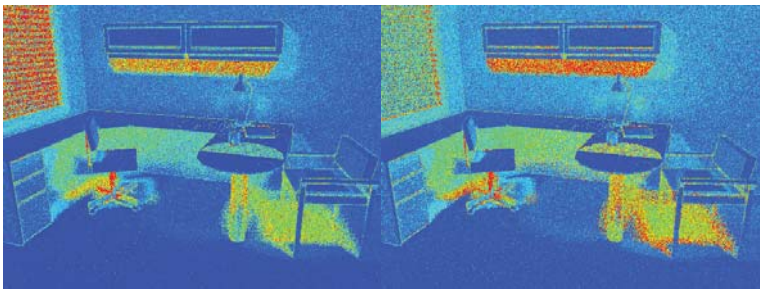
Definition 4.8. Given a pixel p and a color system $\mathbf{c}$, the pixel color binary contrast is defined by

$$\mathbf{C_b^c}(\mathrm{p}) = \frac{\sum_{c\in\mathbf{c}} w^c \overline{c}\, C_b^c(\mathrm{p})}{\sum_{c\in\mathbf{c}} w^c \overline{c}}. \tag{4.15}$$

In an sRGB system, the color contrast measures ($\mathbf{C^{sRGB}}$ and $\mathbf{C_b^{sRGB}}$) have three channels with coefficients w^R, w^G, and w^B (see Equ. 4.10). In Fig. 4.3, we show different color contrast maps to compare the heuristic measure p_c (Equ. 4.2, Fig. 4.3.*b*), with measures $\mathbf{C^c}$ (Fig. 4.3.*c*), and $\mathbf{C_b^c}$ (Fig. 4.3.*d*). We can observe how these last measures present a very good behaviour in critical areas (represented by warm colors) such as object edges and shadow contours. Note that $\mathbf{C^c}$ and $\mathbf{C_b^c}$ are more discriminating, especially the binary contrast.



(*a*) Reference (*b*) p_c



(*c*) $\mathbf{C^c}$ (*d*) $\mathbf{C^c_b}$

Figure 4.3: Color contrast maps. (*a*) Reference image obtained with $N_s^p = 8$. (*b*) Pixel color contrast p_c (Equ. 4.2). (*c*) Pixel color contrast $\mathbf{C^c}$. (*d*) Pixel color binary contrast $\mathbf{C^c_b}$.
Credit: Modelled by Gregory J. Ward, Albany (CA), USA. ([101] © Springer-Verlag, 2002.)

4.3.2 PIXEL GEOMETRY CONTRAST

As we have seen in Sec. 4.2.2, H^g represents the degree of geometric homogeneity of a pixel. From this measure, we define the pixel geometry contrast.

Definition 4.9. Given a pixel p, the pixel geometry contrast is defined by

$$C^g(\mathrm{p}) = 1 - Q^g(\mathrm{p}) = 1 - \frac{H^g(\mathrm{p})}{\log N_s^p}. \tag{4.16}$$

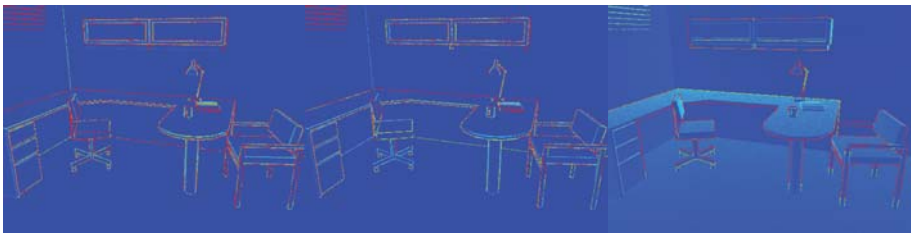
Similarly, to the previous section, the pixel binary contrast from the minimum and maximum geometry factor probabilities of this pixel can be introduced:

Definition 4.10. Given a pixel p, the pixel geometry binary contrast is defined by

$$C_b^g(\mathrm{p}) = 1 - H_b^g(\mathrm{p}) \qquad H_b^g(\mathrm{p}) = H\left(\left\{\frac{p_{\min}}{p_{\min} + p_{\max}}, \frac{p_{\max}}{p_{\min} + p_{\max}}\right\}\right), \tag{4.17}$$

where $H_b^g(\mathrm{p})$ is the binary entropy of the minimum and maximum geometry factor probabilities, $p_{\min}$ and $p_{\max}$, respectively.

The geometry contrast measures are shown in Figs. 4.4.*a*-*b* where the maps can be compared with Fig. 4.4.*c*, created using the depth difference heuristic p_d (Equ. 4.3). It can be seen that C^{g} and $C_{\mathrm{b}}^{\mathrm{g}}$ capture the majority of edges because we take into account two components: distance and orientation. These geometry contrast maps have been generated by using the same thermal scale (see Fig. 4.2) of the color contrast maps in order to be able to visually compare them. It can be clearly seen that color maps show the heterogeneity of regions while geometric maps identify edges.



(*a*) C^{g} (*b*) $C_{\mathrm{b}}^{\mathrm{g}}$ (*c*) p_d

Figure 4.4: Geometry contrast maps from Fig. 4.3.*a* obtained with $N_{\mathrm{s}}^{\mathrm{p}} = 8$. (*a*) Pixel geometry contrast C^{g}. (*b*) Pixel geometry binary contrast $C_{\mathrm{b}}^{\mathrm{g}}$. (*c*) Pixel depth difference p_d.
Credit: Modelled by Gregory J. Ward, Albany (CA), USA. ([101] © Springer-Verlag, 2002.)

4.3.3 PIXEL COLOR-GEOMETRY CONTRAST

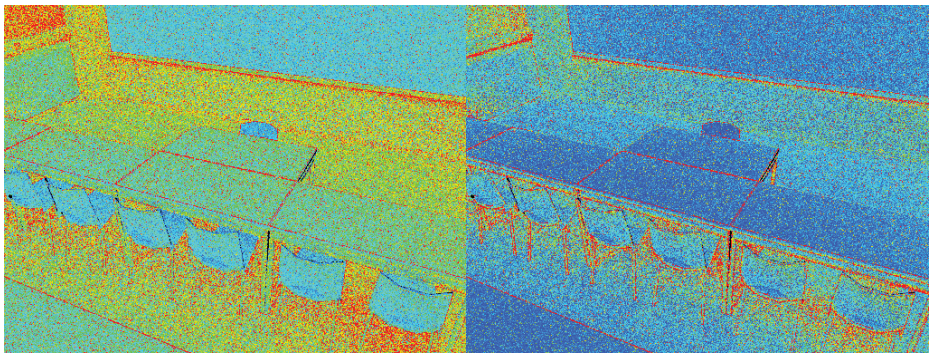
Finally, a combination of color and geometry contrasts is considered. This combination enables us to graduate, with a coefficient δ between 0 and 1, the influence of both measures.

Definition 4.11. Given a color system **c**, the pixel contrast is given by

$$C^{\mathbf{c}}(\mathrm{p}) = \delta \mathbf{C}^{\mathbf{c}}(\mathrm{p}) + (1-\delta) C^{g}(\mathrm{p}). \tag{4.18}$$

This combination can be made with any type of pixel color contrast and geometry contrast.

We show **in** Fig. 4.5 two different linear combinations. On the one hand, in Fig. 4.5.*a* we use the priority-value combination p_v (Equ. 4.4), made up also of color, p_c (Equ. 4.2), and geometry, p_d (Equ. 4.3). And, on the other hand, in Fig. 4.5.*b*, we combine $\mathbf{C}_{\mathrm{b}}^{\mathbf{c}}$ (Equ. 4.15) and C^{g} (Equ. 4.16). The same values $N_{\mathrm{s}}^{\mathrm{p}} = 4$ and $\delta = 0.9$ are used in both cases. A significant difference is observed: the combination in Fig. 4.5.*b* tends to obtain more radical contrasts as opposed to the other option which takes more homogeneous values. The explanation lies in the behaviour of the binary color contrast, which works exclusively with the extreme values of the data.



(a) p_v (b) $\mathcal{C}^{\mathbf{c}}$

Figure 4.5: Pixel contrast obtained with $N_s^p = 4$ and a linear combination with $\delta = 0.9$. (a) Pixel priority-value p_v (Equ. 4.4). (b) Pixel contrast $\mathcal{C}^{\mathbf{c}}$ using $\mathbf{C}_b^{\mathbf{c}}$ and C^g.
Credit: Model included in RenderPark [23], Computer Graphics Research Group, Department of Computer Science, Katholieke Universiteit Leuven, Leuven, Belgium. ([101] © Springer-Verlag, 2002.)

4.4 ENTROPY-BASED SUPERSAMPLING

In this section, we apply the previously defined contrast measures to supersampling in a stochastic ray-tracing implementation.

4.4.1 ALGORITHM

In the algorithm proposed, a pixel is first sampled at a relatively low density. From this set of samples, a refinement criterion is used to decide whether more sampling is required or not. Finally, all samples are used to obtain the final pixel color values. We can consider two kinds of adaptive sampling: first, when the refinement criterion plays the role of an oracle which decides the place and the quantity of supersampling necessary in one evaluation of the initial sampling only and, second, when the refinement criterion constantly evaluates the information received because of the supersampling and acts in consequence until it becomes satisfied. Here, we use the term supersampling exclusively for the first case and adaptive sampling for the second (Sec. 4.5).

We implement a simple supersampling technique: the sample set S will be proportionally distributed over the image plane according to the contrast $\mathcal{C}^{\mathbf{c}}$ estimated in each $\mathrm{p} \in \mathcal{P}$. This is equivalent to the use of the pixel contrast as an oracle. From the definition of the measure itself (Sec. 4.3), the cost in samples is controlled by the diversity of color and geometry in the pixel (i.e., low quality). The measure adapts the densities of sampling to the necessity of improvement in pixel quality. The generic procedure is made up of three sequentially different phases:

Oracle A pixel contrast measure is selected as oracle and a first estimate of actual contrast per pixel is obtained using an initial stratified sampling against the image plane. Usual values are 2, 4, and 8 (n_s^p). If we consider that the total number of samples $N_s^{\mathcal{P}}$ is prefixed, the final value of

the average of samples per pixel is $N_{\bar{s}}^{p} = N_s^{\mathcal{P}}/N_p$. The result of this phase is the answer from the oracle: the contrast map.

Sampling The unused samples in the calculation of the contrast, $N_p(N_{\bar{s}}^{p} - n_s^{p})$, are proportionally distributed to the values of contrast obtained per pixel. In each one of them, the sampling is also carried out with stratification.

Reconstruction The color information gathered in the previous phase is put together with the one obtained in the initial phase. It only remains for the signal to be recovered and to carry out the resampling process for each one of the pixels $p \in \mathcal{P}$ with any of the reconstruction methods applicable to the sampling system used. The result of this phase is the solution for the image plane thanks to the assignment of the final color to all of its pixels.

4.4.2 RESULTS

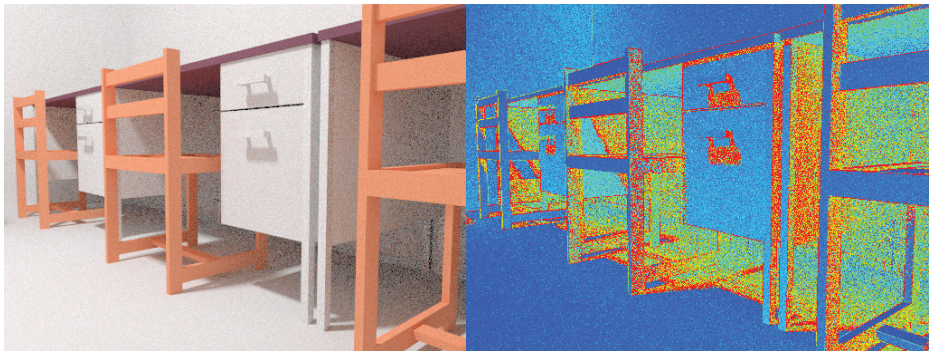
We show an example of the contrast measure $\mathcal{C}^{\mathbf{c}}$ (Equ. 4.18) used as a supersampling oracle in path-tracing [123]. In Fig. 4.6.*a.i*, we show a supersampling image obtained with $N_{\bar{s}}^{p} = 32$ in the following way. First, a uniform stratified sampling with $n_s^{p} = 8$ has been carried out in order to obtain the contrast map in Fig. 4.6.*a.ii*. Second, this map has been used in the supersampling process with an average of 24 rays per pixel. And third, in order to analyse the behaviour of the contrast, the signal reconstruction in the last phase is carried out by a piece-wise continuous reconstruction [87] using a box filter. The final pixel value corresponds to its signal average. The contrast measure used is a color and geometry combination with $\delta = 0.5$ based on binary contrasts $\mathbf{C}_b^{\mathbf{c}}$ (Equ. 4.15) and C_b^{g} (Equ. 4.17). This means that the more heterogeneous the region, the more supersampled it is (warm colors), and the less heterogeneous, the more undersampled it is (cool colors, with a minimum of 8 rays per pixel). Two close-ups are compared from the supersampling image (Figs. 4.6.*b–c.i*) and a similar image obtained by uniform stratified sampling with $N_s^{p} = 32$ (Figs. 4.6.*b–c.ii*). We can observe a decrease in noise in the supersampled regions, and a better representation of shadow contours and edges.

4.5 ENTROPY-BASED ADAPTIVE SAMPLING

In Equ. 4.8, we have seen that image plane and pixel channel entropies are related, thanks to the grouping property. It is important to note that this kind of decomposition can be applied recursively if the pixels are recursively subdivided. We will show, in this section, that this recursive decomposition provides us with a natural method of dealing with an adaptive sampling technique. This scheme, valid for any pixel sampling and ray-tracing method, is applied to stochastic ray-tracing [123] and compared with other techniques.

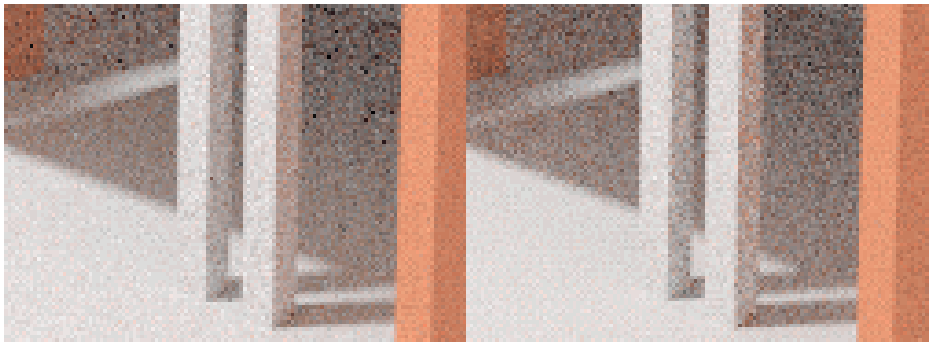
4.5.1 ADAPTIVE SAMPLING

We consider three phases in order to describe a generic process of adaptive sampling [44] (Fig. 4.7), for which the scheme used in the supersampling procedure of Sec. 4.4.1 is a particular case:



(*a.i*) Supersampling image (*a.ii*) Oracle $\mathcal{C}^{\mathbf{c}}$ for (*a.i*)

(*b.i*) Close-up from (*a.i*) (*b.ii*) Close-up from uniform sampling



(*c.i*) Close-up from (*a.i*) (*c.ii*) Close-up from uniform sampling

Figure 4.6: Entropy-based supersampling versus uniform sampling. (*a.i*) Supersampling with $N_{\mathrm{s}}^{\mathrm{p}} = 32$. (*a.ii*) Binary contrast map $\mathcal{C}^{\mathbf{c}}$ used as oracle to obtain (*a.i*). It has been calculated with $n_{\mathrm{s}}^{\mathrm{p}} = 8$, $\mathbf{C}_{\mathrm{b}}^{\mathbf{c}}$, $C_{\mathrm{b}}^{\mathrm{g}}$, and $\delta = 0.5$. Close-ups from (*a.i*) are shown in (*b-c.i*). They are compared with the same regions, (*b-c.ii*), respectively, taken from a uniform stratified sampling image with $N_{\mathrm{s}}^{\mathrm{p}} = 32$. ([101] © Springer-Verlag, 2002.)

Initial sampling An initial sampling pattern at a predetermined density is established.

Refinement tree The image space is divided into regions (e.g., pixels). For each of them, a refinement test selects a subset of samples for evaluation. If the result is negative, a new set of samples is generated at the points indicated by the new sampling geometry and the process goes back to the refinement test until the density of sampling in the region is accurate enough. The result of this process is a refinement tree of the image space where every node is a region with a density of sampling adapted to its own signal. In order to control extreme cases, we usually have other criteria to finish the recursion (e.g., minimum area of the regions and/or maximum depth of tree).

Reconstruction The information of the signal collected at every region is unified by a reconstruction process and, if necessary, sent to a filtering process. Finally, a resampling process (e.g., at the centre of pixel) determines the final values for each of the pixels of the image plane.

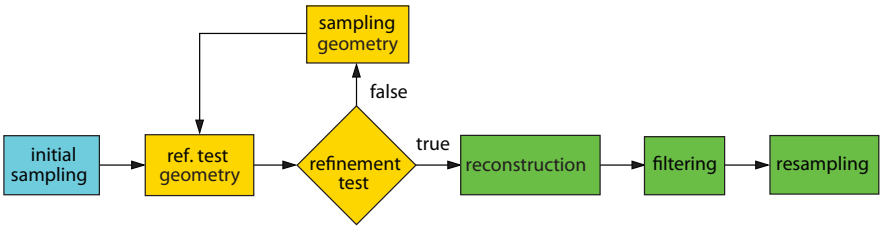


Figure 4.7: Adaptive sampling process with three phases: initial sampling (blue), refinement tree (gold), and image reconstruction (green).

Note that however much we increase the density of sampling locally, given that the signal is not usually band-limited, the sampling theory tell us that we can never capture it correctly [44]. Thus, fine details of edges, shadings, textures, and others will hardly have enough quality in the final image. The approximation done by the method consists in looking for the minimum set of samples which estimates locally the signal with enough accuracy. A critical problem appears in each of its phases and many approaches are found to deal with them [87, 44]: efficient sample generation, control of the sampling rate, and filtering. We focus our attention on obtaining an adaptive algorithm centered mainly on the refinement tree phase bringing a new perspective to the problem of controlling the sampling rate (refinement criterion).

The fundamental idea behind our refinement scheme is to capture the information in the refinement tree which results from the recursive decomposition of the entropy (Equ. 4.8). First and second terms on the right of Equ. 4.8 represent, respectively, the hidden information (pending to be discovered) and the information already acquired in the descent of the refinement tree. The following sampling algorithm will extract more information from the regions with more sample variation.

4.5.2 ALGORITHM

For the sake of simplicity, in the following analysis, we only consider the color information of one channel, although in the final algorithm, we will take the combination of color and geometry contrasts into account (Equ. 4.18).

A general description of the algorithm is as follows. On the image plane, we sample each pixel to capture the color of hitpoints and thus evaluate the information content (entropy) from the color probability distribution. If the information of a pixel is high enough (i.e., the rays provide us with sufficient color homogeneity on that pixel), refinement is not made, and the color reconstruction of this pixel is done. When the information is not high enough, this pixel is subdivided into regions, and we proceed in the same way for each region (subpixel). This recursive process defines a tree with two well-separated phases for a pixel:

- Pixel is refined until enough information is extracted (tree descent).
- Computation of the final color (tree ascent).

The descent in the refinement tree can be interpreted as a progressive gain in information. The information acquired at each level is added together so that, at the end of the refinement process, the total information from the tree is the sum of the information obtained over all the branches (Equ. 4.8). The measure used to decide the need of refinement will be the pixel contrast (Sec. 4.3).

Before introducing the algorithm, we will give the definitions of the data used in it. Concerning the tree data structure, n represents the tree level where

- $n = 0$ is the image level (root).
- $n = 1$ is the pixel level (composed of N_p pixels of the image).
- $n > 1$ is a subpixel level.

We consider an n-node as any node of the tree with $n > 0$ (i.e., not the root). The set of data is described in Table 4.1. To compute the final color of a pixel, we follow a path through the tree (Fig. 4.8). In the analysis below, we focus our attention on the tree-path k of length m going from pixel k_0 to subpixel k_{m-1}. In this path, p_n represents the probability of the tree-branch at level n and q_n the importance of the n-node. In our algorithm, this quantity appears naturally due to recursive decomposition of the entropy (see Equ. 4.8, Fig. 4.1, and Fig. 4.8). The value of importance is given by the probability of a node:

$$q_n = \begin{cases} 1, & \text{if } n = 0, \\ p_0 \cdots p_{n-1} = \frac{\overline{c}_{0,k_0}}{\sum_{i \in R_0} \overline{c}_{0,i}} \prod_{\ell=1}^{n-1} p_\ell, & \text{if } n > 0. \end{cases} \tag{4.19}$$

For our purposes, q_n does not need to be normalised, thus we omit the normalisation constant $\sum_{i \in R_0} \overline{c}_{0,i}$, and we take $q_n = \overline{c}_{0,k_0} \prod_{\ell=1}^{n-1} p_\ell$. Then, the computation of q_n can be simplified to the

fraction between the color average of the node and the number of regions to the power of $n-1$ (for more details, see [102]):

$$q_n \approx \frac{\overline{c}_n}{N_r^{n-1}}. \tag{4.20}$$

Table 4.1: Description of the data set of the refinement phase of entropy-based adaptive sampling in an image plane of N_p pixels. An n-node is a node of level $n > 0$ in the refinement-tree.

id	description
N_r	Number of regions in which an n-node can potentially be subdivided.
N_s^r	Number of samples of an n-node.
R_n	Set of regions of an n-node.
S_n	Set of samples of an n-node.
$S_{n,i}$	Set of samples of an n-node region $i \in R_n$.
k	Path-tree $k = (k_0, \ldots, k_{m-1})$ where k_n is the region chosen at level n.
c_n	Color data in an n-node.
$\overline{c}_n$	Average color channel data in an n-node.
$\overline{c}_{n,i}$	Average color in an n-node region $i \in R_n$.
$\widehat{c}_n$	Color estimation in an n-node leaf.
p_n	Probability of region k_n of an n-node in a path k.
q_n	Probability of an n-node in a path k.

Now, we can proceed to explain the algorithm. In the descent phase, we sample an n-node and compute the contrast $\mathcal{C}_n^\mathbf{c}$, from the color and geometry contrast combination (Equ. 4.18), using

$$\mathcal{C}_n^\mathbf{c} = \delta \mathbf{C}_n^\mathbf{c} + (1-\delta) C_n^\mathrm{g}, \tag{4.21}$$

where we made explicit the dependence upon the n-node and

$$\mathbf{C}_n^\mathbf{c} = \sum_{c \in \mathbf{c}} w^c q_n^c C_n^c, \tag{4.22}$$

where, with respect to the color contrast (Equ. 4.14), we have substituted the importance $\overline{c}$ by the importance q_n. Note that the Equ. 4.21 could also be calculated from the respective binary versions of color and geometry contrasts (Sec. 4.3).

In the algorithm, we subdivide the pixel or subpixel when the contrast of an n-node is not less than a given threshold ($\mathcal{C}_n^\mathbf{c} \geq \epsilon$). Thus, the phase of ascent begins when the test fails ($\mathcal{C}_n^\mathbf{c} < \epsilon$). This happens because either the contrast (which represents the color heterogeneity) or the importance ($q_n \to 0$ for growing n) are low. In this phase, each n-node leaf in the path provides its color estimation $\widehat{c}_n$ from the signal reconstructed and weighted using a filter. The final color of an n-node

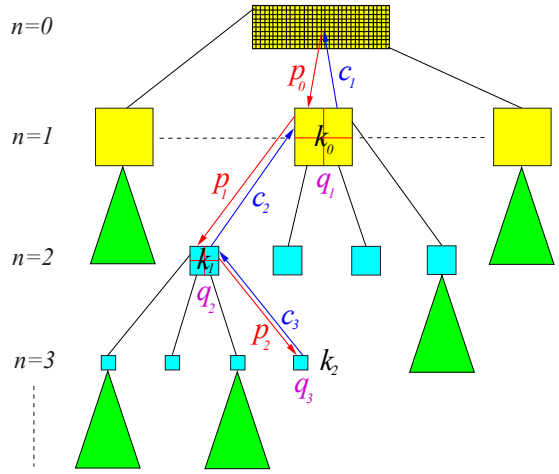


Figure 4.8: A refinement-tree-path $k = (k_0, k_1, k_2)$ of length $m = 3$ of entropy-based sampling. The number of regions of an n-node is $N_r = 4$. We show the computation of the k_0-pixel color: $c_{0,k_0} = c_1$ from the refinement (red) and reconstruction (blue) phases. The probabilities p_n and importances q_n are computed in the refinement phase to evaluate the pixel contrast (Equ. 4.18). ([102] © The Canadian Information Processing Society, 2003.)

is given by

$$c_n = \begin{cases} \widehat{c}_n, & \text{if } \mathcal{C}_n^{\mathbf{c}} < \epsilon \text{ (i.e., a leaf)}, \\ \sum_{i \in R_n} c_{n,i}, & \text{otherwise}, \end{cases} \tag{4.23}$$

where $c_{n,i}$ is the final color of i-region of the n-node. Finally, we get c_1 for the color of the pixels (or equivalently c_{0,k_0} in the path considered). An example of this process is shown in Fig. 4.8. Observe that importance sampling is naturally integrated in the algorithm. Following importance sampling criteria, a function should be sampled proportionally to its value, which is what we obtain with our adaptive descent.

4.5.3 IMPLEMENTATION

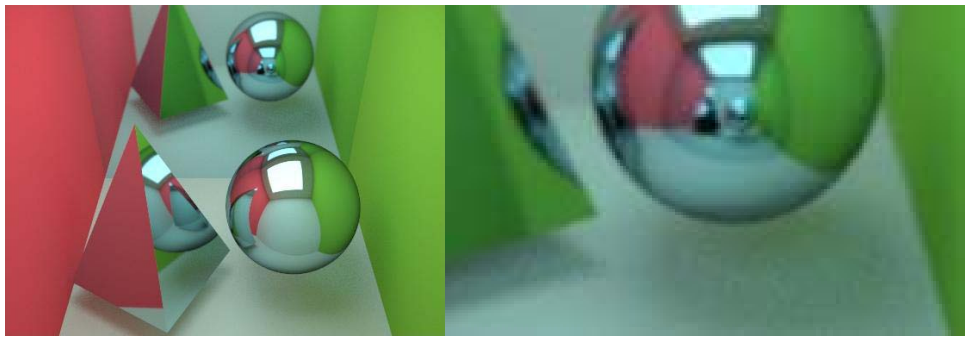
We have implemented the entropy-based adaptive algorithm using a path-tracing method. With respect to the three phases of an adaptive sampling scheme (Sec. 4.5.1), we should consider the following:

Sampling Generation The subdivision corresponds to a split into k equal subregions. Usually a binary-tree or quad-tree [141, 61, 87] is used. We use this last option ($k = 4$). For each region, we choose an adaptive stratified sampling [44]. In order to simplify the implementation, the stratification is adapted to the number of subregions and samples (one sample per region). This makes the re-use of the samples at every new level of subdivision easier.

Sampling Rate As refinement criteria, our scheme used the pixel contrast $\mathcal{C}^{\mathbf{c}}$ (Equ. 4.18) in any of its variants (e.g., binary contrast). The entropy-based refinement tree, built in Sec. 4.5.2, enables us to evaluate the information from the signal in order to adapt the density of sampling. The structure implements an importance sampling approach (contrast-based) directed towards the complex regions (heterogeneity in color and geometry). For extreme situations, the refinement criterion evaluates additional parameters of stop, independent of the signal sampled, as those already mentioned in Sec. 4.5.1. We assume the same area for all the pixels.

Filtering The same considerations made in supersampling are valid (Sec. 4.4.1). Due to the stratified system, we use the piecewise-continuous image reconstruction [87] method using a box filter directly incorporated into the color computation phase (tree ascent). For this case, the final color is equivalent to adding each one of the sample color weighted by the area of its corresponding stratum. Other local filters can be applied in this context [44].

4.5.4 RESULTS



(*i*) General view (*ii*) Close-up of (*i*)

Figure 4.9: Reference image used in the test in Fig. 4.10: (*i*) general view and (*ii*) close-up of (*i*). The image has been obtained with a path-tracing algorithm with 1,024 samples per pixel in a stratified way. ([102] © The Canadian Information Processing Society, 2003.)

In Fig. 4.10, we present the results for different techniques applied to the test scene in Fig. 4.9. We compare the following methods:

- *Classic contrast* (CC): A recursive adaptive sampling scheme based on contrast by channel (Equ. 4.1, with thresholds proportional to the relative luminance) weighted by its respective channel color average [44, 118]. The maximum recursive level has been limited to 4 (Fig. 4.10.*a*).
- *Importance-weighted contrast* (IC): The same as in CC but each channel contrast is weighted with the respective importance q (Equ. 4.20), as in the entropy-based contrast approach (Fig. 4.10.*b*).

- *Confidence test* (CT): Statistical approach based on a confidence interval (Equ. 4.5) with a confidence level of $\alpha = 0.1$ and a tolerance $t = 0.025$ (Fig. 4.10.*c*).

- *Entropy-based contrast* (EC): The approach presented in the previous section (Equ. 4.23) but taking only color contrast, that is, with $\delta = 1$ in Equ. 4.21 (Fig. 4.10.*d*).

In CC, IC, and EC, the number of subdivisions is $N_r = 4$ and the number of samples is $N_s^r = 8$. To compute the contrast measures for the refinement decision, the samples have been cast in a stratified way at each n-node (i.e., pixel or subpixel) and re-used at the next levels in the tree. In CT, groups of 8 samples were added in a stratified way until meeting the condition of the criterion. An implementation of path-tracing with next event estimator was used to compute all images [123]. The parameters were tuned so that all four test images were obtained with a similar average number of rays per pixel ($N_s^p = 60$) and computational cost. The resulting images are shown in Figs. 4.10.∗.*i* with close-ups in Figs. 4.10.∗.*ii*. A sampling density map[5] (SDM) for each one is given in Figs. 4.10.∗.*iii*.

The overall aspect of the images in Figs. 4.10.∗.*i* shows that EC scheme performs best. Observe, for instance, the reduced noise in the shadows cast by the objects. This is further checked in the close-up images in Figs. 4.10.∗.*ii*. Observe also the detail of the sphere shadow reflected on the pyramid. It is important to note that we managed to improve the classic contrast approach in CC greatly by including the importance used in the EC scheme (compare results in Fig. 4.10.*a* with Fig. 4.10.*b*). A comparison of the SDMs shows a better discrimination of complex regions of the scene in the entropy case against the classic contrast case. This explains the better results obtained by the EC approach. Moreover, the confidence test approach CT (Fig. 4.10.*c*) also performs better than the classic contrast-based methods CC (Fig. 4.10.*a*) and IC (Fig. 4.10.*b*). The SDM of CT also explains why it performs better. However, it is unable to render the reflected shadows under the mirrored pyramid and sphere with precision (see close-up in Fig. 4.10.*c.ii*).

In Table 4.2, we show two measures (error and quality) obtained from the results shown in Figs. 4.10.∗.*i–ii* with respect to the test scene in Figs. 4.9.*i–ii*, respectively. We select the root of the mean square error[6] (RMSE) and the peak signal to noise ratio[7] (PSNR) to evaluate the results. For each one, we consider a weight balanced by every color channel ($RMSE_a$ and $PSNR_a$) and a perceptual one ($RMSE_p$ and $PSNR_p$) in accordance with the sRGB system (weighted by w^c). These measures reflect the good behaviour of the CT and EC oracles (i.e., low RMSEs and high PSNRs). Although the error obtained using the EC approach is bigger than with the CT method, the visual results are better in the EC case (observe Figs. 4.10.*c–d*). This is due to the fact that the measures do not manage to reflect exactly the perceptual quality of the image. The EC images look better because the oracle distributes the samples more accurately in the most complex regions (see SDMs).

[5] Generated under the same conditions as the quality (Sec. 4.2) and contrast (Sec. 4.3) maps.
[6] It is calculated from the Mean Squared Error (MSE) of each color channel.
[7] Measure of the quality of a reconstructed image compared with an original image computing the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Because many signals have a very wide dynamic range, it is usually expressed in terms of the logarithmic decibel scale: $10 \log_{10}(I_{max}^2/MSE)$ dB.

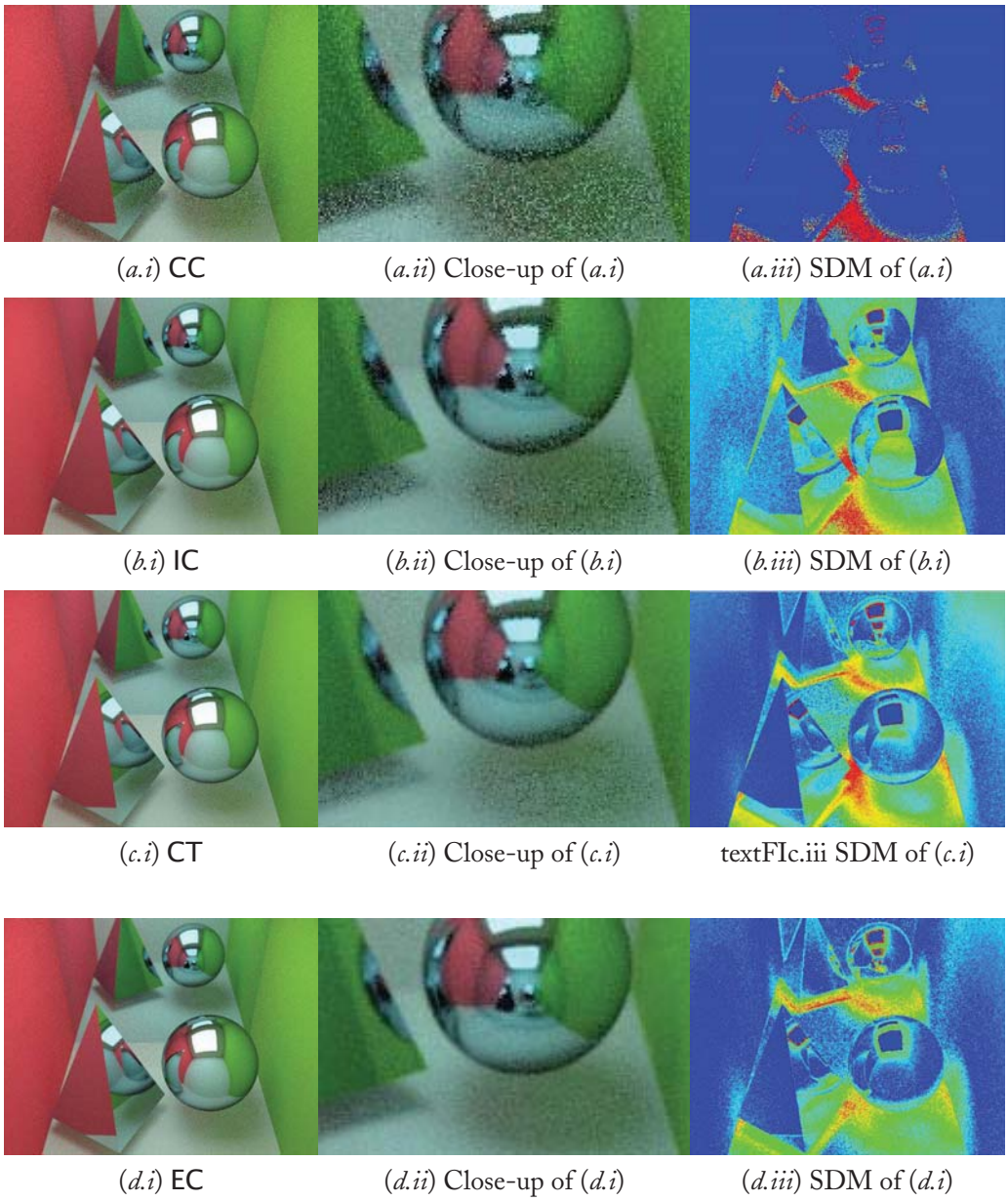


Figure 4.10: (*a*) Adaptive sampling scheme based on classic contrast (CC), (*b*) importance-weighted contrast (IC), same as in (*a*) but weighting with importance q (Equ. 4.20), (*c*) confidence test method (CT), and (*d*) entropy-based method (EC) with color contrast only ($\delta = 1$). By columns: (*i*) shows the resulting images, (*ii*) close-ups of (*i*), and (*iii*) the sampling density maps of (*i*). The average number of rays per pixel is $N_{\bar{s}}^{p} = 60$ in all methods, with a similar computational cost. ([102] © The Canadian Information Processing Society, 2003.)

Good results are also obtained using the geometry component and the binary contrast with color and geometry (see [100]).

Table 4.2: The RMSE and PSNR of CC, IC, CT, and EC oracles applied to the general view (Fig. 4.9.*i*) and close-up (Fig. 4.9.*ii*) of the test scene. The average number of rays per pixel is $N_s^p = 60$ in all methods (see Fig. 4.10).

oracle	general view				close-up			
	$RMSE_a$	$RMSE_p$	$PSNR_a$	$PSNR_p$	$RMSE_a$	$RMSE_p$	$PSNR_a$	$PSNR_p$
CC	13.727	13.599	25.379	25.461	20.276	20.024	21.991	22.100
IC	8.124	8.110	29.935	29.951	13.751	13.568	25.364	25.481
CT	5.194	5.174	33.822	33.855	8.407	8.338	29.638	29.710
EC	6.937	7.018	31.308	31.207	9.886	9.933	28.231	28.189

4.6 *f*-DIVERGENCES IN ADAPTIVE SAMPLING FOR RAY-TRACING

In this section, we apply the *f*-divergences (Sec. 1.8) as refinement criteria in the ray-tracing technique. To do this, we incorporate the divergences into the supersampling scheme using the same basic idea as in hierarchical radiosity (Sec. 2.5) but considering the luminance information instead of the geometric information of the form factors. Therefore, we evaluate the homogeneity of a region of the image plane in accordance with the divergence between its luminance distribution and the uniform distribution.

4.6.1 ALGORITHM

The *f*-divergences defined in Sec. 1.8 will be used to evaluate the heterogeneity of a set of samples in a region. The scheme used is the following:

1. A first batch of N_s^p samples is cast through a pixel and the corresponding luminances $L_{i\in\{1,\dots,N_s^p\}}$ are obtained[8].
2. The *f*-divergences $D_f(p,q)$ are taken between the normalised distribution of the obtained luminances,
$$p = \{p_i = \frac{L_i}{\sum_{j=1}^{N_s^p} L_j} \mid 1 \le i \le N_s^p\}, \tag{4.24}$$
and the uniform distribution $q = \{q_i = 1/N_s^p \mid 1 \le i \le N_s^p\}$.
3. The refinement criterion, given by
$$\frac{1}{N_s^p}\overline{L}\, D_f(p,q) < \epsilon, \tag{4.25}$$

[8] For an sRGB color system, the luminance corresponds to the value of $Y = 0.2126R + 0.7152G + 0.0722B$ [22].

is evaluated, where D_f represents the Kullback-Leibler[9], Chi-square, or Hellinger divergences, $\overline{L}$ is the average luminance

$$\overline{L} = \frac{1}{N_s^p} \sum_{i=1}^{N_s^p} L_i, \tag{4.26}$$

and ϵ is a predefined threshold for the refinement test. The divergence measure $D_f(p, q)$ in Equ. 4.25 plays the role of a contrast. Note that to assign an importance to this value, we weight it with the average luminance (Equ. 4.26), as in Glassner's version of classic contrast [44], used also in the method CC in Sec. 4.5.4, and in the pixel color contrast (Equ. 4.14). The division by the number of samples N_s^p in Equ. 4.25 ensures that the refinement process stops.

4. Successive batches of N_s^p rays are cast until the result of the test is true and no more refinement is necessary.

The new criteria give good visual results, but the RMSE obtained in the tests (see Table 4.3), although better than for the classic contrast, is higher than with the confidence test criterion (Equ. 4.5 and CT in Sec. 4.5.4). A next logical step is to try the square root of Hellinger divergence, as it is a true metric. The results obtained are very encouraging and, by analogy, the experimentation can be extended to the square root of the other divergences. The results also improved the previous ones and were also better than in the confidence test case. The square root versions of this set of *f*-divergences have already been used previously in statistics [97, 145, 130].

Definition 4.12. Three refinement criteria for adaptive ray-tracing, based on their respective f-divergences (Sec. 1.8), are given by

- Square root of Kullback-Leibler divergence

$$\frac{1}{N_s^p} \overline{L} D_{KL}^{\frac{1}{2}}(p, q) < \epsilon, \tag{KL$^{\frac{1}{2}}$}$$

- Square root of Chi-square divergence

$$\frac{1}{N_s^p} \overline{L} D_{\chi^2}^{\frac{1}{2}}(p, q) < \epsilon, \tag{CS$^{\frac{1}{2}}$}$$

- Square root of Hellinger divergence

$$\frac{1}{N_s^p} \overline{L} D_{h^2}^{\frac{1}{2}}(p, q) < \epsilon. \tag{HL$^{\frac{1}{2}}$}$$

[9]Note that in this particular case, i.e., obtaining the divergence with respect to the constant distribution $1/N_s^p$, KL divergence reduces to entropy contrast (Equ. 4.12), save for the log N_s^p normalization constant and the direct use of luminances instead of color channel. Thus, results of KL are consistent with the ones of entropy-based contrast (EC) when compared to confidence test (CT) and classic contrast (CC).

4.6.2 RESULTS

In Fig. 4.11 and Fig. 4.12, we present comparative results with different techniques for the test scene in Fig. 4.9. The following two methods are compared with the three f-divergence-based criteria ($\mathsf{KL}^{\frac{1}{2}}$, $\mathsf{CS}^{\frac{1}{2}}$, and $\mathsf{HL}^{\frac{1}{2}}$):

- *Classic contrast* (CC): Contrast of Equ. 4.1 of the luminance weighted with the importance $\overline{L}$.
- *Confidence test* (CT): Confidence level of $\alpha = 0.1$ and tolerance $t = 0.025$ in Equ. 4.5 (as in Sec. 4.5.4).

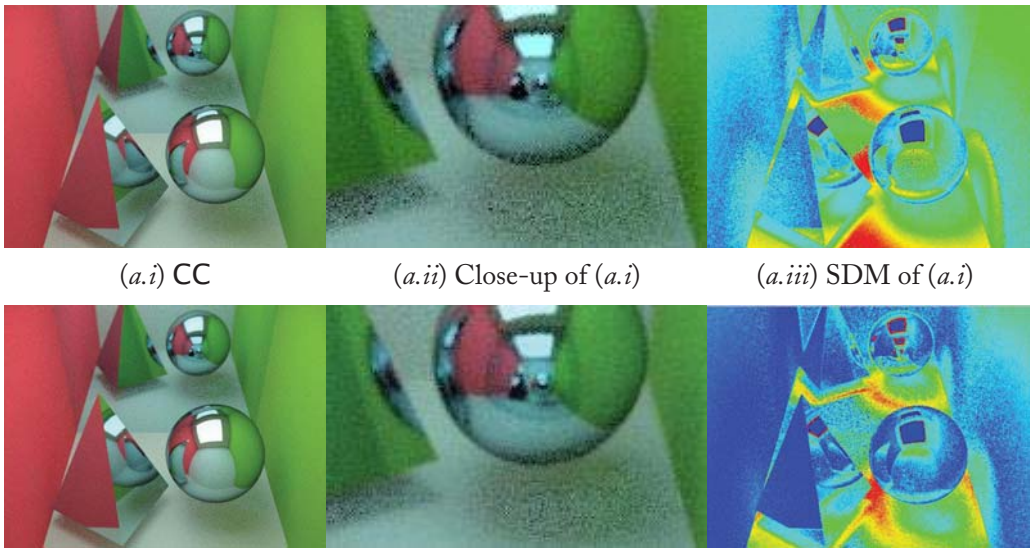
In order to evaluate their behaviour, the images are generated by a similar process to that of adaptive sampling ray-tracing in Sec. 4.4. However, a geometric hierarchical subdivision is not developed; in all the methods, 8 initial rays are cast in a stratified way at each pixel to compute the contrast measures for the refinement decision, and 8 additional rays are successively added until the condition of the criterion is met. An implementation of classic path-tracing with next event estimator was used to compute all images. The parameters were tuned so that all five test images were obtained with a similar average number of rays per pixel ($N_{\bar{s}}^{p} = 60$) and a similar computational cost. The reconstruction method applied is the piecewise-continuous image [87] with box filter. Finally, the pixel value is the reconstructed signal average at pixel domain.

The resulting images are shown in Figs. 4.11.$*$.*i* (CC and CT) and Figs. 4.12.$*$.*i* ($\mathsf{KL}^{\frac{1}{2}}$, $\mathsf{CS}^{\frac{1}{2}}$, and $\mathsf{HL}^{\frac{1}{2}}$), with the sampling density maps in Figs. 4.11.$*$.*iii* and Figs. 4.12.$*$.*iii*, respectively. The analysis of the critical regions of the images shows how the proposed sampling scheme performs the best. Observe, for instance, the reduced noise in the shadows cast by the objects. Observe also the detail of the shadow of the sphere reflected on the pyramid.

Comparison of the SDMs shows a better discrimination of the most complex regions of the scene in the three divergence cases against the classic contrast and confidence test cases. This explains the better results obtained by the f-divergence approach. On the other hand, the confidence test approach also performs better than the classic contrast-based method. Its SDM also explains why it performs better than the contrast-based method. However, it is unable to render the reflected shadows under the mirrored pyramid and sphere with enough precision (*ii*).

In Table 4.3, we show the RMSE and PSNR of the images obtained with classic (Figs. 4.11.$*$.*i*), f-divergence, and square root of f-divergence (Figs. 4.12.$*$.*i*) methods with respect to the test image in Fig. 4.9. Visual comparison is in concordance with numerical data. The f-divergence-based criteria used in the experiments outperform both classic contrast and confidence test experiments. Finally, the better results of the $\mathsf{HL}^{\frac{1}{2}}$ criterion could be explained by the fact that the square root of the Hellinger distance is a true metric.

As a general conclusion, by evaluating the visual (see Fig. 4.10, Fig. 4.11, and Fig. 4.12) and quantitative (see Table 4.2 and Table 4.3) results, we observe that both entropy-based contrast and square root versions of f-divergences criteria present the best improvements to the whole visual aspect of the image, dealing better with the most complex regions and spending less samples in the

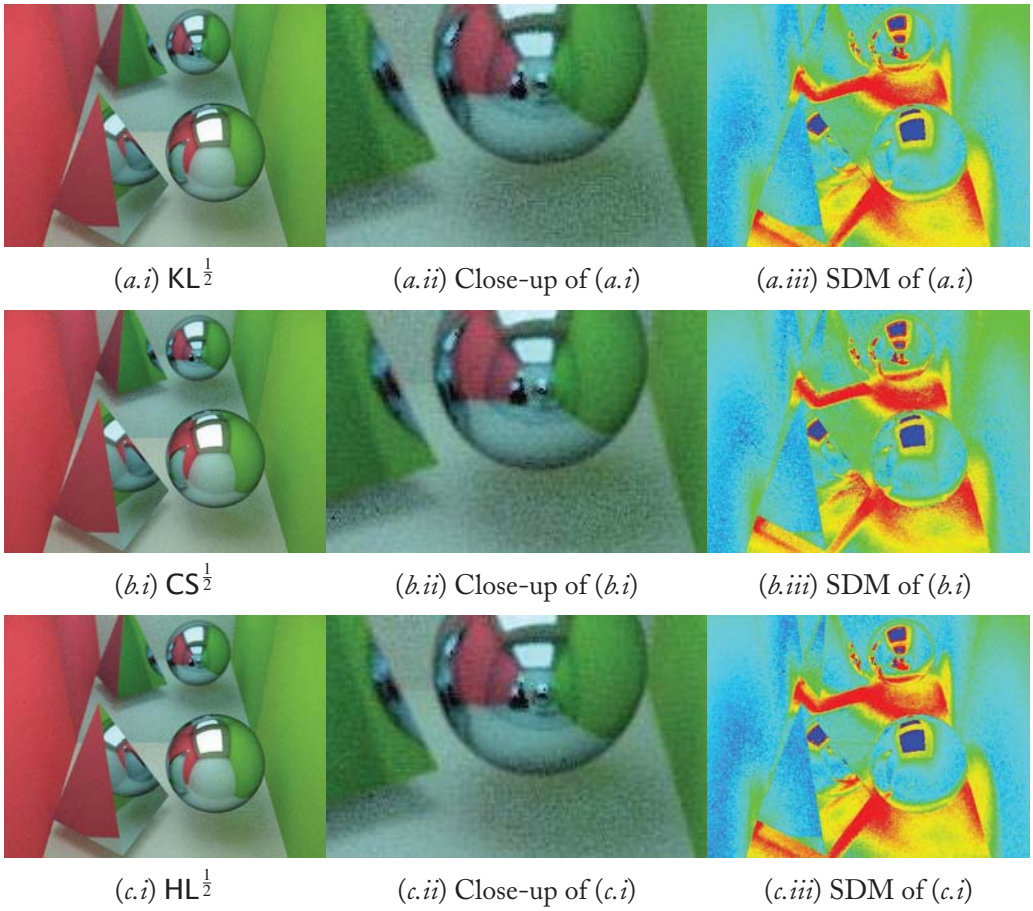


(*a.i*) CC (*a.ii*) Close-up of (*a.i*) (*a.iii*) SDM of (*a.i*)

(*b.i*) CT (*b.ii*) Close-up of (*b.i*) (*b.ii*) SDM of (*b.i*)

Figure 4.11: Images of the test scene (Fig. 4.9) obtained with an adaptive sampling scheme based on (*a*) classic contrast (CC) and (*b*) confidence test (CT) methods. By columns, (*i*) shows the resulting images, (*ii*) close-ups of (*i*), and (*iii*) the sampling density maps of (*i*). The average number of rays per pixel is 60. Compare with the images in Fig. 4.12. ([102] © The Canadian Information Processing Society, 2003.)

least complex ones. This is in spite of the RMSE and PSNR values of the entropy case are slightly worse than for the confidence test. The criteria based on the square root versions of *f*-divergences, specially, the Hellinger one, give the best results, both on image quality and numerical error.



(*a.i*) $\mathsf{KL}^{\frac{1}{2}}$ (*a.ii*) Close-up of (*a.i*) (*a.iii*) SDM of (*a.i*)

(*b.i*) $\mathsf{CS}^{\frac{1}{2}}$ (*b.ii*) Close-up of (*b.i*) (*b.iii*) SDM of (*b.i*)

(*c.i*) $\mathsf{HL}^{\frac{1}{2}}$ (*c.ii*) Close-up of (*c.i*) (*c.iii*) SDM of (*c.i*)

Figure 4.12: Images of the test scene (Fig. 4.9) obtained with an adaptive sampling scheme based on the square root of (*a*) Kullback-Leibler ($\mathsf{KL}^{\frac{1}{2}}$), (*b*) Chi-square ($\mathsf{CS}^{\frac{1}{2}}$), and (*c*) Hellinger ($\mathsf{HL}^{\frac{1}{2}}$) f-divergences. By columns, (*i*) shows the resulting images, (*ii*) close-ups of (*i*), and (*iii*) the sampling density maps of (*i*). The average number of samples per pixel is 60 in all the methods. Compare with the images in Fig. 4.11. ([103] © The Eurographics Association, 2003.)

Table 4.3: The RMSE and PSNR measures of the CC, CT, and f-divergence-based refinement criteria applied to Fig. 4.9. The images for the CC and CT methods are shown in Fig. 4.11, and for the f-divergence-based ones, in Fig. 4.12. The average number of samples per pixel is $N_s^p = 60$.

method	oracle	$RMSE_a$	$RMSE_p$	$PSNR_a$	$PSNR_p$
Classic	Contrast (CC)	6.157	6.126	32.344	32.387
	Confidence test (CT)	5.194	5.174	33.822	33.855
f-divergences	Kullback-Leibler (KL)	5.508	5.553	33.311	33.241
	Chi-square (CS)	5.414	5.452	33.461	33.400
	Hellinger (HL)	5.807	5.862	32.852	32.770
Square root of f-divergences	Kullback-Leibler ($KL^{\frac{1}{2}}$)	4.824	4.793	34.463	34.519
	Chi-square ($CS^{\frac{1}{2}}$)	4.772	4.736	34.557	34.623
	Hellinger ($HL^{\frac{1}{2}}$)	4.595	4.560	34.884	34.951

CHAPTER 5

Viewpoint Selection and Mesh Saliency

The basic question underlying the viewpoint selection study and application is "what is a *good* viewpoint of a scene?" Obviously, this question does not have a simple answer. Depending on our objective, the best viewpoint can be, for instance, the most representative one or the most unstable one (i.e., the one that maximally changes when it is moved within its close neighborhood [8]). Palmer et al. [88] and Blanz et al. [6] have presented different experiments demonstrating that observers prefer views (called canonical views) that avoid occlusions and that are off-axis (such as a three-quarter viewpoint), salient (the most significant characteristics of an object are visible), stable, and with a large number of visible surfaces.

In computer graphics, several viewpoint quality measures have been applied in areas such as scene understanding [93, 135, 94], scene exploration [2, 122], image-based modeling [136], and volume visualization [8, 124, 140]. In other areas, such as object recognition and mobile robotics, best view selection is also a fundamental task. Many works have demonstrated that the recognition process is view-dependent [88, 9, 128, 6]. Tarr et al. [128] found that "visual recognition may be explained by a view-based theory in which viewpoint-specific representations encode both quantitative and qualitative features".

In this chapter, a set of viewpoint measures that can be grouped in a unified framework are presented to deal with viewpoint selection and mesh saliency [135, 111, 34]. Given a set of viewpoints surrounding an object, an information channel is defined between the viewpoints and the polygons of the object. From this channel, the viewpoint entropy, the viewpoint mutual information, and the viewpoint Kullback-Leibler distance are defined to obtain the best views of an object. In particular, viewpoint mutual information is also used to calculate the stability of a viewpoint and to guide the object exploration. The channel is reversed and both the information and the saliency associated with each polygon are computed. Finally, this polygonal saliency is used to calculate how salient a viewpoint is, and it is incorporated into viewpoint mutual information to drive the viewpoint selection.

5.1 BACKGROUND

We review now some related work that is not based on information theory. In Plemenos and Benayada [93], the quality of a viewpoint v of a scene is computed using the following heuristic measure.

Definition 5.1. The heuristic measure (HM) is defined by

$$C(v) = \frac{\sum_{i=1}^{n} \lceil \frac{P_i(v)}{P_i(v)+1} \rceil}{n} + \frac{\sum_{i=1}^{n} P_i(v)}{r} \quad , \tag{5.1}$$

where $P_i(v)$ is the number of pixels corresponding to the polygon i in the image obtained from the viewpoint v, r is the total number of pixels of the image (resolution of the image), and n is the total number of polygons of the scene[1]. The first term in Equ. 5.1 is the fraction of visible polygons with respect to the total number of polygons, while the second term is the ratio between the projected area of the scene (or object) and the screen area (thus, its value is 1 for a closed scene).

Polonsky et al. [94] describe a number of different ways to measure the goodness of a view of an object. After analyzing different view descriptors, they conclude that no single descriptor does a perfect job and possibly a combination of them would amplify the advantage that each one has. Given a sphere of viewpoints, Yamauchi et al. [144] compute the similarity between each pair of disjoint views using Zernike moments analysis and obtain a similarity weighted spherical graph. A view is considered to be stable if all incident edges on its viewpoint in the spherical graph have high similarity weights. Sokolov et al. [122] present two different exploration algorithms guided by the total curvature of a visible surface.

Based on the investigation on canonical views, Gooch et al. [46] present a new method for constructing images where the viewpoint is chosen to be off-axis, and Lu et al. [75] obtain the viewing direction from the combination of factors such as saliency, occlusion, stability, and familiarity. Lee et al. [71] have introduced the saliency as a measure for regional importance for graphics meshes, and Kim and Varshney [65] presented a visual-saliency-based operator to enhance selected regions of a volume. Gal and Cohen-Or [39] introduced a method for partial matching of surfaces by using the abstraction of salient geometric features and a method to construct them.

5.2 VIEWPOINT CHANNEL

In this section, an information channel (see Sec. 1.1) between a set of viewpoints and the set of polygons of an object is introduced to define a set of viewpoint measures.

5.2.1 VIEWPOINT ENTROPY AND MUTUAL INFORMATION

A viewpoint selection framework is constructed from an information channel $V \rightarrow Z$ between the random variables V (input) and Z (output), which represent, respectively, a set of viewpoints $\mathcal{V}$ and the set of polygons $\mathcal{Z}$ of an object (see Fig. 5.1.*left*) [34]. This channel, called viewpoint channel, is defined by a transition probability matrix obtained from the projected areas of polygons at each viewpoint. Viewpoints will be indexed by v and polygons by z. Following the convention introduced in Sec. 1.1, the capital letters V and Z as arguments of $p(.)$ will be used to denote probability

[1] $\lceil x \rceil$ denotes the smallest integer, greater than or equal to x.

distributions. For instance, while $p(v)$ will denote the probability of a single viewpoint v, $p(V)$ will represent the input distribution of the set of viewpoints.

The viewpoint channel can be interpreted as an observation channel where the conditional probabilities represent the probability of "seeing" a determined polygon from a given viewpoint (see Fig. 5.1.*right*). The three basic elements of this channel are:

- Transition probability matrix $p(Z|V)$, where each element $p(z|v) = a_z(v)/A_T(v)$ is defined by the normalized projected area of polygon z over the sphere of directions centered at viewpoint v, $a_z(v)$ is the projected area of polygon z at viewpoint v, and $A_T(v)$ is the projected area of all polygons over the sphere of directions. Conditional probabilities fulfil $\sum_{z\in\mathcal{Z}} p(z|v) = 1$. In this chapter, the background is not taken into account, but it could be considered as another polygon.

- Input distribution $p(V)$, which represents the probability of selecting each viewpoint, will be obtained from the normalization of the projected area of the object at each viewpoint: $p(v) = A_T(v)/\sum_{v\in\mathcal{V}} A_T(v)$. This can be interpreted as the probability that a random ray originated at v hits (or "sees") the object. This assignation is consistent with the objective of selecting the viewpoints which "see" more projected area. Let us remember that this is a characteristic of a canonical view. The input distribution can also be interpreted as the importance assigned to each viewpoint v. For instance, the input distribution could be defined by $p(v) = 1/N_v$, where N_v is the number of viewpoints and, in this case, the same importance would be assigned to each viewpoint.

- Output distribution $p(Z)$ is given by

$$p(z) = \sum_{v\in\mathcal{V}} p(v)\,p(z|v), \tag{5.2}$$

  which represents the average projected area of polygon z (i.e., the probability of polygon z to be hit or "seen" by a random ray cast from the viewpoint sphere).

From the previous definitions and Equations 1.2, 1.5, and 1.9, Shannon's information measures can be defined for the viewpoint channel. We first introduce the viewpoint entropy [135] and the viewpoint conditional entropy [34].

Definition 5.2. The viewpoint entropy (VE) of viewpoint v is defined by

$$H(Z|v) = -\sum_{z\in\mathcal{Z}} p(z|v)\log p(z|v). \tag{5.3}$$

$H(Z|v)$ measures the degree of uniformity of the projected area distribution at viewpoint v. In this book, $H(Z|v)$ will be also denoted as $H(v)$. The maximum viewpoint entropy is obtained when a

$p(Z\|V)$		$\mathcal{Z}$			$p(V)$
		z_1	$\cdots$	z_m	
$\mathcal{V}$	v_1	$p(z_1\|v_1)$	$\cdots$	$p(z_m\|v_1)$	$p(v_1)$
	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
	v_n	$p(z_1\|v_n)$	$\cdots$	$p(z_m\|v_n)$	$p(v_n)$
$p(Z)$		$p(z_1)$	$\cdots$	$p(z_m)$	

Figure 5.1: On the left, a polygonal model is surrounded by a "sphere of viewpoints" obtained from the recursive decomposition of an icosahedron. On the right, the viewpoint information channel is represented by the transition probability matrix $p(Z|V)$ and the marginal probability distributions $p(V)$ and $p(Z)$, where n and m represent the number of viewpoints and polygons, respectively.

certain viewpoint can see the maximum number of polygons with the same projected area. The best viewpoint is defined as the one that has maximum entropy [135].

Definition 5.3. The conditional entropy of channel $V \rightarrow Z$ is defined by

$$H(Z|V) = -\sum_{v\in\mathcal{V}} p(v) \sum_{z\in\mathcal{Z}} p(z|v) \log p(z|v) = \sum_{v\in\mathcal{V}} p(v) H(Z|v). \tag{5.4}$$

That is, $H(Z|V)$ is given by the average of all viewpoint entropies. Both measures $H(Z|v)$ and $H(Z|V)$ tend to infinity when the polygons are infinitely refined (see Equ. 1.35). This makes these measures very sensitive to the discretization of the object.

We now focus our attention to the mutual information between V and Z, that expresses the degree of dependence or correlation between the set of viewpoints and the object [34].

Definition 5.4. The mutual information of channel $V \rightarrow Z$ is defined by

$$I(V;Z) = \sum_{v\in\mathcal{V}} p(v) \sum_{z\in\mathcal{Z}} p(z|v) \log \frac{p(z|v)}{p(z)} = \sum_{v\in\mathcal{V}} p(v) I(v;Z), \tag{5.5}$$

where $I(v;Z)$ is the viewpoint mutual information defined as follows:

Definition 5.5. The viewpoint mutual information (VMI) of viewpoint v is defined by

$$I(v;Z) = \sum_{z\in\mathcal{Z}} p(z|v) \log \frac{p(z|v)}{p(z)}. \tag{5.6}$$

$I(v;Z)$ gives us the degree of dependence between the viewpoint v and the set of polygons, and is

interpreted as a measure of the quality of viewpoint v. Consequently, the mutual information $I(V;Z)$ gives us the average quality of the set of viewpoints. Here, High quality is considered equivalent to high representativeness, which corresponds to the lowest VMI values. Thus, the best viewpoint is defined as the one that has minimum VMI. This is due to the fact that the lowest values correspond to the most representative or relevant views, showing the maximum possible number of polygons in a balanced way. On the other hand, high values of the measure mean a high dependence between viewpoint v and the object, indicating a highly coupled view (for instance, between the viewpoint and a small number of polygons with low average visibility).

Related to this framework, the viewpoint Kullback-Leibler distance [111] can be defined from Equ. 1.7.

Definition 5.6. The viewpoint Kullback-Leibler distance (VKL) of viewpoint v is defined by

$$D_{KL}(p(Z|v), a(Z)) = \sum_{z \in \mathcal{Z}} p(z|v) \log \frac{p(z|v)}{a(z)}, \tag{5.7}$$

where $a(z)$ is the normalized area of polygon z obtained from the area of polygon z divided by the total area of the object. The VKL measure is interpreted as the distance between the normalized distribution of projected areas and the normalized distribution of the areas of polygons. Note that, in this case, the background can not be taken into account. The minimum value 0 would be obtained when $p(z|v) = a(z)$. Since the target is in this case to look for the distribution of projected areas as much near as possible to the distribution of actual areas, selecting best views means to minimize VKL.

We have to note that VMI can be rewritten in the following way: $I(v;Z) = D_{KL}(p(Z|v), p(Z))$. It is worth observing that $p(Z)$ plays the role of the target distribution in the Kullback-Leibler distance (Equ. 1.7) and, therefore, the role of the optimal distribution since the objective is that $p(Z|v)$ becomes similar to $p(Z)$ to obtain the best views. On the other hand, this role agrees with intuition since $p(Z)$ is the average visibility of polygon z over all viewpoints (i.e., like a mixed distribution of all views), and we can think of $p(Z)$ as representing, with a single distribution, the knowledge about the scene. Note the difference between VMI (Equ. 5.6) and VKL (Equ. 5.7), due to the fact that in the last case the distance is taken with respect to the actual areas of polygons.

In Viola et al. [140], it has been shown that the main advantage of VMI over VE is its robustness to deal with any type of discretisation or resolution of the volumetric dataset. The same advantage can be observed for polygonal data. Thus, while a highly refined mesh will attract the attention of VE, VMI will be almost insensitive to changes in the mesh resolution. In general, if we compare both measures for finer and finer discretisations, VMI will converge to an upper bound and VE will increase to infinity (see Sec. 1.6) [32, 34]. Note that the heuristic measure HM (Equ. 5.1) is also highly dependent on the discretisation since the first term in Equ. 5.1 is given by the quotient

between the number of visible polygons and the total number of polygons. The behavior of all these measures, with respect to the discretisation, will be experimentally shown in the next section.

5.2.2 RESULTS

In this section, the behavior of VMI (Equ. 5.6) is compared with the one of HM (Equ. 5.1), VE (Equ. 5.3), and VKL (Equ. 5.7). To compute these viewpoint quality measures, we need a preprocess step to estimate the projected area of the visible polygons of the object at each viewpoint. Before projection, a different color is assigned to each polygon. The number of pixels with a given color divided by the total number of pixels projected by the object gives us the relative area of the polygon represented by this color (conditional probability $p(z|v)$).

In this chapter, all measures have been computed without taking into account the background and using a projection resolution of 640×480[2]. In all the experiments, the objects are centered in a sphere of 642 viewpoints built from the recursive discretisation of an icosahedron, and the camera is looking at the center of this sphere. This framework could be extended to any other placement of viewpoints but the choice of a sphere of viewpoints allows us to analyze an object in an isotropic manner. Note that all the measures analyzed here are sensitive to the relative size of the viewpoint sphere with respect to the object. The viewpoint sphere is built in the following way: first, the smallest bounding sphere of the model is obtained, and then, the viewpoint sphere adopts the same center as the bounding sphere and a radius three times the radius of the bounding sphere. The experiments shown in this chapter have been created using the 3D-rendering engine Ogre3D (`http://www.ogre3d.org`) and run on a 3GHz machine with 2Gb RAM and an NVIDIA GeForce 8800-GTX with 768Mb.

Four models have been used: a cow (Fig. 5.2.*a*), a coffee-cup-and-dish with two different discretisations of the dish (Fig. 5.4.*a.i* and Fig. 5.4.*b.i*), a ship (Fig. 5.2.*b*), and the lady of Elche (Fig. 5.2.*c*). In Table 5.1 we show the number of polygons of the models used in this section and the cost of the preprocess step (i.e., the cost of computing the probability distributions $p(V)$, $p(Z|V)$ and $p(Z)$). Even though a large number of viewpoints have been used, an acceptable quality could be achieved with less viewpoints and the consequent reduction of timings. To show the behavior of the measures, the sphere of viewpoints is represented by a thermic scale where red and blue colors correspond, respectively, to the best and worst views. Note that a high quality viewpoint corresponds to a high value for both HM (Equ. 5.1) and VE (Equ. 5.3), and to a low value for both VKL (Equ. 5.7) and VMI (Equ. 5.6).

Fig. 5.3 has been organized as follows. Rows (*a*), (*b*) and (*c*) show, respectively, the behavior of HM, VE, and VMI measures. Columns (*i*) and (*ii*) show, respectively, the best and worst views, and columns (*iii*) and (*iv*) show two different projections of the viewpoint spheres. Fig. 5.3 illustrates how VMI selects better views than both HM and VE. Observe how VE chooses to "see" the most highly discretised parts of the cow. The same occurs with HM, although this one also searches for a

[2]For practical purposes, we use projection on the tangent plane instead of on the sphere of directions.

Table 5.1: Number of triangles of the models used and computational time (seconds) of the preprocess step for each model. Models are shown in Fig. 5.2 and Fig. 5.4.*a.i*.

	Cow	Coffee cup	Ship	Lady of Elche
Number of triangles	9,593	43,935	47,365	51,978
Computational time	41	81	62	80

(*a*) (*b*) (*c*)

Figure 5.2: (a) Cow, (b) ship and (c) lady of Elche wireframe models. ([34] © ACM, 2009.)

view with higher projected area. While the worst views for HM and VE correspond to the ones that see the less discretised parts, in the VMI case, a more restricted (or low quality) view is obtained.

Fig. 5.4 shows the behavior of HM, VE, and VMI measures when the discretisation of the object varies outstandingly. Rows (*a*) and (*b*) show the viewpoint spheres computed, respectively, for the coffee-cup-and-dish model of Fig. 5.4.*a.i* and for the same model with a more refined dish (Fig. 5.4.*b.i*). We can clearly observe how the spheres obtained from HM and VE change according to the discretisation variation, whereas VMI spheres are almost insensitive to this variation. The different behavior between VKL and VMI is shown in Fig. 5.5. Remember that the main difference between VMI and VKL is that while the former computes the distance between the projected areas of polygons, and their average area "seen" by the set of viewpoints, the latter calculates the distance with respect to the actual areas of polygons. Due to this fact, the reliability of VKL is much affected by the existence of many non visible or poorly visible polygons, as in the case of the ship and lady of Elche models.

5.3 VIEWPOINT SIMILARITY AND STABILITY

As we have mentioned in the introduction of this chapter, a basic property of a canonical view is its stability [6]. That is, observers prefer a view which minimally changes when it is moved within its nearest neighborhood. In this section, viewpoint stability is defined from the notion of dissimilarity between two viewpoints, which is given by the Jensen-Shannon divergence between their respective distributions (Equ. 1.16). The use of Jensen-Shannon divergence as a measure of view similarity has

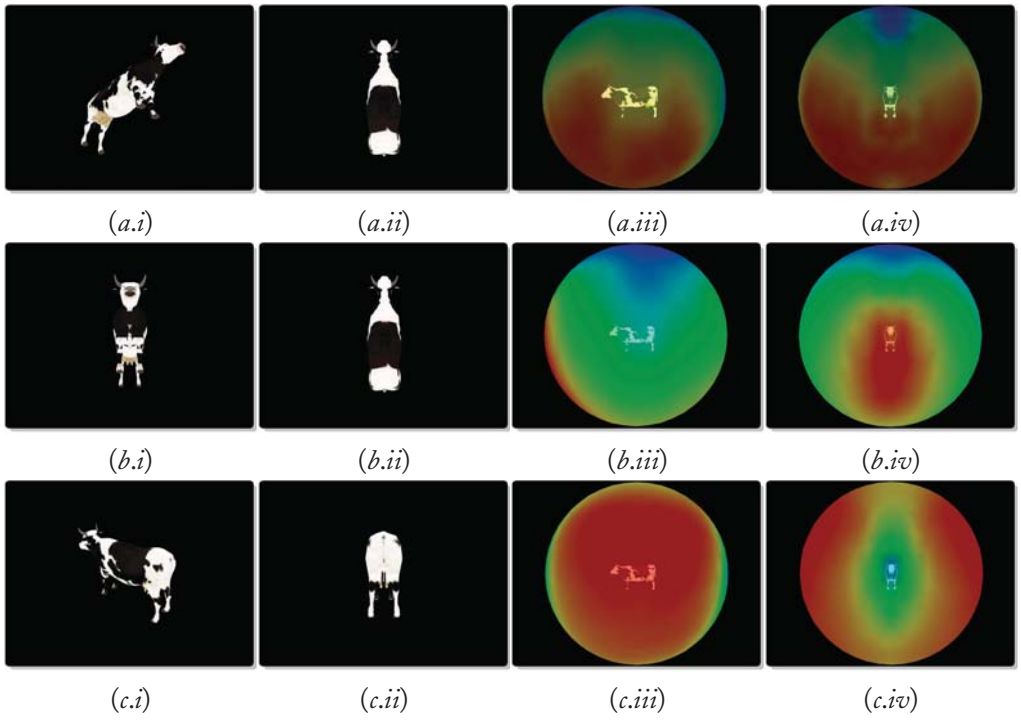


Figure 5.3: (*i*) The most representative (high quality) and (*ii*) the most restricted (low quality) views, and (*iii-iv*) the viewpoint spheres obtained, respectively, from the (*a*) HM, (*b*) VE, and (*c*) VMI measures. Red colors on the sphere represent the highest quality viewpoints and blue colors represent the lowest quality viewpoints. ([34] © ACM, 2009.)

been proposed by Bordoloi and Shen [8] within the volume rendering field. From the information channel introduced in Sec. 5.2.1, the viewpoint dissimilarity appears naturally from the variation of mutual information [34].

If we apply the data processing inequality (Equ. 1.18) or the information bottleneck method (Sec. 1.7) to the channel $V \rightarrow Z$, we find that any clustering over V or Z, respectively, denoted by $\widehat{V}$ and $\widehat{Z}$, will reduce $I(V;Z)$. Therefore, if neighbor viewpoints (or polygons) are clustered, then $I(\widehat{V};Z) \leq I(V;Z)$ (or $I(V;\widehat{Z}) \leq I(V;Z)$). The result of clustering (or merging) two viewpoints v_i and v_j is defined as a "virtual" viewpoint $\widehat{v}$ such that

$$p(\widehat{v}) = p(v_i) + p(v_j) \tag{5.8}$$

and the conditional probability becomes

$$p(z|\widehat{v}) = \frac{p(v_i)p(z|v_i) + p(v_j)p(z|v_j)}{p(\widehat{v})}. \tag{5.9}$$

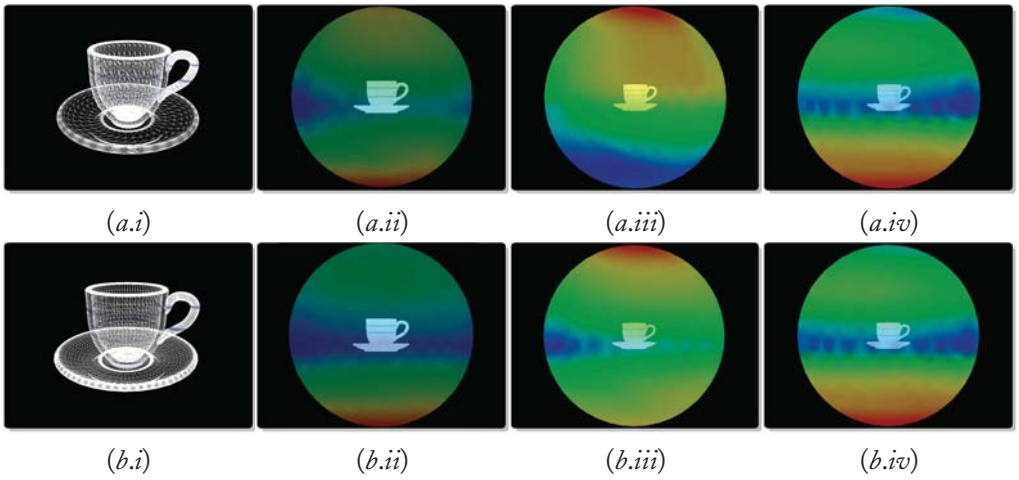


Figure 5.4: Column (*i*) shows the models used to compute the viewpoint spheres where the dish in (*b.i*) is more refined than the one in (*a.i*). The viewpoint spheres are obtained, respectively, from (*ii*) HM, (*iii*) VE, and (*iv*) VMI measures. ([34] © ACM, 2009.)

The reduction of mutual information when two viewpoints v_i and v_j are merged is given by

$$\begin{aligned}\delta I &= I(V;Z) - I(\widehat{V};Z) = \big(p(v_i)I(v_i;Z) + p(v_j)I(v_j;Z)\big) - p(\widehat{v})I(\widehat{v};Z) \\ &= p(\widehat{v})\left(\frac{p(v_i)}{p(\widehat{v})}I(v_i;Z) + \frac{p(v_j)}{p(\widehat{v})}I(v_j;Z) - I(\widehat{v};Z)\right) = p(\widehat{v})D(v_i,v_j), \end{aligned} \quad (5.10)$$

where $D(v_i, v_j)$ is the viewpoint dissimilarity defined as:

Definition 5.7. The viewpoint dissimilarity between v_i and v_j is defined by

$$D(v_i,v_j) = \frac{p(v_i)}{p(\widehat{v})}I(v_i;Z) + \frac{p(v_j)}{p(\widehat{v})}I(v_j;Z) - I(\widehat{v};Z). \quad (5.11)$$

Hence, the loss of information when two viewpoints are merged is given by the weighted dissimilarity between them. It can be seen that the dissimilarity will be null when both viewpoints capture the same distribution of projected areas: if $p(Z|v_i) = p(Z|v_j)$, then $\delta I = 0$.

From the definition of the Jensen-Shannon divergence (Equ. 1.16) and the information bottleneck method (Sec. 1.7), it can be shown that the viewpoint dissimilarity can also be written as

$$D(v_i,v_j) = JS\left(\frac{p(v_i)}{p(\widehat{v})}, \frac{p(v_j)}{p(\widehat{v})}; p(Z|v_i), p(Z|v_j)\right), \quad (5.12)$$

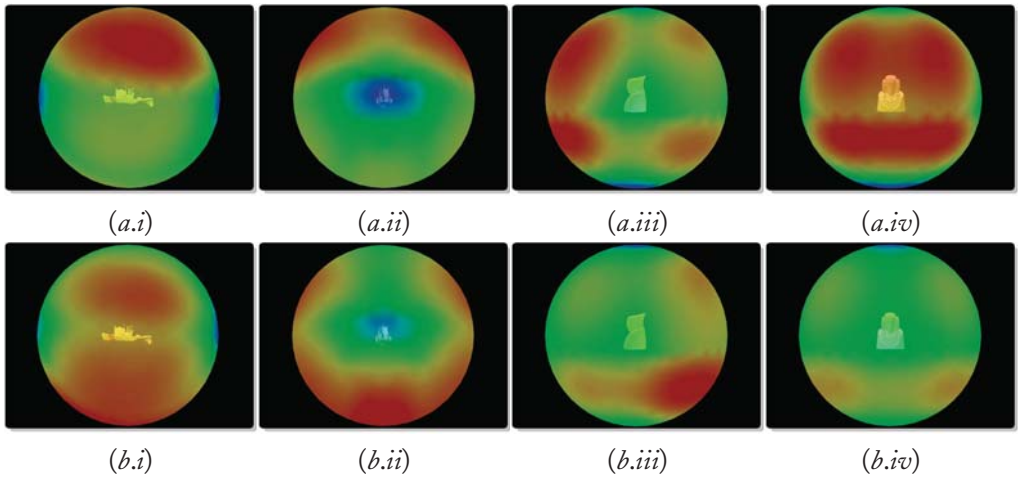


Figure 5.5: Viewpoint spheres obtained, respectively, from (*a*) VKL and (*b*) VMI measures. ([34] © ACM, 2009.)

where the second term is the Jensen-Shannon divergence between the distributions $p(Z|v_i)$ and $p(Z|v_j)$ captured by v_i and v_j with weights $p(v_i)/p(\widehat{v})$ and $p(v_j)/p(\widehat{v})$, respectively. If two views are very similar (i.e., the JS-divergence between them is small), the channel could be simplified by substituting these two viewpoints by their merging, without a significant loss of information. It can be seen that the clustering $\widehat{V}$ of all viewpoints would give $\delta I = I(V;Z)$ and, thus, $I(\widehat{V};Z) = 0$.

View unstability was defined in [8] as the maximum change in view that occur when the camera position is shifted within a small neighborhood. Thus, a small change corresponds to a stable viewpoint and a large change to an unstable one. Here, the unstability of a viewpoint v is defined as the average variation of dissimilarity between v and its neighbor viewpoints. That is, v_i is stable if $p(Z|v_i)$ is close to the probability distributions $p(Z|v_j)$ of its neighbors, where v_j stands for a neighbor of v_i.

Definition 5.8. The viewpoint unstability of viewpoint v_i is defined by

$$U(v_i) = \frac{1}{N}\sum_{j=1}^{N} D(v_i, v_j), \tag{5.13}$$

where v_j is a neighbor of v_i and N is the number of neighbors of v_i.

Fig. 5.6 shows the behavior of the viewpoint unstability measure for the coffee-cup-and-dish, cow, and lady of Elche models. Observe how the results obtained agree with intuition.

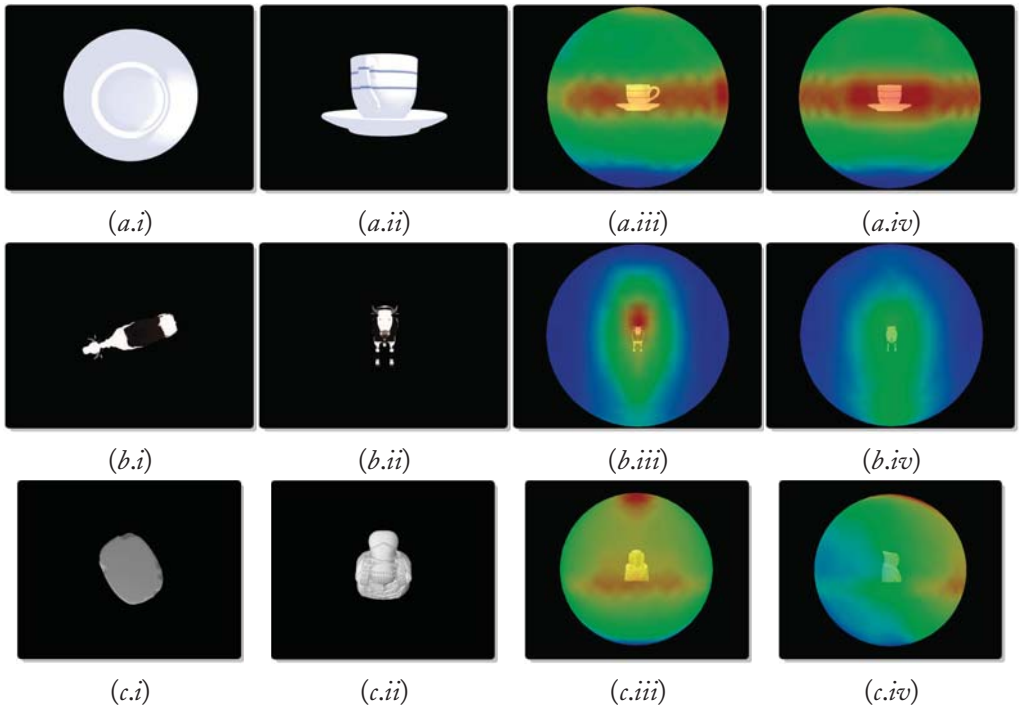


Figure 5.6: The (*i*) most stable and (*ii*) most unstable viewpoints, and (*iii-iv*) the unstability spheres obtained for the (*a*) coffee-cup-and-dish, (*b*) cow and (*c*) lady of Elche models. Red colors on the sphere represent high unstability values, blue colors represent low unstability values. ([34] © ACM, 2009.)

5.4 BEST VIEW SELECTION AND OBJECT EXPLORATION

In order to understand or model an object, we are interested in selecting a set of representative views which provides a complete representation of the object. In this section, algorithms based on viewpoint mutual information are applied to both the selection of the N best representative views and object exploration [34]. Other algorithms of viewpoint selection and scene exploration based on viewpoint entropy can be seen in [135, 134, 136, 2].

5.4.1 SELECTION OF N BEST VIEWS

With the goal of obtaining the best representation of the object using the minimum number of views, a VMI-based viewpoint selection algorithm is presented. If we look for a good set of views within the set of viewpoints, we will obtain the most representative set by selecting the views such that their mixing (merging) minimizes VMI. This mixing provide us with a balanced view of the object.

Thus, the algorithm proposed should select the N viewpoints so that their merging $\widehat{v}$ minimizes the viewpoint mutual information $I(\widehat{v}; Z)$, that is, the Kullback-Leibler distance between $p(Z|v)$ and the target distribution $p(Z)$. Due to the fact that this optimization algorithm is NP-complete, a greedy strategy is adopted by selecting successive viewpoints that minimize $I(\widehat{v}; Z)$. This algorithm permits us to find in an automated and efficient way a minimal set of views which represent the object or scene.

The algorithm proceeds as follows. First, we select the best viewpoint v_1 with distribution $p(Z|v_1)$ corresponding to the minimum $I(v; Z)$. Next, we select v_2 such that the mixed distribution $(p(v_1)/p(\widehat{v}))p(Z|v_1) + (p(v_2)/p(\widehat{v}))p(Z|v_2)$ will minimize $I(\widehat{v}; Z)$, where $\widehat{v}$ represents the clustering of v_1 and v_2 and $p(\widehat{v}) = p(v_1) + p(v_2)$. At each step, a new mixed distribution $(p(v_1)/p(\widehat{v}))p(Z|v_1) + (p(v_2)/p(\widehat{v}))p(Z|v_2) + \ldots + (p(v_n)/p(\widehat{v}))p(Z|v_n)$, where $p(\widehat{v}) = p(v_1) + p(v_2) + \ldots + p(v_n)$, is produced until the VMI-ratio given by $I(\widehat{v}; Z)/I(V; Z)$ is lower than a given threshold or a fixed number of views is achieved. This ratio can be interpreted as a measure of the goodness or representativeness of the selected viewpoints.

Fig. 5.7 shows the six best views obtained with the VMI-based selection algorithm for three different models. In Table 5.2, for each new viewpoint selected we show the viewpoint mutual information $I(\widehat{v}; Z)$ of the clustering of selected viewpoints and the corresponding VMI-ratio. For instance, to achieve a degree of representativeness given by a VMI-ratio lower than 0.15, four views are needed for the coffee-cup-and-dish and lady of Elche models, and five for the armadillo model. Table 5.2 also shows the computation cost of selecting the six best views. It is important to note that the best views for the selected models (Fig. 5.7.*a*) are not the ones our intuition would expect as more representative. This is due to the fact that, from a purely geometric approach, the best views of Fig. 5.7 correspond to the viewpoints such that their projected area distribution is more similar (in the Kullback-Leibler sense) to the average projected area distribution (target distribution). This problem will be tackled in Sec. 5.6, showing how to introduce additional criteria to select the best views.

From the N best representative viewpoints, a simple greedy clustering algorithm is proposed in order to partition the sphere of viewpoints. The two main steps of this algorithm are the following. First, we select the N best viewpoints from a given VMI-ratio. These viewpoints will play the role of centroids in the algorithm. Second, each viewpoint is assigned or clustered with the nearest centroid, where the distance is given by the Jensen-Shannon divergence between two viewpoints. The behavior of this clustering algorithm is shown in Fig. 5.8 for the (a) coffee cup, (b) cow, (c) ship, and (d) lady of Elche models.

5.4.2 OBJECT EXPLORATION

In this section, a greedy algorithm is presented to explore an object. In this algorithm, the path visits a set of N preselected best views which ensure a good exploration of the object.

First, we obtain the list of the N best viewpoints. Then, the algorithm starts at the best viewpoint and visits all the other best viewpoints as follows. From the best viewpoint, we find

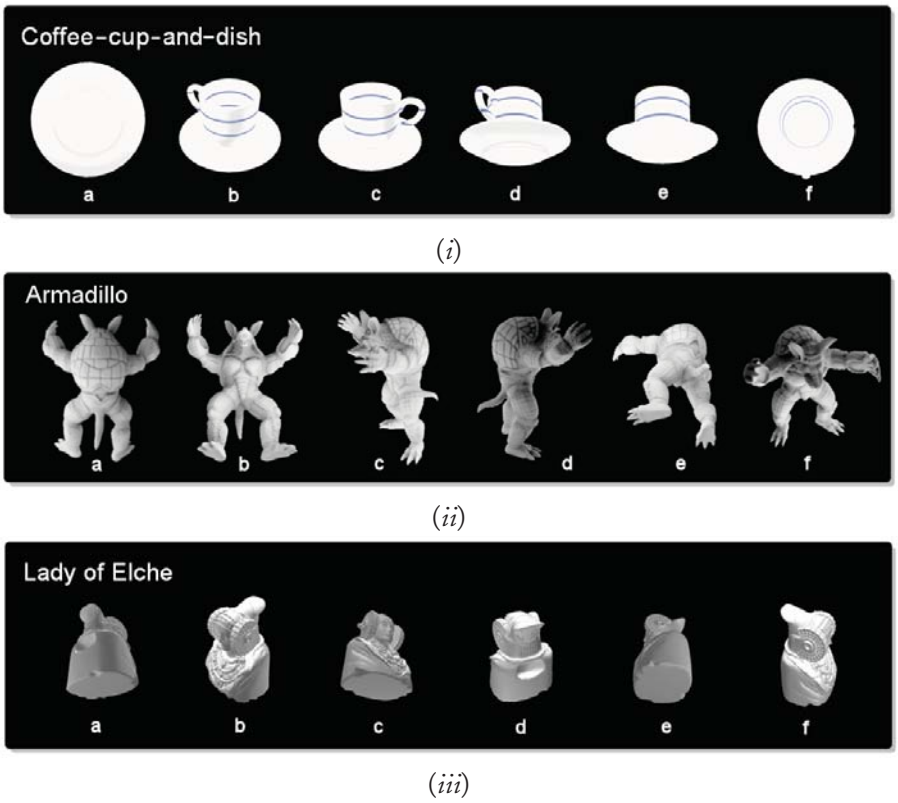
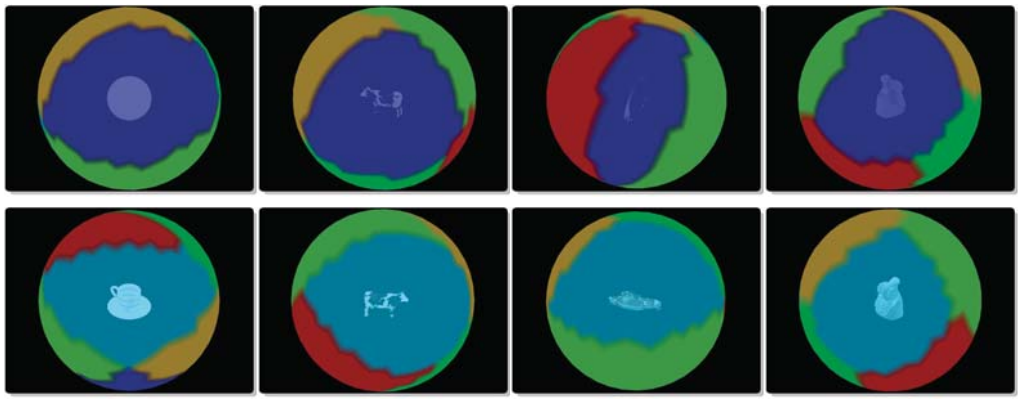


Figure 5.7: From (*a*) to (*f*), the six most representative views selected by the VMI-based algorithm for the (*i*) coffee-cup-and-dish, (*ii*) armadillo, and (*iii*) lady of Elche models. ([34] © ACM, 2009.)

the nearest (minimum JS-divergence) best viewpoint in the list. This is now the target viewpoint. Then, from the best viewpoint, successive neighbor viewpoints will be selected so that, without any viewpoint repetition, their distance to the target viewpoint is minimum. The distance between two viewpoints is always calculated from the JS-divergence. When the first target viewpoint is achieved, we select a new target one among the rest of best viewpoints in the list. Then we proceed in the same way until the last best view is reached or the cycle is completed, arriving at the initial best viewpoint. This algorithm, being a greedy one, is fast but it can cause a small detour over the minimum path. Fig. 5.9.*i* shows the exploration of the coffee-cup-and-dish and the lady of Elche models from the six best views obtained in each case (the blue, green and red light points correspond to the starting, intermediate and ending viewpoints, respectively). Two different projections of the sphere are shown to see better the trajectory.

Table 5.2: For the coffee cup, armadillo, and lady of Elche models, the values of the pair $(I(\widehat{v}; Z)$, VMI-ratio) are shown after the selection of a new viewpoint. The six best views for each model can be seen in Fig. 5.7. The computational time (seconds) of selecting these views is also shown.

Best view	Coffe cup	Armadillo	Lady of Elche
a	(1.471, 0.730)	(1.791, 0.850)	(1.355, 0.703)
b	(0.692, 0.343)	(0.837, 0.397)	(0.644, 0.334)
c	(0.346, 0.172)	(0.616, 0.292)	(0.458, 0.237)
d	(0.262, 0.130)	(0.416, 0.197)	(0.275, 0.143)
e	(0.207, 0.103)	(0.310, 0.147)	(0.219, 0.113)
f	(0.190, 0.095)	(0.238, 0.113)	(0.153, 0.079)
Time	36	77	38

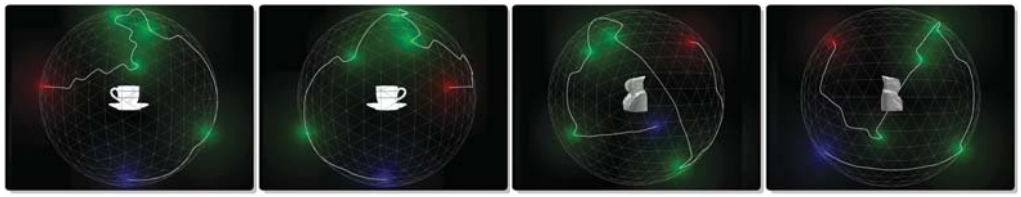


(*a*) (*b*) (*c*) (*d*)

Figure 5.8: Viewpoint clustering spheres with six clusters for the (*a*) coffee-cup-and-dish, (*b*) cow, (*c*) ship and (*d*) lady of Elche models. ([34] © ACM, 2009.)

5.5 VIEW-BASED POLYGONAL INFORMATION AND SALIENCY

In Sec. 5.2, we have seen that the information associated with each viewpoint has been obtained from the definition of a channel between the sphere of viewpoints and the polygons of the object. Now, we want to introduce the information associated with a polygon, which is defined as the contribution of this polygon to the MI of that channel. To illustrate this approach, the reversed channel $Z \to V$ is considered, so that Z is now the input and V the output [34]. Remember that MI is invariant to the reversion of the channel: $I(V; Z) = I(Z; V)$.



(*a*) (*b*) (*c*) (*d*)

Figure 5.9: Exploration around the (*a–b*) coffee-cup-and-dish and (*c–d*) lady of Elche models, respectively. ([34] © ACM, 2009.)

5.5.1 VIEW-BASED POLYGONAL INFORMATION

From the Bayes theorem $p(v, z) = p(v)p(z|v) = p(z)p(v|z)$, the mutual information (Equ. 5.5) can be rewritten as

$$I(Z; V) = \sum_{z \in \mathcal{Z}} p(z) \sum_{v \in \mathcal{V}} p(v|z) \log \frac{p(v|z)}{p(v)} = \sum_{z \in \mathcal{Z}} p(z) I(z; V), \tag{5.14}$$

where $I(z; V)$ is defined as the polygonal mutual information.

Definition 5.9. The polygonal mutual information (PMI) of polygon z is defined by

$$I(z; V) = \sum_{v \in \mathcal{V}} p(v|z) \log \frac{p(v|z)}{p(v)}. \tag{5.15}$$

PMI represents the degree of correlation between the polygon z and the set of viewpoints, and can be interpreted as the information associated with polygon z. Analogously to the behavior of VMI, low values of PMI correspond to polygons that "see" the maximum number of viewpoints in a balanced way, that is, $p(V|z)$ is close to $p(V)$. The opposite happens for high values.

In Fig. 5.10, we show the polygonal information maps of (*a*) the coffee-cup-and-dish, (*b*) car, (*c*) Hebe, and (*d*) lady of Elche models. To obtain these images, the PMI has been normalized between 0 and 1 and subtracted from 1. Thus, low values of PMI, corresponding to non-occluded or visible (from many viewpoints) polygons, are represented by values near 1 in the grey-map, while high values of PMI, corresponding to occluded polygons, are represented by values near 0 in the grey-map. In Fig. 5.10, we show the polygonal information values computed from the center of each polygon, while in Fig. 5.11, these values have been linearly interpolated at the vertices of the polygons. Observe that these maps look as an ambient occlusion or obscurance map[3] (see [70, 16, 148, 57]). In Fig. 5.11, we show one example of the use of polygonal information as ambient occlusion where this is added to a textured model.

[3] Ambient occlusion is a powerful technique that mimics indirect global illumination at a fraction of its cost. Zhukov et al. [148, 57] introduced obscurances, the first ambient-occlusion technique, in the computer-game context to allow fast editing, and later used it in production rendering.

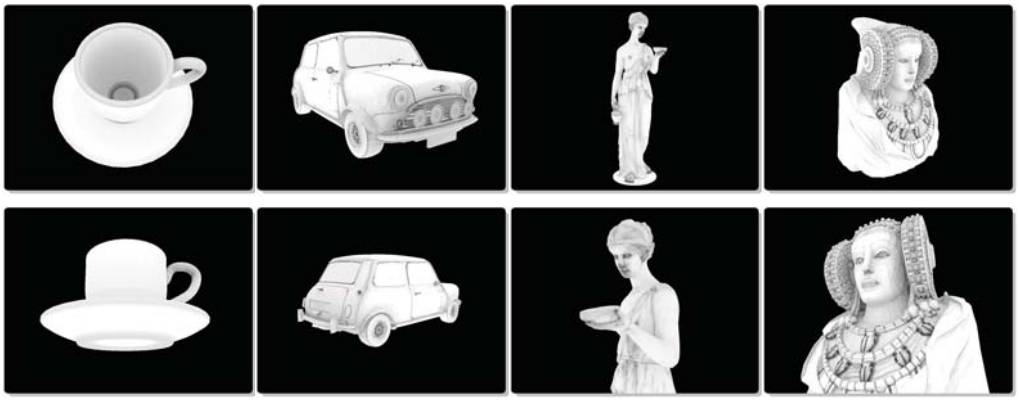


Figure 5.10: View-based polygonal information for the (*a*) coffee-cup-and-dish, (*b*) car, (*c*) Hebe, and (*d*) lady of Elche models. ([34] © ACM, 2009.)

Figure 5.11: The figure shows the application of polygonal information as ambient occlusion. ([34] © ACM, 2009.)

5.5.2 VIEW-BASED MESH SALIENCY

Itti et al. [58] maintain that visual attention is saliency-dependent, and use a saliency map to represent the conspicuity or saliency at every location in the visual field by a scalar quantity and to guide the selection of attended locations. In [71], mesh saliency, applied to mesh simplification and viewpoint selection, is captured from surface curvatures and is considered as a perception-inspired measure of regional importance.

Analogous to the view unstability (Sec. 5.3), defined from the dissimilarity between two views, the view-based mesh saliency is defined from the dissimilarity between two polygons, which is given by the variation of mutual information when two polygons are clustered. In this approach, mesh saliency is formulated in terms of how the polygons "see" the set of viewpoints. Thus, following the same scheme developed in Sec. 5.3, the saliency of a polygon is defined as the average dissimilarity between this polygon and its neighbors.

Similarly, to Equ. 5.10, the reduction of mutual information when two polygons z_i and z_j are clustered is given by

$$\begin{aligned}\delta I &= I(Z;V) - I(\widehat{Z};V) = (p(z_i)I(z_i;V) + p(z_j)I(z_j;V)) - p(\widehat{z})I(\widehat{z};V) \\ &= p(\widehat{z})\left(\frac{p(z_i)}{p(\widehat{z})}I(z_i;V) + \frac{p(z_j)}{p(\widehat{z})}I(z_j;V) - I(\widehat{z};V)\right) = p(\widehat{z})D(z_i,z_j), \end{aligned} \quad (5.16)$$

where $\widehat{z}$ is the result of clustering z_i and z_j and $D(z_i,z_j)$ is defined as the polygonal dissimilarity between z_i and z_j:

Definition 5.10. The polygonal dissimilarity between polygons z_i and z_j is defined by

$$D(z_i,z_j) = \frac{p(z_i)}{p(\widehat{z})}I(z_i;V) + \frac{p(z_j)}{p(\widehat{z})}I(z_j;V) - I(\widehat{z};V). \quad (5.17)$$

This polygonal dissimilarity can also be written (see information bottleneck method in Sec. 1.7) as

$$D(z_i,z_j) = JS\left(\frac{p(z_i)}{p(\widehat{z})}, \frac{p(z_j)}{p(\widehat{z})}; p(V|z_i), p(V|z_j)\right), \quad (5.18)$$

where the second term is the Jensen-Shannon divergence (Equ. 1.16) between $p(V|z_i)$ and $p(V|z_j)$ with weights $p(z_i)/p(\widehat{z})$ and $p(z_j)/p(\widehat{z})$, respectively. Hence, two polygons are "similar" when the JS-divergence between them is small.

Similarly, to the unstability of a viewpoint (Equ. 5.13), the saliency of a polygon is defined as the average variation of the polygonal dissimilarity between a polygon and its neighbors.

Definition 5.11. The polygonal saliency of polygon z_i is defined by

$$S(z_i) = \frac{1}{N}\sum_{j=1}^{N} D(z_i,z_j) \geq 0, \quad (5.19)$$

where z_j is a neighbor polygon of z_i and N is the number of neighbor polygons of z_i. Thus, a polygon z will be salient when the average of JS-divergences between z and its neighbors is high. For example, a polygon at the center of a smooth region will have probably low saliency since the polygons of this region will present small visibility differences with respect to the set of viewpoints. Fig. 5.12 shows the behavior of the saliency measure. The most salient parts are represented in red and the least salient ones in blue. For instance, the handle of the coffee cup and the nose, mouth and eyes of the other models are the most salient surfaces.

Similarly, to Lee et al. [71], where mesh saliency was used to select the best views, a method to calculate the saliency of a viewpoint is now proposed. After calculating the saliency of polygons,

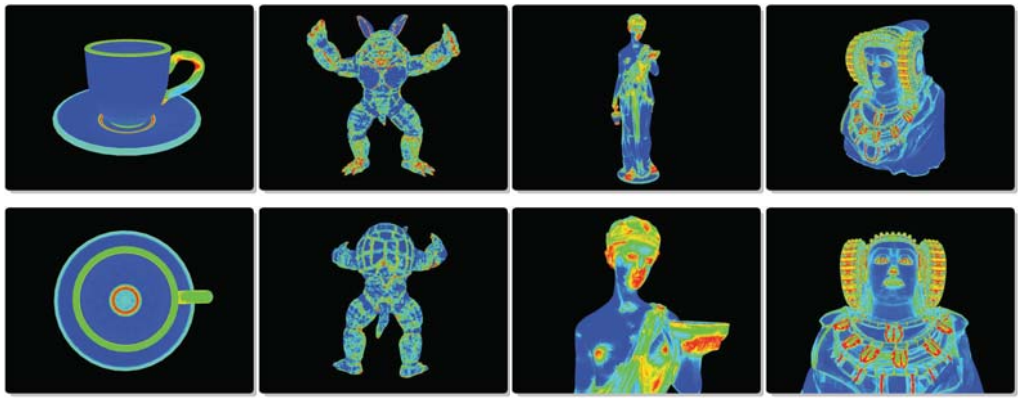


Figure 5.12: Mesh saliency for the (*a*) coffee-cup-and-dish, (*b*) armadillo, (*c*) Hebe, and (*d*) lady of Elche models. ([34] © ACM, 2009.)

we can now convey or project this information to the sphere of viewpoints using the conditional probabilities of the reverse channel.

Definition 5.12. The viewpoint saliency is defined by

$$S(v) = \sum_{z \in \mathcal{Z}} S(z)\, p(v|z). \tag{5.20}$$

Fig. 5.13 shows the viewpoint saliency for the coffee-cup-and-dish, armadillo and lady of Elche models. Columns (*i*) and (*ii*) illustrate the most salient view and the least one, respectively. Columns (*iii*) and (*iv*) show two different projections of the corresponding saliency spheres. Observe how the most salient views show us the most salient parts of each object.

5.6 IMPORTANCE-DRIVEN VIEWPOINT SELECTION

As we have mentioned in the introduction of this chapter, it is desirable that a canonical view of an object shows its most salient parts and also the largest number of visible surfaces [88, 6]. However, the viewpoint quality measure VMI only takes into account the geometric relationship between the object and the set of viewpoints. Therefore, we can not expect that, in general, the best VMI-based views fulfill the desired properties for a canonical view. This fact motivates the investigation of how perceptual criteria such as saliency can be introduced into the viewpoint mutual information measure in order to improve the automatic selection of good views [34].

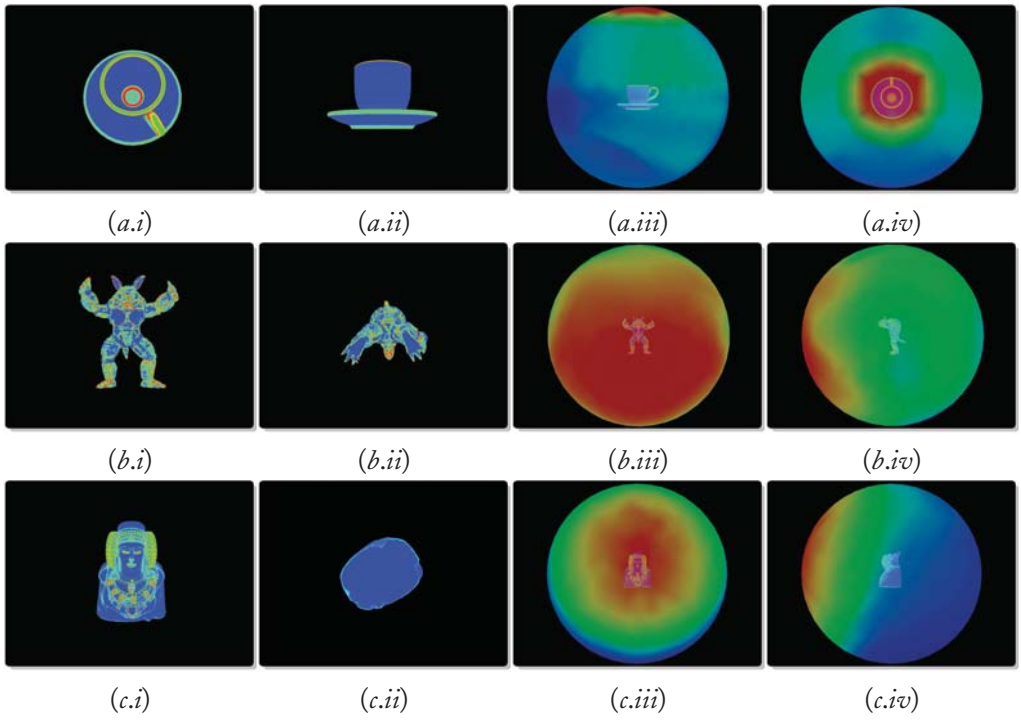


Figure 5.13: The (*i*) most salient and (*ii*) least salient views, and (*iii-iv*) saliency spheres obtained for the (*a*) coffee-cup-and-dish, (*b*) armadillo, and (*c*) lady of Elche models. Red colors on the sphere represent high saliency values, blue colors represent low saliency values. ([34] © ACM, 2009.)

In the previous section, we have presented a method to compute how salient a viewpoint is, but we now aim to incorporate the polygonal saliency to the viewpoint mutual information in order to take into account different factors concerning, respectively, the amount of projected area, the geometric representativeness, and the saliency of a polygon. First, we show how the importance can be introduced into the object space by modifying directly the target distribution $p(Z)$. Second, we present the results obtained by the use of the polygonal saliency as an importance factor in the viewpoint mutual information measure.

Due to the fact that VMI represents the distance between the projected visibility distribution $p(Z|v)$ at viewpoint v and the target distribution $p(Z)$, VMI can be extended by weighting the target distribution with an importance factor. Thus, adding importance to VMI means simply weighting the original target distribution by an importance factor in order to obtain the new target distribution.

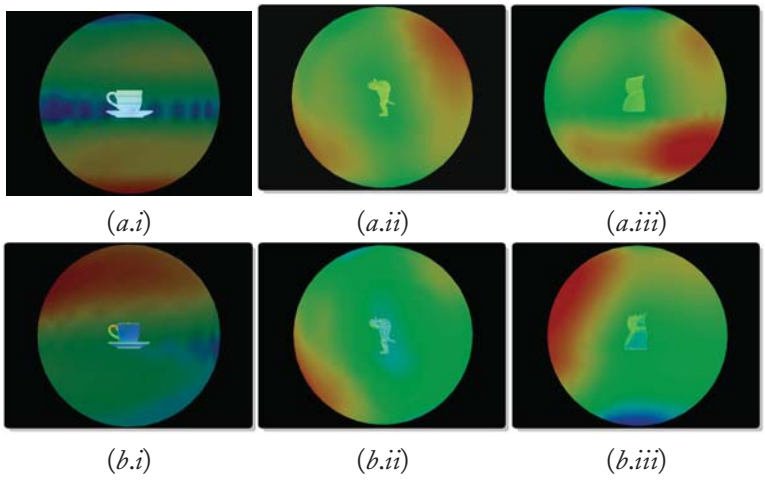


Figure 5.14: (*a*) VMI and (*b*) saliency-based EVMI spheres for the (*i*) coffee-cup-and-dish, (*ii*) armadillo, and (*iii*) lady of Elche models. ([34] © ACM, 2009.)

The optimal viewpoint would be the one viewing every polygon proportional to its average projected area multiplied by its importance.

Definition 5.13. The extended viewpoint mutual information (EVMI) is defined by

$$I'(v;\, Z) = \sum_{z \in \mathcal{Z}} p(z|v) \log \frac{p(z|v)}{p'(z)} = D_{KL}(p(Z|v),\, p'(Z)), \tag{5.21}$$

where

$$p'(z) = \frac{p(z) imp(z)}{\sum_{z \in \mathcal{Z}} p(z) imp(z)}, \tag{5.22}$$

and $imp(z)$ is the importance of polygon z. In the experiments of this section, $imp(z)$ has been substituted by the polygonal saliency $S(z)$. We follow the convention that if $imp(z) = 0$ then polygon z should not be taken into account[4]. Other features, such as illumination, could be introduced as importance factors in the EVMI. In Viola et al. [140], the object importance has been used to calculate the best views for a volumetric dataset (see Sec. 6.2.2). Note that, if importance is taken as $imp(z) = a(z)/p(z)$, where $a(z)$ is the normalized area of polygon z, then $I'(v;\, Z) = D_{KL}(p(Z|v), a(Z))$ (i.e., EVMI turns into VKL). That is, the more occluded the polygon the more important it is. Observe that when $p(z) = 0$, $imp(z)$ is not defined. This is not a problem as, in the sum of Sec. 5.21, terms with $p(z) = 0$ are null.

[4]To avoid having to recalculate all probabilities, we can demand $imp(z) > 0$.

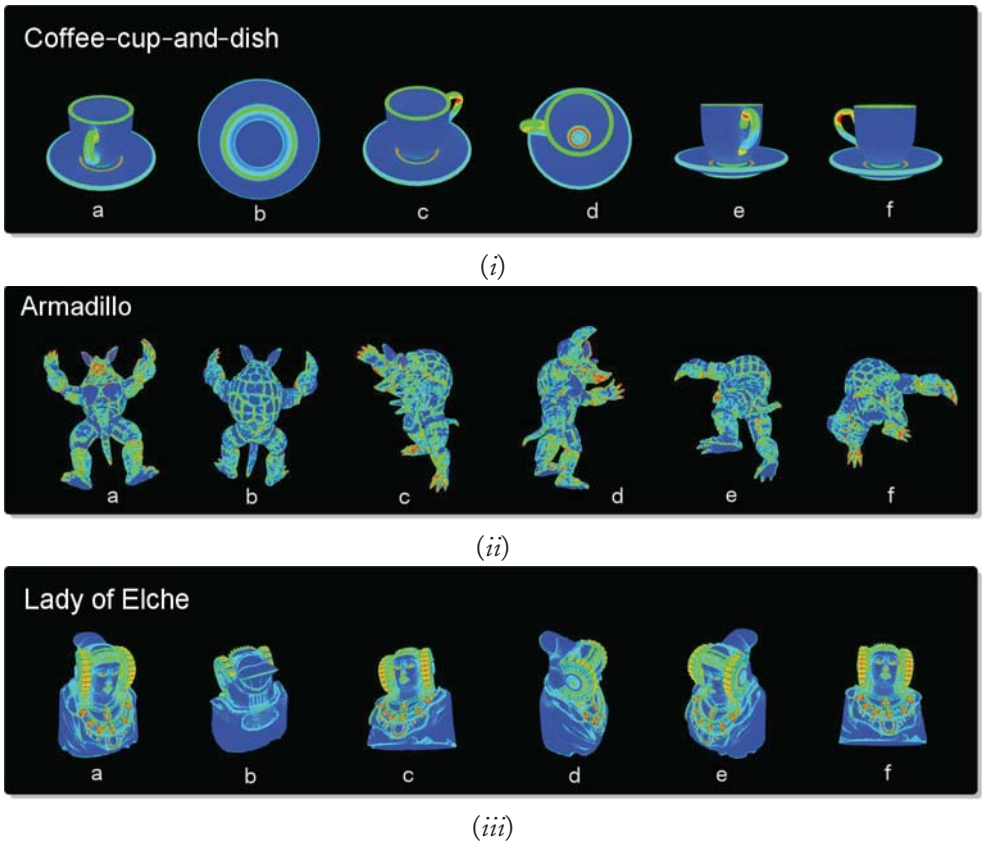


Figure 5.15: The six most representative views for the (*i*) coffee-cup-and-dish, (*ii*) armadillo, and (*iii*) lady of Elche models using the saliency-based EVMI algorithm. ([34] © ACM, 2009.)

The effects of incorporating saliency in VMI are illustrated in Fig. 5.14 and Fig. 5.15, which show for the coffee-cup-and-dish, armadillo, and lady of Elche models the saliency-based EVMI spheres and the six most representative views, obtained with the best view selection algorithm (Sec. 5.4.1), where VMI has been substituted by the saliency-based EVMI. The saliency-based EVMI spheres of Fig. 5.14.*b* show the perceptual improvement obtained with respect to the corresponding VMI spheres (Fig. 5.14.*a*). For instance, whereas the VMI-based best view of the coffee-cup-and-dish shows the bottom of the dish (Fig. 5.7.*i.a*), the best view based on EVMI shows a lateral view of the coffee-cup-and-dish (Fig. 5.15.*i.a*) which is perceptually much better than the one of Fig. 5.7.*i.a*. The same conclusion can be obtained for the armadillo and the lady of Elche (see the respective best views shown in Fig. 5.7 and Fig. 5.15).

CHAPTER 6

View Selection in Scientific Visualization

Many disciplines in science and industry generate huge amounts of digital data on a daily basis. These data are acquired from various kinds of measurements or simulations. Typical examples include tomographic scans from diagnostic medicine or reservoir simulations from the oil industry. Similar examples can be found in climate research, oceanology, astronomy, and other scientific disciplines. These domains use various approaches to understand the hidden insights behind the acquired numbers. Sometimes, fully automatic algorithms can be used to quantitatively obtain insights from the data, sometimes methods which are borrowed from statistics are utilized. However, acquired data is often so complex that quantitative measures are not effective in providing insights, and more qualitative analytical methods have to be employed.

For a human observer, visual perception covers approximately 80% of the entire sensory input. Therefore, the human visual system is generally considered to be the most effective channel to convey complex information, such as shape or flow characteristics of objects. Data sets from above mentioned domains encode information about human anatomy or spatial characteristics of interesting seismic layers. In analytical software tools, the visual channel is exploited to look into such data. The term *scientific visualization* expresses approaches that provide insights by way of visual means into complex computerized information acquired in scientific disciplines. Visual elements that represent these complex data utilize computer graphics technology to be brought-up to the computer screen for the user's investigation. Visualization is, therefore, often seen as computer graphics technology applied on particular types of data in order to convey what the digital data represents.

Information theory, applied to three-dimensional computer graphics data representations, provides measures to evaluate the quality of viewpoints from which a given scene is rendered. This accommodation of information-theoretic measures has been used in the previous chapter for estimating the most informative viewpoint or a minimal set of viewpoints that characterize a given polygonal scene in the most expressive way. Automatic selection of the most informative viewpoint is a very useful focusing mechanism in visualization of scientific data. The viewer can be immediately guided to the most interesting information avoiding cumbersome tuning of viewpoint parameters. Alternatively, a selection of the most informative viewpoints can be used, for example, for generation of a data storyboard (i.e., a compact representation of the information the data contains).

Approaches for estimation of the most informative viewpoints for scientific data sets are similar to those developed for polygonal data; however, there are differences that arise due to the different focus of scientific visualization as compared to more general computer graphics. One difference to polygonal computer graphics is that the underlying data is generally more complex. Scientific data

are generated by measurements and simulations that have very heterogeneous output. Some data types can be aligned to a structured grid, simulations usually generate unstructured finite element meshes, while other data types are entirely devoid of connectivity information and are represented as cloud of sparse points. Medical data sets, for example, are often represented as scalar values per sample point. In other areas, such as meteorological or flow simulations, several attributes per sample point are common. Many natural phenomena are studied through their development in time. These studies are stored as time-varying data sets, for which visualization is often the only effective way of providing insights. Moreover, scientific data often differ in the level of associated semantics. Some data contain solely measurement in numerical values without any a-priori knowledge about structures; other data sets contain information about the most relevant structures such as critical points in flow data or segmentation masks of anatomical objects. Consequently, the heterogeneity among scientific data types, as opposed to simple polygonal scene, corresponds to the heterogeneity in visualization approaches for viewpoint quality evaluation.

A second important aspect of scientific visualization, as opposed to standard computer graphics, is that it serves the purpose of gaining insights into complex phenomena. There is always a purpose that drives visualization, attempting to find answers to open questions of a particular scientific domain. Here, visualization can serve three purposes: (a) exploration of new unknown data seeking for structures and forming hypotheses, (b) visual analysis of partly known data to validate already drawn hypotheses, and (c) visual presentation of validated findings from the data for the purpose of dissemination of knowledge gained during the explorative or analytical process. Computer-guided view selection seems to be a promising tool for addressing all three visualization scenarios, that is, knowledge gain, knowledge validation, as well as knowledge dissemination.

6.1 ADAPTATION FROM POLYGONS TO VOLUMES

Automatic view selection has been originally designed for polygonal data. Volumetric data sets, in general, consist of an order of magnitude more data elements as compared to a scene described by polygons. Therefore, to be able to use view selection approaches with other data types, modifications to original techniques are necessary. In case of scalar volumetric data, there are essentially two criteria for computation of the most informative viewpoints (i.e., computation time vs. the amount of processed information). As volume rendering permits to use semi-transparent occlusion, visibility evaluation of each data element (i.e., voxel), comes with high computational costs. While this is acceptable when view selection is evaluated during preprocessing, a per-frame view selection calculation during interactive visualization will significantly drop the performance. Therefore, in such a case, the visibility has to be computed from a representative subset of the volumetric data.

6.1.1 ISOSURFACES

One of the first approaches in view selection for volume data has been designed for fast evaluation of viewpoint quality based on the visibility of extracted iso-surfaces or interval volumes [124]. This approach represents a compromise between surface-based visibility estimation techniques, applied

to polygons, and purely voxel-based visibility estimation. The algorithm first decomposes the entire volume into a set of feature components (i.e., surfaces defined as interval volumes). Then, for each feature component, locally optimal viewpoints are computed using viewpoint entropy for polygonal data (see Equ. 5.3). A globally optimal viewpoint is defined by the highest global viewpoint entropy, which is calculated as the weighted sum of per-feature viewpoint entropies $H_i(v)$ $(i = 1..n)$ for viewpoint v. This importance weighting of feature components can be, for example, the average opacity of voxels in the interval volume specified by the transfer function $\overline{\alpha}_i$. Viewpoint quality considers occlusion among features in the calculation in order to avoid feature overlapping in the case of the best viewpoints. While evaluating view quality on a specific feature, only the unoccluded regions of that interval volume are visible to the view selection calculation.

Definition 6.1. The average opacity-weighted feature-driven viewpoint selection is defined by

$$H(v) = \sum_{i=1}^{n} \frac{\overline{\alpha}_i}{\sum_{j=1}^{n} \overline{\alpha}_j} H_i(v), \tag{6.1}$$

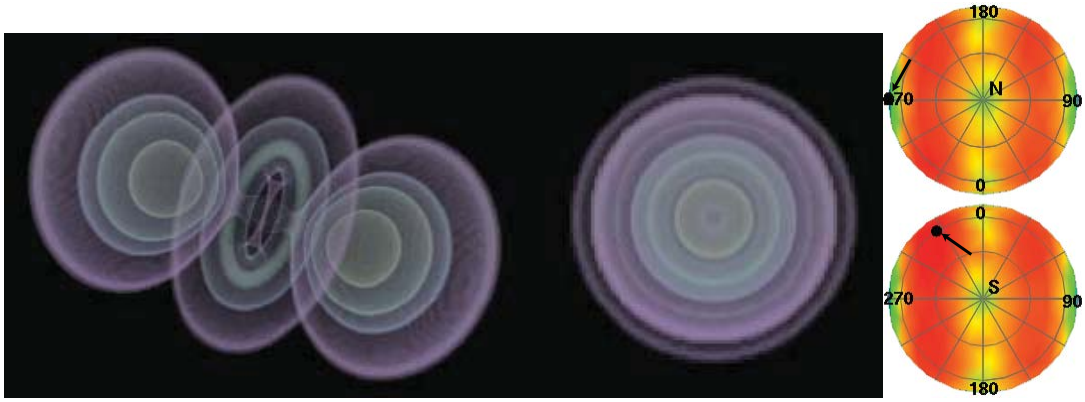
where the sum of viewpoint entropy $H_i(v)$ for feature i and viewpoint v is weighted by importance factor which can be proportional to average density values of a feature $\overline{\alpha}_i$, for example.

This technique performs well during the visibility estimation phase as the computation of polygonal viewpoint entropy is achieving real-time performance. This can be of considerable advantage when performance time is critical. However, the approach does not compute visibilities of the volume but of a set of surfaces representing the volume. Therefore, only a small fraction of the information contained in the volume is taken into account. Volumetric information is integrated into the viewpoint estimation process, through an importance weight computed as the average opacity of the interval volume. Fig. 6.1 shows best and worst views of a set of interval volumes extracted from the hydrogen data set, including the bounding sphere plot indicating areas of varying viewpoint quality.

6.1.2 VOLUMETRIC DATA

Visibility estimation using interval volumes enables fast computation of view quality on volumetric datasets. To compute a precise per-voxel visibility, however, the entire volume has to be considered in the evaluation. Therefore, viewpoint entropy for volumetric data [8] has been adapted from the viewpoint entropy approach for polygonal data [135] (see Equ. 5.3). For the probability distribution function, the area visibility distribution of polygonal faces is replaced by the visual probability $q_i(v)$ of voxel i obtained from the fraction between the voxel visibility $\nu_i(v)$ for a given viewpoint v and the voxel importance W_i . The term visibility ν denotes the transparency of the material between the camera and the voxel and is equal to:

$$\nu_i(v) = \frac{1}{N_r} \sum_{r=1}^{N_r} \prod_{k=1}^{N_s(r)} (1 - tf(s_k).\alpha), \tag{6.2}$$



(*a*) (*b*)

Figure 6.1: Viewpoint estimation for interval volumes: (*a*) best and worst views of interval volumes extracted from a data set containing simulated electron density distribution in a hydrogen atom; (*b*) the sphere plot shows the viewpoint quality distribution where bright region encodes low quality viewpoints and dark region encodes good and informative viewpoints. Worst (top) and best (bottom) views are indicated by black dots. Used with permission. ([124] © IEEE, 2005.)

where N_{r} is the number of rays intersecting the voxel i, k is the sample iterator along the ray r from the first sample s_1, which is entering the volume, to the last sample $s_{N_{\mathrm{s}}(r)}$ before intersecting the voxel i. The $tf().\alpha$ is the opacity transfer function.

The importance distribution can, for example, be defined as the opacity value specified in the transfer function. This means that more opaque voxels will get more prominence than more transparent regions. A more advanced voxel relevance function can incorporate shape characteristics and color in addition to the opacity. The Equ. 6.3 defines the probability distribution function:

$$q_i(v) = \frac{v_i(v)}{W_i} \frac{1}{\sum_{i=1}^{N_{\mathrm{v}}} \frac{v_i(v)}{W_i}}, \tag{6.3}$$

where N_{v} is the number of voxels.

The viewpoint entropy of viewpoint v is then analogous to the polygonal case (Equ. 5.3):

$$H(v) = -\sum_{i=1}^{n} q_i(v) \log q_i(v). \tag{6.4}$$

In addition, this scheme can be extended for the estimation of static viewpoints for time-varying data. This is realized through conditional entropy where the different random variable distributions are obtained from neighboring time-steps considering the time series as the Markov sequence model. The final viewpoint entropy value is a sum of the conditional entropies over the

entire time series:

$$\begin{aligned} H(v) &= H_v(t_1, t_2, \ldots, t_n) \\ &= H(t_1) + H(t_2|t_1) + \ldots + H(t_n|t_1, \ldots, t_{n-1}) \\ &= H(t_1) + H(t_2|t_1) + \ldots + H(t_n|t_{n-1}). \end{aligned} \tag{6.5}$$

View selection for volumes has been extended to support dynamic viewpoint changes for static and time-varying data [60] to enable guided flythrough over the most interesting viewpoints. When the transitional path along the views is defined, the following criteria are considered in the optimization step: (a) camera should move at a near-constant speed, (b) the view direction should not change abruptly, and (c) information perceived from the time-varying data should be maximized.

Having evaluated the viewpoint quality for a particular data type, volumetric or polygonal, it is natural that viewpoints nearby frequently have similar viewpoint qualities. When the visualization goal is to provide a set of representative viewpoints instead of a single best one, a viewpoint clustering scheme is needed. Here again information-theoretic measures for clustering views according to similarity can be used. The viewpoint similarity is computed using the Jensen-Shannon divergence (Equ. 1.16) and according to these values, views either belong to the same cluster, or in case of strong dissimilarity, they each belong to a separate cluster. For all clusters, the most representative viewpoint is selected[1]. Fig. 6.2 shows a tooth data set from several viewpoints that together capture the most information about the scene and a static viewpoint on a more complex time-varying shockwave data set.

6.2 INTEGRATION OF DOMAIN SEMANTICS

Evaluation of viewpoint quality in computer graphics is based on general visibility metrics. In visualization, these metrics are also applicable; however, as the visualization scenario defines what the user wants to visually analyze, the visibility evaluation can be influenced by these domain scenario specifics. In the previous section, the adaptation of different data types, typical for visualization tasks, has been discussed. This section describes how visibility evaluation is extended to serve its purpose of viewpoint quality evaluation for different visualization tasks.

6.2.1 VISUALIZATION OF MOLECULAR STRUCTURES

Optimal viewpoint selection for the visualization of complex molecular structures is a good example of where the viewpoint optimality criterion is a property highly dependent on the visualization task itself. In molecular visualization, a combination of two specific views is interesting in order to efficiently convey information on the molecular structure [137, 138]. One of these two views provides the most information about the molecule atoms as well as the distances and angles of the bonds. In fact, such a condition fulfills the view with the highest entropy value. The second type of view conveys the ordering and spatial arrangement of structures which determines the physical properties

[1] See the framework presented in Chapter 5.

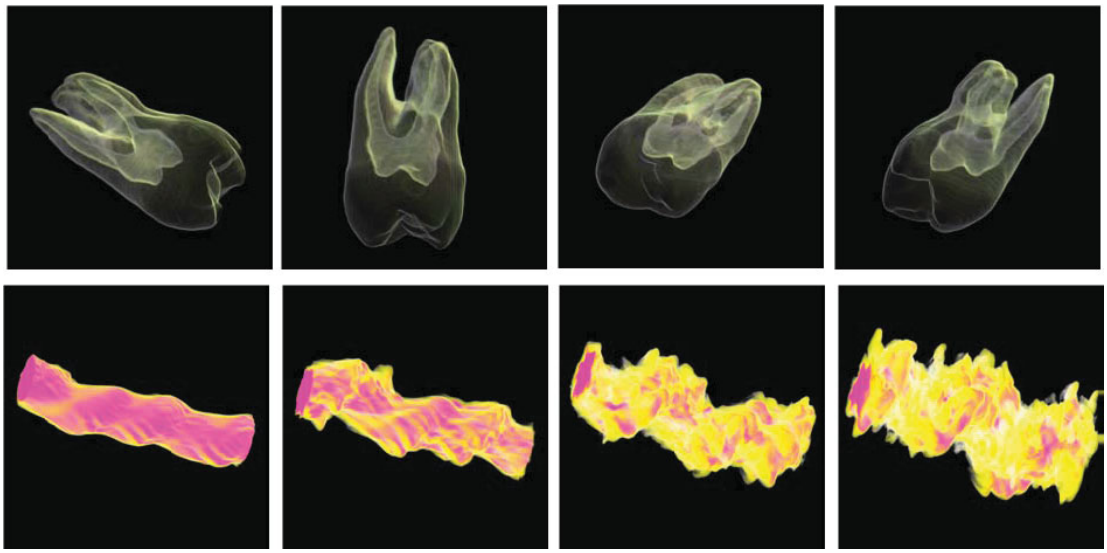


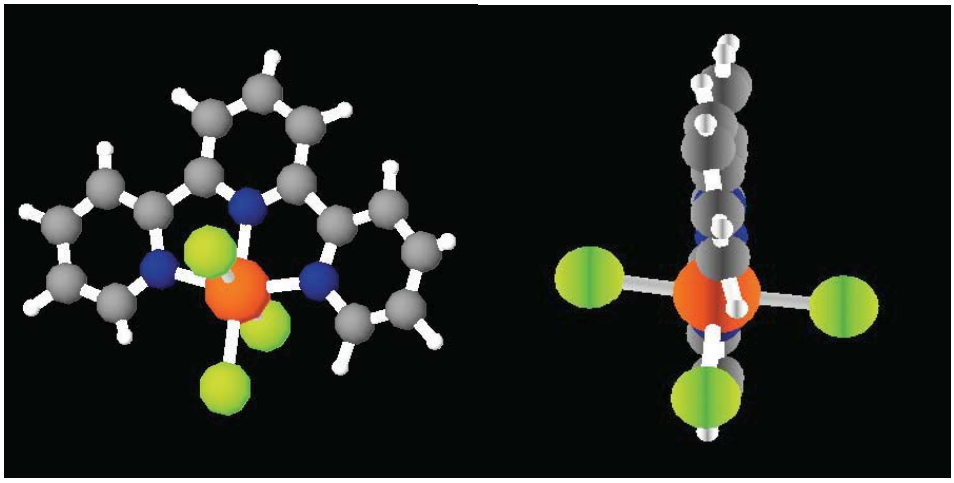
Figure 6.2: View selection for static and time-varying volumes: (top row) four selected views of the tooth data set from four bounding sphere partitions; (bottom row) time-series of the shockwave data set from best temporal domain preserving viewpoint. Used with permission. ([8] © IEEE, 2005.)

of the molecules. This type of interesting view on a molecule has a very low viewpoint entropy. These characteristics are demonstrated in Fig. 6.3 where two different views of the same molecule are shown. The view with the highest viewpoint entropy shows most of the molecular elements, and the lowest viewpoint entropy value conveys the information of the molecular structure, such as symmetry information which is an important property to be communicated for the application domain.

In case of molecular visualization, we want to measure the amount of information provided by a single image, which does not cover all the sphere of directions, but only a small sub-region. The definition of viewpoint entropy (Equ. 5.3) was based on the projected area of polygons on the entire viewing sphere. To consider only the amount of available image space, the visibility computation of molecular elements has been modified so that instead of the projected area on the viewing sphere, the number of pixels of a face N_{p_i} are related to the total count of the pixels N_p for all faces N_f[2]. The frustum entropy is similar to initial viewpoint entropy but is more appearance-based as it measures what can be really seen from one image:

$$H(v) = -\sum_{i=0}^{N_f} \frac{N_{p_i}}{N_p} \log \frac{N_{p_i}}{N_p}, \tag{6.6}$$

[2]For practical reasons, the same implementation has been used to compute the viewpoint quality measures in Chapter 5.



(*a*) (*b*)

Figure 6.3: Images of the same molecule from two different viewpoints: (*a*) viewpoint with the highest viewpoint entropy and (*b*) viewpoint with lowest viewpoint entropy. The combination of these two images gives better information about the molecular structure. ([137] © Eurographics-IEEE, 2002.)

where $i = 0$ refers to background.

6.2.2 GUIDED NAVIGATION IN DATA SEMANTICS

In visualization, frequently, a specific region within the data is of more interest than other surrounding parts. In molecular visualization, the visibility of all structures as well as their structural arrangement was important. Semantics are present as domain-specific guidelines rather than particular chemical or physical properties. No special visual emphasis was given to any part of the molecule as the entire molecule is of viewer's interest.

In volume visualization, a region of interest (ROI) is defined by interpretation and segmentation operations. Voxels are grouped into clusters forming domain objects. These objects are often of high interest to the users and good viewpoints are generally those which clearly depict the object's structure. The visualization task can be, for example, to communicate the data interpretation to another audience. For such a task, viewpoint selection can be utilized to implement high-level interaction request such as: "Show me object *X*". View selection, therefore, should not be based solely on the visibility of graphic primitives (e.g., voxels). To provide the best view of a certain feature, the visibility of these objects has to be computed. The simplest and fastest way to compute the optimal viewpoint of domain objects can be along the lines of the feature-driven viewpoint estimation approach [124].

Importance-driven focus of attention [140] is one example where guided navigation through pre-classified features in a volumetric data set uses automatic viewpoint selection for setting the view

to most informative viewpoints on a given object of interest. The object in focus is directly selected by the user by choosing it from a list of classified objects. A characteristic viewpoint for this object is selected in combination with a visually pleasing discrimination of the focus from the remaining context information. By changing the object of interest, both viewpoint settings and visual parameters are smoothly modified to put emphasis on the newly selected object of interest. The most informative viewpoints of each structure are pre-computed using viewpoint mutual information (Equ. 5.6). This measure was used instead of the more common viewpoint entropy as it has better properties when dealing with view selection for importance-weighted objects within the volumetric data. Viewpoint mutual information acts as information channel between two random variables, that is, viewpoint positions and volume objects. For a specific viewpoint position v and set of objects $\mathcal{O}$, VMI expresses the level of correlation of object visibility with respect to the viewpoint position. The best views of an object are those with the lowest mutual information. This is in contrast to the view evaluation using viewpoint entropy, which is maximal for the best views. The low VMI expresses that the viewpoint change has a small influence on the object visibility, and thus the viewpoint is stable. Similarly, to Equ. 5.21 for polygonal data, viewpoint mutual information $I'(v;\, O)$ for importance-weighted data objects is defined as:

$$I'(v;\, O) = \sum_{o \in \mathcal{O}} p(o|v) \log \frac{p(o|v)}{p'(o)}, \text{ where } p'(o) = \frac{p(o) imp(o)}{\sum_{o \in \mathcal{O}} p(o) imp(o)}, \tag{6.7}$$

where $p(o|v)$ is the data object o visibility from viewpoint position v and $p(o)$ is the marginal probability (i.e., the sum of object visibilities from all viewpoints and $imp(o)$ is the importance of object o). The importance allows flexibly to integrate focusing on object(s) of interest in the view selection computation. One, two, or more objects can be assigned a high importance value. All these objects will then have high visibility in the selected viewpoint.

Finding a viewpoint where the characteristics of a specific feature are clearly visible naturally requires a visibility estimation scheme. Complete visibility information requires ray casting of the whole data set from various viewpoints, similarly, to the view selection for volumes based on viewpoint entropy computed from voxels [8]. The visibility computation is based on the opacity contribution of each voxel, and object visibility is computed as the sum of voxel visibilities corresponding to the object. Additionally, two weights influence the visibility of an object (i.e., image-space weight and object-space weight). Image-space weight penalizes the visibility of objects when they are located outside the image center. Object-space weight assigns higher visibility to objects, which are closer to the viewing plane and penalizes those that are farther away.

The final object visibility is then mapped to a conditional probability of the object for a given viewpoint $p(o|v)$. These values are used for the computation of good viewpoints for a given object by using the viewpoint mutual information weighted with object importance information.

After selecting visual representations of objects and identifying representative viewpoints, the crucial information to perform interactive focus of attention is available. During the guided navigation, the object in focus is assigned a higher importance value. This value is directly mapped to all focusing mechanisms: dense visual style, level of ghosting of cut-away views, and position of

Figure 6.4: Importance-driven focusing: Smooth guided navigation to focus on the object of interest. ([140] © IEEE, 2006.)

the viewpoint. Thus, the viewpoint transformation is also controlled by the importance distribution, and its variation during guided navigation smoothly changes to the most informative viewpoint of the object in focus.

Importance-driven focus of attention guides user's focus to the object of interest, while still permitting interactive viewpoint manipulation. Focusing on a specific feature in the human hand dataset is shown in Fig. 6.4. The viewpoint smoothly changes from the most informative viewpoint for the entire volume to viewpoint emphasizing the object of interest. As shown, parallel to the viewpoint change, the focus is discriminated from neighboring structures by using a different visual style and ghosting is employed to suppress occluding structures.

Guided navigation through semantically-enriched data consisting of anatomical objects and their textual descriptions can be realized in multiple ways. One approach is, as in the case of importance-driven focus of attention, by selecting the textual description of a particular object. However, this link can also be realized in the opposite direction. When a particular view is selected, the textual description is automatically updated to provide textual information on the object with best visibility from the user-specified view [49].

Guidance among diagnostically relevant viewpoints for intervention planning in various medical scenarios [81] is an example of how the basic technology of view selection is applied to address a particular problem in the medical imaging pipeline. The data is first interpreted into anatomical objects and represented as polygonal meshes. The visibility of objects is calculated from these extracted iso-surfaces. Good viewpoints are estimated using many parameters with adjustable influence: object entropy, importance of occluders, size of unoccluded surface, preferred view region by surgeons, distance to viewpoint, and viewpoint stability. The choice of parameters demonstrate as how tightly viewpoint estimation is bound to specific domains: the distance to important feature defines importance of other features (e.g., neck muscles close to lymph node which is in focus). Furthermore, guided navigation supports zooming to the object of interest. Guided navigation of lymph nodes for neck intervention planning is shown in Fig. 6.5.

Sometimes domains utilizing visualization have well-defined standardized (also called canonical) views and clear rules for their identification. Usually these rules are based on semantic information which is enriching the raw data. View selection for molecular visualization was a very data-near

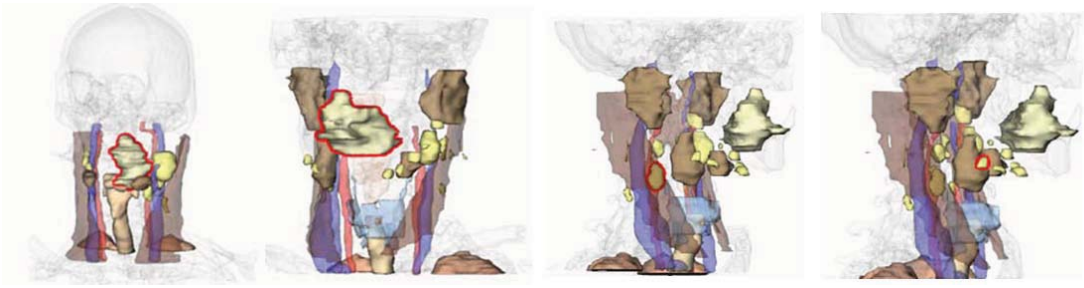


Figure 6.5: Semantics-driven view selection: Guided navigation through features in the human neck data set assists studying the correspondence between focus objects (i.e., lymph nodes and surrounding tissue such as neck muscle). Used with permission. ([81] © Eurographics-IEEE, 2007.)

approach, where rough structural semantics were assigned to viewpoint entropy values, which has steered the view selection process. However, as the optimal viewpoint is still not very well defined in that case, the use of information-theoretic measures is the most effective approach in selection of informative views on the molecule. When this definition would become more domain-specific, the utility of information-theoretic measures will become lower.

The current trend in application-oriented visualization research indicates that knowledge-assisted approaches are a promising direction for optimal viewpoint estimation. This is especially true for domains such as medicine where anatomical structures already have well defined canonical views, which are also related to particular treatment procedures. Approaches, such as viewpoint selection for intervention planning [81] or the LiveSync [67] interaction metaphor, demonstrate this trend in medical visualization. The latter mentioned allows the automatic generation of meaningful 3D views based on the selection of a point in a 2D slice. The view selection has several different criteria of optimality, which are combined as a sum or multiplication of the evaluated weights. However, these techniques are fine-tuned for specific applications and are not easily transferable to other scenarios.

From the state-of-the-art view selection techniques, only the information-theoretic measures are truly application-independent. Their potential in view selection utilization is mainly for the structures in the data, which do not have well-defined canonical views when the structure of studied phenomenon is unknown, and when data has no associated semantics that would express user's interest or domain specifics.

A scenario which matches these characteristics and where the visualization is already employed to provide insights, is visualization of computer simulations. Simulations can, for example, originate from a broad spectrum of disciplines within natural sciences. Very often, they are highly dependent on the choice of input parameters. The user often does not have a clear understanding how these parameters influence development of particular phenomenon over time. Information-theoretic view selection algorithms can be, in such a case, employed to generate fast previews of simulation runs.

These views are selected without a-priori knowledge of what the simulation data sets represent but provide a good guess of what could be interesting for the human observer.

View selection for visualization of simulations is just one example of a promising application. In general, automatic technique for obtaining good views on complex data is a useful tool and should be an integrated part of an entire visualization tool-set provided by various visualization toolkit libraries. In order to keep general applicability of such a tool-set, the view selection approach has to be general, without being tailored to a very specific data or application scenario. In such a case, view selection based on information-theoretic measures is the best way to cover all possible spectrum of scenarios where visualization could be employed.

CHAPTER 7

Viewpoint-based Geometry Simplification

Currently, polygonal models dominate interactive computer graphics. Polygons are the simplest primitive and lead to regular rendering algorithms that fit well in hardware. Unfortunately, the complexity of these models seems to grow faster than the ability of graphics hardware to render them interactively. Polygonal simplification offers one solution to this problem.

Most common polygonal simplification methods use some technique based on a geometric distance as a measure of quality between an original mesh and the one obtained from simplification. With these methods, we can achieve meshes that are very similar to the original. In general, one of the most important advantages of geometry-oriented methods is their low computational cost. This fact makes them suitable for scanned models since these models are composed of thousands or even millions of polygons. In addition, geometric methods are very useful in applications that require exact geometric tolerances with regard to the original model. Examples of such applications include collision detection and path planning for part insertion and removal.

In contrast, image-based simplification methods carry out a simplification guided by differences between images more than by geometric distances. In other words, their goal is to create simplified meshes that appear similar according to visual criteria. These methods present a high computational cost compared to geometric ones. The applications that can benefit from image-based methods are those in which the main requirement is visual similarity. Examples of such applications are video games, vehicle simulations, and walkthroughs.

7.1 BACKGROUND

The most important improvement in geometry-oriented simplification methods in recent years was the incorporation of mesh attributes such as color, normals, and textures. For example, Hoppe extended his initial work [55] by proposing a new quadric metric that includes colors and texture coordinates [56], and the QSlim algorithm [40] was also extended with those attributes [41]. Cohen et al. [18] developed an algorithm based on edge collapses that samples the vertex position, normal, and color attributes of the original mesh and then converts them to normal and texture maps. This algorithm is based on a texture deviation metric. Lately, a general method to incorporate texture information for edge collapse-based simplification algorithms has been proposed in [45].

Lindstrom et al. [74] were the first to address the problem of *visual similarity* by developing a pure image-based metric. Basically, their method determines the cost of an edge collapse operation by rendering the model from several viewpoints. The algorithm compares the rendered images to

the original ones and adds the mean-square error in luminance across all the pixels of all the images. Then, all the edges are sorted by the total error induced in the images, and after that, the edge collapse that produces the least error is chosen. Lindstrom et al. used 20 viewpoints in their implementation to compute the error. The main advantage of this method is that the metric provides a natural way to balance the geometric and shading properties without requiring the user to perform an arbitrary weighting of them. On the other hand, its main disadvantage is the high computational cost.

Karni et al. [63] proposed a metric to capture the visual difference between two approximations, the average of the norm of the geometric distance between models and the norm of the Laplacian difference. By introducing the Laplacian component, some visual properties that the human eye appreciates such as smoothness are better captured.

Luebke et al. [76] presented a method to perform a view-dependent polygonal simplification using perceptual metrics. These metrics derive from a measure of low-level perceptibility of visual stimuli in humans. Later on, Williams et al. [142] extended this work for lit and textured meshes.

Zhang et al. [147] proposed a new algorithm that takes visibility into account. This approach defines a visibility function between the surfaces of a model and a surrounding sphere of cameras. The number of cameras increases both accuracy and calculation time. Zhang et al. used up to 258 cameras. In order to guide the simplification process, they combined their visibility measure with the quadric measure introduced by Garland et al. [40].

Lee et al. [71] introduced the idea of mesh saliency as a measure of regional importance for graphic meshes. This measure was incorporated into mesh simplification. Briefly, this approach consists in generating a saliency map, and then simplifying by using this map in the QSlim algorithm [147]. The new edge collapse cost is that of the quadric multiplied by the saliency of this edge.

7.2 VIEWPOINT-BASED ERROR METRIC

As we have seen in Chapter 5, information-theoretic-based viewpoint selection metrics have been successfully applied in different areas of computer graphics, such as scene understanding, virtual exploration and volume visualization. In this chapter, viewpoint entropy (VE, Equ. 5.3), viewpoint Kullback-Leibler distance (VKL, Equ. 5.7), and viewpoint mutual information (VMI, Equ. 5.6) are used to compute the simplification error.

In this section, a new error metric based on information-theoretic viewpoint measures is presented. This metric can be used to evaluate the cost of a decimation operation. The edge collapse is chosen as the decimation operation, although any other simplification operation could be performed such as removing a vertex, replacing a cluster of vertices by a single one, and contracting an edge.

Viewpoint quality measures quantify the accessible information about an object from a particular viewpoint. For instance, given a viewpoint, we can consider that if the simplification is produced near the silhouette, it will probably change the shape of the object. Therefore, if the goal is to preserve the silhouette of the model, we should try to reduce this change. In addition, in order to preserve the global appearance of the model, several equidistant viewpoints surrounding the model are required,

so that the whole model will be fully covered. This distribution guarantees a uniform simplification. Taking into account these facts, the variation of a viewpoint quality measure for each viewpoint (such as VE, VKL, and VMI) can provide us with an error metric to guide the simplification process:

Definition 7.1. The simplification error deviation for an edge collapse e from all viewpoints $\mathcal{V}$ is defined by

$$C_e = \sum_{v \in \mathcal{V}} |I_v - I'_v|, \tag{7.1}$$

where I_v represents the viewpoint quality measure before the edge collapse e and I'_v afterwards. Clearly, the error associated with the simplification of an edge depends largely on the measurement used. The first work following this idea and uses the entropy to guide the simplification process [12]. Subsequently, there are studies on other measures such as mutual information [14] and f-divergences [13]. In this later work, there is an analysis about the following f-divergences: Kullback-Leibler, Hellinger, and Chi-Square.

7.2.1 ANALYSIS

The viewpoint quality measures mentioned in the previous section are based on the distribution of areas of polygons seen from a viewpoint. The area of the background is also included as the polygon number 0 in VE and VMI. This fact allows VE and VMI to better preserve the silhouette. But maybe the main implication of considering the projected areas in VE and VMI is that the hidden geometry will be initially removed, because if a polygon is not seen from any point of view, its simplification will not introduce error. In the following analysis, we focus our attention on the behavior of VMI.

Fig. 7.1 shows the original test model and how the viewpoints are distributed around it. These viewpoints are associated with the vertices of the cube in which the object is inscribed. Fig. 7.2 shows the mesh from four viewpoints around the cube. Only 4 viewpoints are shown because the rest are symmetric. As it can be seen, the different viewpoints have the same VMI ($I(v; Z)$=0.004097 where $v = \{1, .., 8\}$). This is because the object is equally seen from each viewpoint. In more complex models, every viewpoint will usually have a different VMI.

Fig. 7.3 and Fig. 7.4 illustrate how VMI can be employed to conduct the simplification. Fig. 7.3 shows the test model after performing the best edge collapse e and Fig. 7.4 after performing the worst edge collapse e'. The best edge collapse **corresponds** to the lowest simplification error C_e (Equ. 7.1) and the worst to the highest. In the examples shown in Fig. 7.3.*a* and Fig. 7.4.*a*, all VMI values decreased after an edge collapse with respect to Fig. 7.2.*a*. In these cases, the visible area did not increase, but in a more general case, it is possible that after an edge collapse some previously hidden parts of the mesh may now appear, thus increasing the visible area. If we pay attention, for instance, to Fig. 7.2.*b* and compare this same viewpoint after the best edge collapse (see Fig. 7.3.*b*), it can be appreciated that although the number of triangles N_T is reduced to 8, the visible area remains the same. The simplification error for this viewpoint using VMI is C_e=0.004097-0.003651=0.000446. If we analyze the same viewpoint in the worst edge collapse operation (see Fig. 7.4.*b*), it can be

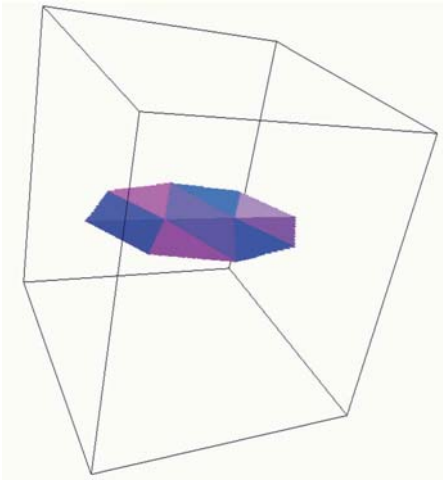


Figure 7.1: Model example consisting of ten triangles. The vertices of the cube represent the position of eight cameras pointing toward the center of the figure. These cameras allow us to completely view the object and are used to demonstrate the behaviour of the proposed measures. ([14] © Elsevier Ltd, 2009.)

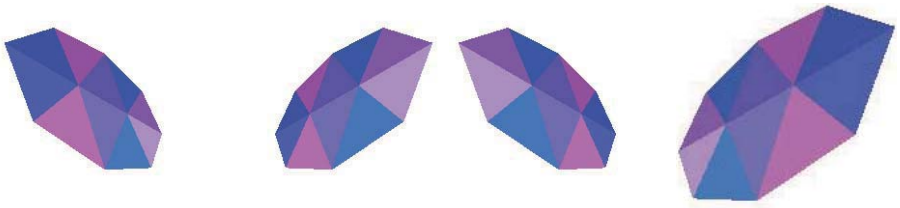


Figure 7.2: Four views of the test model in Fig. 7.1 with $I(v;\, Z)$=0.004097, where $v \in \{1, .., 8\}$. Number of triangles $N_{\mathrm{T}} = 10$. ([14] © Elsevier Ltd, 2009.)

seen that although the number of triangles is less reduced, the total visible area has decreased. The simplification error for this viewpoint is $C_{e'}$=0.004097-0.003372=0.000725, which is higher than the error committed in the best edge collapse.

If we consider the VKL, due to the fact that it uses the actual area of polygons, after an edge collapse, normally one or two polygons will be removed, thus decreasing the total actual area. This will change the value for VKL after an edge collapse. Therefore, the error committed will be distinct from zero. The consequence is that even hidden polygons will have error when simplifying and will

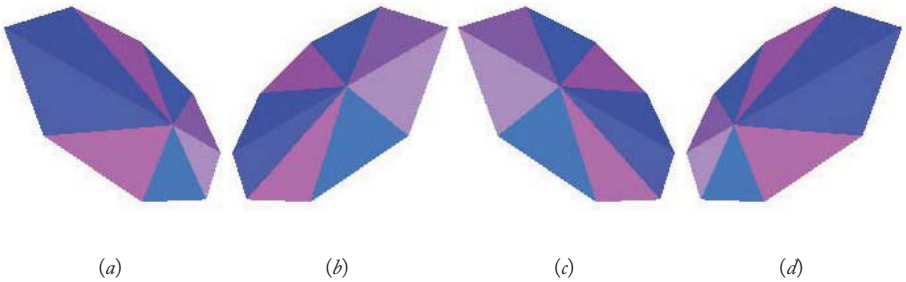


Figure 7.3: Four views of the test model in Fig. 7.1 after performing the best edge collapse e using VMI. $N_T = 8$ and $C_e = 0.002573$. ([14] © Elsevier Ltd, 2009.)

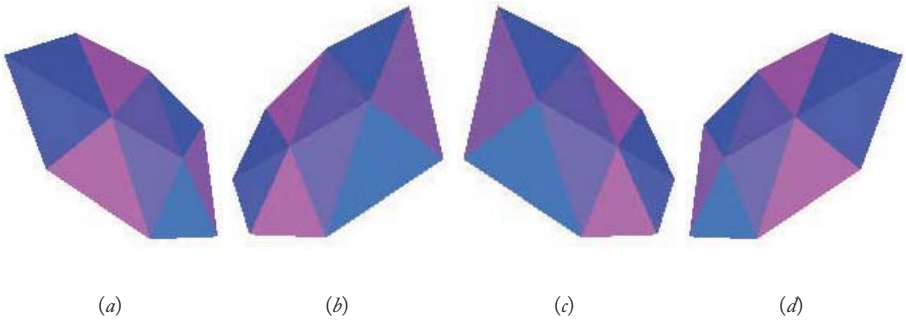


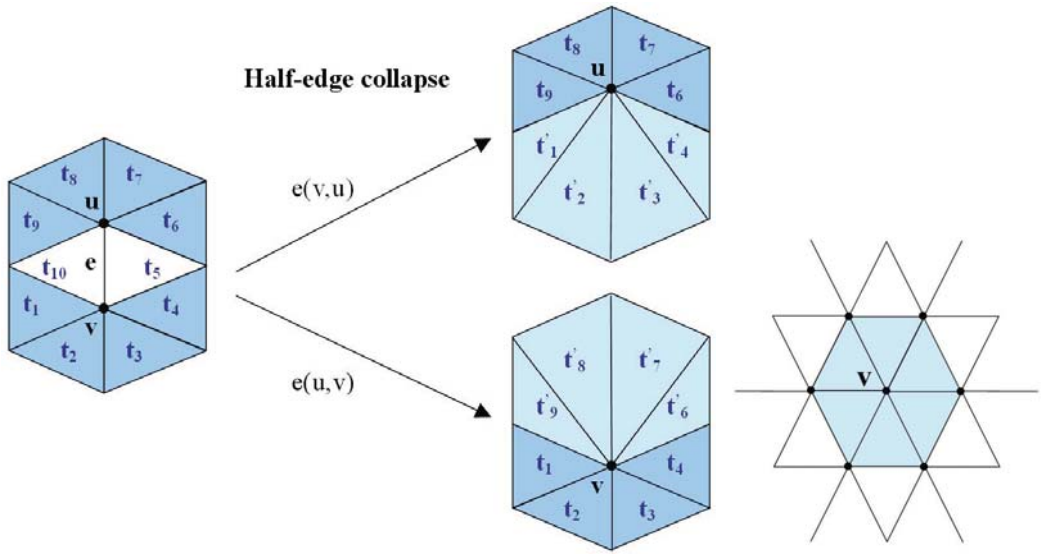
Figure 7.4: Four views of the test model in Fig. 7.1 after performing the worst edge collapse e' using VMI. $N_T = 9$ and $C_{e'} = 0.006228$. ([14] © Elsevier Ltd, 2009.)

not be completely removed during the initial steps of the algorithm. As hidden polygons will be removed according to their actual area, the smallest polygons will be simplified first, preserving the main features of the object in its internal parts.

7.3 SIMPLIFICATION ALGORITHM

The simplification process we use, like many other simplification algorithms, is based on the edge collapse operation. However, we use the *half-edge collapse* operation. According to this, the remaining vertex for an edge collapse $e(u, v)$ is vertex u or v (see Fig. 7.5.*a*). By using half-edge collapses, it is possible to reuse the simplification process in order to generate multiresolution models. These models can use the current hardware in a more efficient way because no new vertices are added to the original model. Furthermore, the half-edge representation is useful for progressive transmission. The main disadvantage is a slight loss of quality of the final mesh, although the complexity of the

simplification algorithm is reduced because we do not have to compute the position of the new vertex v' resulting from the edge collapse. In any case, the general edge collapse operation can be applied to the algorithm. However, a strategy is required to compute the position of the resulting vertex.



(*a*) The half-edge collapse operation

(*b*) Edges adjacent to vertices adjacent to vertex v.

Figure 7.5: In this example edge e is collapsed into vertex u (see $e(v, u)$), but it is also collapsed into v (see $e(u, v)$). The triangles t_{10} and t_5 are removed. ([14] © Elsevier Ltd, 2009.)

Brute force selection of edges can introduce mesh inconsistencies. In order to avoid these artifacts, we only take into account the edges which have at most two adjacent polygons, that is 2-manifold edges. And we also consider boundary edges, that is, edges which have one single adjacent polygon.

The best half-edge collapse is the decimation operation chosen in the algorithm. Note that the cost of collapsing vertex u to v may be different from the cost of collapsing v to u. In this strategy, in order to determine the best orientation of an edge collapse, we would have to render the two possibilities and compute that error. To avoid this computational cost, we used the approach developed by Melax [78] that takes into account polygon normals. Within this approach, the two orientations $e(u, v)$ and $e(v, u)$ are calculated and finally the orientation that produces a minor change in the curvature of the local region around the edge collapse is applied. Hence, the simplification error deviation is only computed for that orientation. In Fig. 7.6, we show the simplification algorithm.

At each iteration, the edge cost has to be evaluated for the entire set of remaining edges. An edge collapse in our algorithm could, in principle, affect the cost of any remaining edge. But this does not always happen to each edge. Thus, at each step, we only choose a small group of edges that are affected by an edge collapse, and then the cost is recalculated for these edges. These edges are the ones that are adjacent to the vertices that, in turn, are adjacent to the vertex v resulting from a half-edge collapse (see Fig. 7.5.*b*).

```
// Compute initial viewpoint selection
// measure for mesh M
Compute I_v, where v = {1, .., n}

// Build initial heap of edge collapses
for(e ∈ M )
   Choose the best orientation of e
   Perform collapse e
   Compute I_v, where v = {1, .., n}
   Compute collapse cost C_e = Σ_{v=1}^{n} |I_v − I'_v|
   Insert the tuple (e, C_e) in heap h
   Undo collapse e
end for

// Update mesh M
while (heap h not empty)
   Delete from heap h edge e with lowest C_e
   Perform collapse e
   Recalculate cost for the neighborhood of e
   and update their location in heap h
end while
```

Figure 7.6: Pseudo-code of the viewpoint-based simplification algorithm.

7.4 EXPERIMENTS

We show some tests with low complexity models from CAD programs. All models were simplified on a Pentium Xeon 2GHz with 1Gb RAM and an NVIDIA 7800-GTX 512Mb graphics card from 20 viewpoints. The results obtained with the viewpoint-driven simplification method were compared to the results with QSlim v2.0 [40], using the best half-edge collapse, at the same level of simplification. We chose QSlim because it is a well-known pure geometry-based algorithm, freely available, and produces high quality simplifications. The images shown were obtained using different viewpoints from those used during the simplification process.

We have implemented the root mean square error (RMSE) of the pixel-to-pixel image difference defined in [74] to measure the mean visual error between the original and the simplified model. This error was taken using 24 viewpoints and 512×512 resolution images. We must emphasize that each viewpoint was different from the one used during the simplification and the resolution was higher.

We measured the geometric error using the mesh comparison tool called *Metro* v4.06 [17]. This tool measures the Hausdorff distance between two meshes.

7.4.1 VIEWPOINT ENTROPY

Table 7.1: Errors measured for the models shown in Fig. 7.7 using VE.

Model	Triangles		RMSE		Metro	
	Original	Final	VE	QSlim	VE	QSlim
Fish	815	100	11.40	22.83	0.05	0.09
Galleon	4,698	500	17.74	36.84	0.11	0.22
Octopus	8,468	500	17.35	25.84	0.09	0.16
Unicycle	13,810	1,000	10.32	11.06	0.03	0.07

Table 7.1 presents the error committed in the experiments using VE. It analyzes the visual error and the geometric error. The obtained results are clearly better than with the geometric method QSlim. For example, the geometric error committed in the Galleon and Unicycle models using VE is 50% less than with QSlim.

Fig. 7.7 shows the results for the analyzed models. The VE achieves much better simplification than QSlim. For example, in the Fish model the tail and the mouth shape is kept better, and in the Galleon model, the same can be said for the sails and the masts.

Fig. 7.8 shows how VE acts at several degrees of simplification for the Galleon model. We have measured the RMSE and the geometric error. As shown in Fig. 7.8.*a*, if the level of simplification is increased, the difference between VE and QSlim becomes larger and the visual quality of VE is even much higher. The geometric error of VE is also lower than QSlim, except during the very first stages as can be observed in Fig. 7.8.*b*. This could be accounted for by the fact that VE is a global measure, and it is possible that, at these stages, QSlim could often be better because it evaluates the error locally.

7.4.2 VIEWPOINT MUTUAL INFORMATION

Table 7.2: Results for VE and VMI measuring visual error (RMSE) and simplification time in seconds. Models ares shown in Fig. 7.10.

Model	Triangles		RMSE		Time	
	Original	Final	VE	VMI	VE	VMI
Shark	734	80	14.78	14.65	10.24	10.23
Galo	6,592	500	9.05	8.38	141.75	142.24
Greekship	9,510	600	13.37	12.85	241.78	246.72
Elephant	31,548	900	13.75	11.60	2,197.67	2,309.79

(*a*) Original Fish model $N_T = 815$ (*b*) VE $N_T = 100$ (*c*) QSlim $N_T = 100$

(*d*) Original Galleon model $N_T = 4,698$ (*e*) VE $N_T = 500$ (*f*) QSlim $N_T = 500$

(*g*) Original Octopus model $N_T = 8,468$ (*h*) VE $N_T = 500$ (*i*) QSlim $N_T = 500$

(*j*) Original Unicycle model $N_T = 13,810$ (*k*) VE $N_T = 1,810$ (*l*) QSlim $N_T = 1,810$

Figure 7.7: Comparative results using VE and QSlim. The corresponding errors can be seen in Table 7.1.

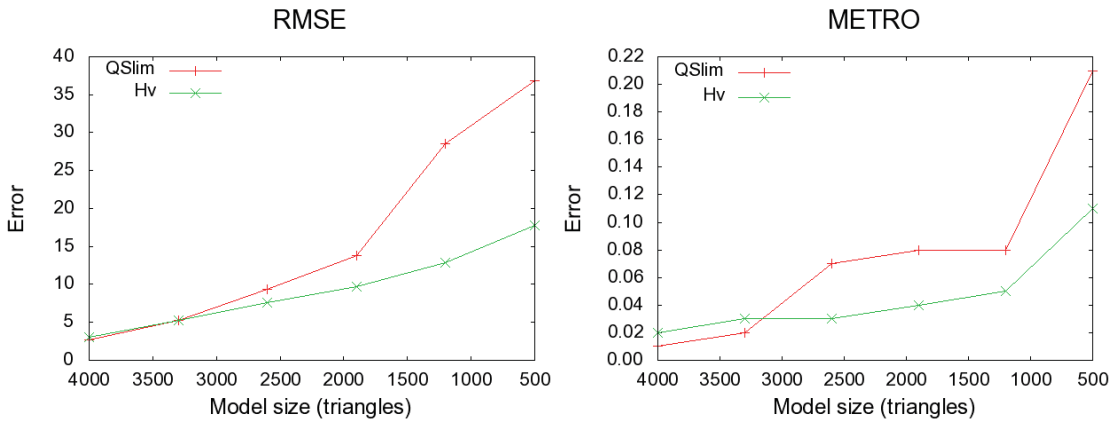


Figure 7.8: Errors measured for the Galleon model at different levels of simplification using QSlim and VE.

First of all, in this section, we perform a comparison between VE and VMI for the models in Fig. 7.10; the results of which appear in Table 7.2. As shown in this table, the visual error is lower in VMI and the computational cost is a bit lower in VE. This difference lies in the calculation performed to obtain the mean projected area of the polygons, that is not necessary in VE.

Table 7.3: Results for QSlim and VMI measuring visual error (RMSE), geometric error (Metro) and simplification time in seconds. Models are shown in Fig. 7.10.

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	10.23
Galo	6,592	500	12.40	8.38	0.05	0.01	0.08	142.24
Greekship	9,510	600	17.20	12.85	0.21	0.09	0.11	246.72
Elephant	31,548	900	25.32	11.60	0.08	0.03	0.52	2,309.79

Table 7.3 depicts the visual and geometric error for QSlim and VMI, and the simplification time. Clearly, the visual error committed with VMI is quite low compared to QSlim, and can even be 50% lower, as shown in the case of the Shark and the Elephant models.

The obtained results for the geometric error (Metro) with VMI are clearly better than with QSlim. For example, the geometric error committed in the Galo and Greekship models using VMI is 50% less than with QSlim and even 75% less in the Shark model. However, it is possible that in particular models this error will be slightly higher than with QSlim. The reason for this is because the

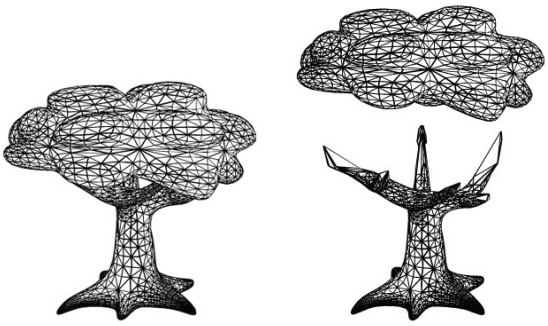


Figure 7.9: VMI operating at the first stages of simplification ($N_T = 7,400$) for the Simpletree model. VMI is able to remove all the hidden interiors.

model has lots of hidden interiors and these are completely removed, thus increasing the geometric error but not the visual one.

Results in Table 7.3 also allow us to analyze the computational cost. This cost is proportional to the complexity of the model and the final number of triangles demanded. However, the QSlim algorithm is extremely fast. Its times for these models are less than one second. However, despite the high computational cost, VMI produces high quality simplifications according to visual similarity.

Fig. 7.10 shows the results for different models. For the Shark model VMI achieves much better simplification than QSlim. The fins, the head, and the tail are kept better in VMI than QSlim. The tail and the crest of the Galo models are kept better in VMI. The oars of the greekship model are kept much better in VMI, as well as the rope that comes from the mast, while in QSlim the rope is removed completely. About the Elephant model, VMI keeps the shape of the ears and the tusks much better than QSlim. In conclusion, VMI attains a better simplification than the geometric method QSlim. The difference between VMI and QSlim is bigger if the model presents lots of hidden interiors, in which case VMI can accomplish much better simplifications.

Fig. 7.9 shows how VMI works at very early simplification levels. In this case, we analyze the Simpletree model since it presents hidden interiors in the branches which are in contact with the tree top. The Simpletree model was simplified at around 66% ($N_T = 7,400$). As shown in this figure, VMI accomplishes a great level of simplification in this region. At this level, most of the simplifications focused on hidden interiors.

7.4.3 VIEWPOINT KULLBACK-LEIBLER DISTANCE

Table 7.4 depicts the visual and geometric error using VKL compared with VE and QSlim. Clearly, the visual error committed with VKL is quite low compared to QSlim, and can even be 50% lower, as shown in case of the Fish and the Elephant model. However, the visual error is slightly improved in VE. This is due to the fact that VE removes completely the hidden interiors and non-visible regions of the model increasing the visual quality.

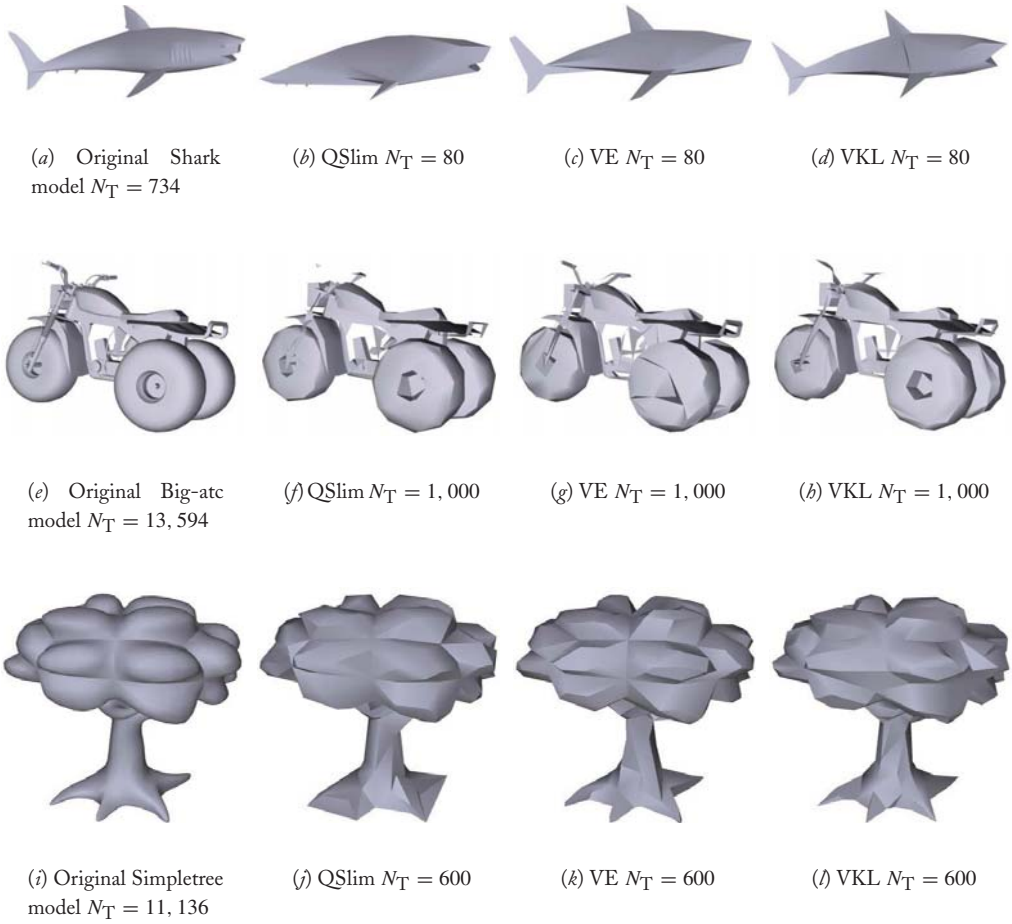
(*a*) Original Shark model $N_T = 734$ (*b*) QSlim $N_T = 80$ (*c*) VMI $N_T = 80$

(*d*) Original Galo model $N_T = 6,592$ (*e*) QSlim $N_T = 500$ (*f*) VMI $N_T = 500$

(*g*) Original Greek-ship model $N_T = 9,510$ (*h*) QSlim $N_T = 600$ (*i*) VMI $N_T = 600$

(*j*) Original Elephant model $N_T = 31,548$ (*k*) QSlim $N_T = 900$ (*l*) VMI $N_T = 900$

Figure 7.10: Comparative results using VMI and QSlim. The corresponding errors and times can be seen in Table 7.3. ([14] © Elsevier Ltd, 2009.)



(*a*) Original Shark model $N_T = 734$ (*b*) QSlim $N_T = 80$ (*c*) VE $N_T = 80$ (*d*) VKL $N_T = 80$

(*e*) Original Big-atc model $N_T = 13,594$ (*f*) QSlim $N_T = 1,000$ (*g*) VE $N_T = 1,000$ (*h*) VKL $N_T = 1,000$

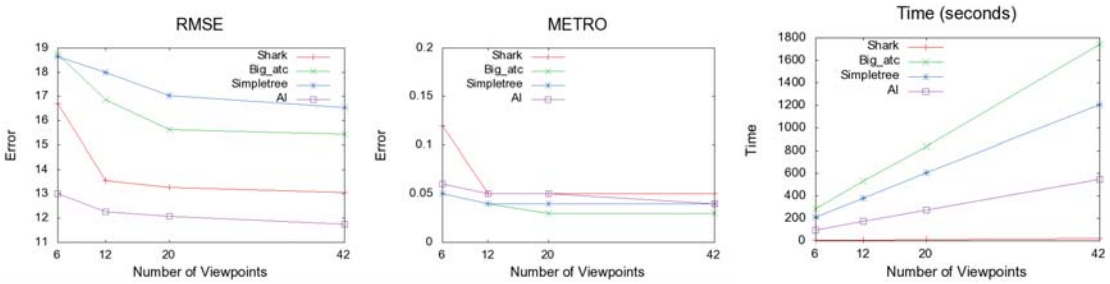
(*i*) Original Simpletree model $N_T = 11,136$ (*j*) QSlim $N_T = 600$ (*k*) VE $N_T = 600$ (*l*) VKL $N_T = 600$

Figure 7.11: Comparative results for QSlim, VE, and VKL. First column shows the original model, second column the model simplified with QSlim, third column with VE, and fourth column with VKL. ([14] © Elsevier Ltd, 2009.)

The geometric error for VKL is better in all the models than for the QSlim method and VE. For example, the geometric error committed in the Galo, the Simpletree, and the Big_atc models using VKL is at least 75% less than with QSlim. Fig. 7.12 shows the Al Capone model rendered with transparency. This model has lots of hidden interiors. For instance, the hidden joints of the arms and the hip are preserved better with VKL (see Fig. 7.12.*d*) whereas with VE (see Fig. 7.12.*c*) these interpenetrating parts are partially removed. Therefore, the approximations produced with VKL preserve the interior regions better than VE and, consequently, present a lower geometric error.

(*a*) Original model (*b*) QSlim $N_T = 1,000$ (*c*) VE $N_T = 1,000$ (*d*) VKL $N_T = 1,000$

Figure 7.12: The Al Capone model rendered with transparency. Image (*a*) shows the original model, (*b*) the model simplified with QSlim, (*c*) with VE, and (*d*) with VKL.



(*a*) (*b*) (*c*)

Figure 7.13: Errors and times measured for some models simplified with VKL using different number of viewpoints.

As shown in Table 7.5, the computational cost is proportional to the complexity of the model and the desired number of triangles. As we have seen previously, the QSlim algorithm is extremely fast. The reader can observe also that VE is faster than VKL due to the fact that VE does not have to compute the actual areas for the polygons, and its initial value can be updated at every simplification step because the total projected area is the image resolution. VKL considers the total actual area and this area changes at every simplification. Accordingly, VKL must be recomputed. In conclusion, VKL improves the geometric error without decreasing significantly the visual quality and can be useful in many applications.

Table 7.4: Errors for some models using QSlim, VKL and VE.

Model	Triangles		RMSE			Metro		
	Original	Final	QSlim	VE	VKL	QSlim	VE	VKL
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	1000	17.66	11.47	12.07	0.03	0.08	0.03
Simpletree	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

Table 7.5: Simplification time (seconds) for some models using QSlim, VE and VKL.

Model	Triangles		Time		
	Original	Final	QSlim	VE	VKL
Fish	815	100	0.03	10.01	11.31
Galo	6,592	500	0.08	141.75	237.30
Al Capone	7,124	1000	0.08	150.90	273.18
Simpletree	11,136	600	0.20	332.49	605.49
Big_atc	13,594	1000	0.27	535.23	835.88
Elephant	31,548	900	0.52	2,197.67	4,016.78

Figs. 7.11.*a-d* show the results for the Shark model. The fins, the head, and the tail are kept better in VKL and VE than in QSlim. Figs. 7.11.*i-l* show the Simpletree model. The roots at the base of the trunk are kept much better in VKL and VE and even both are able to keep the tree top better than QSlim. To sum up, VKL and VE attain better simplification than the quadric method QSlim. The difference between VKL and VE is noticeable if the model presents hidden interiors. In such a case, VE can accomplish a slightly better visual simplifications at the expense of increasing the geometric error since it may remove completely those hidden interiors (see Fig. 7.12).

Some experiments with more viewpoints for some of the test models are shown in Fig. 7.13. The different viewpoint configurations correspond to the vertices of Platonic solids. This guarantees that the viewpoints are distributed uniformly. The last configuration (42 viewpoints) was obtained by subdivision. The visual and the geometric error is analyzed in this figure together with the computational cost. Both visual and geometric errors hardly improve when the number of cameras increases from 20 to 42. Nevertheless, the computational cost is about twice as high (see Fig. 7.13.*c*). Therefore, we believe that 20 viewpoints are a good compromise between quality and efficiency. The two other measures tested in this section (VE and VMI) display the same behavior when different number of viewpoints are considered.

Summary

We have presented in this book several applications of Information Theory (IT) to Computer Graphics. The applications fall broadly into two categories: the mapping of the problem to an information channel, as in radiosity and viewpoint applications, and the direct use of measures as entropy, Kullback-Leibler distance, Jensen-Shannon divergence, and f-divergences, to evaluate the homogeneity of a set of samples and build the corresponding oracle. Sometimes both approaches are used in combination. Other interesting applications come from the hierarchical properties of the entropy and the information bottleneck method. They allow us to progressively extract information from a hierarchical structure.

Do the presented applications exhaust the catalog of prospective utility in computer graphics? Obviously not. As explained above, whenever we can map a problem to an information channel, we can use the nice tools provided by IT to study the channel. Whenever there is a question of homogeneity, upon which an oracle depends, IT measures can be tried. A question arises: why IT tools and measures should work well? The answer is fundamentally simple: we are measuring information, be it geometrical or physical, and IT tools are specially designed to measure information. The decomposition property of the entropy, which was one of its basic required properties for its definition, captures, for instance, the scalability of information along a tree. A nice result of using these tools is that unforeseen meaningful relationships and quantities can be obtained. Take, for instance, the channel paradigm applied to viewpoint selection. Although the mapping was first thought to be between viewpoints and polygons in an object, by reversing the channel new properties were found.

Some future applications obviously come to mind, as they are a direct translation from instance of pixels to voxels. Take the simplification of a polygonal model in Chapter seven. This can be ported to simplification of a volume, in the same way that the information channel in Chapter five can be ported to the information channel for volume visualization in Chapter six. Other applications are perhaps not so obvious. Following with the volume visualization example, consider the information contained in a volume dataset. To visualize it, we need to fix a transfer function. There are infinite possibilities, from which many of them will be redundant, and others will not give any meaningful information at all. If the problem is presented as the extraction of information from a dataset, how can we then use the IT tools at hand to obtain the best informative transfer functions in an automatic way? Of course, for a well established context in volume rendering, such as some medical image datasets, there is already a set of known more informative data transfer functions. But automatic selection could be useful for volume datasets for which we do not have a priori information.

Bibliography

[1] M. S. Ali and S. D. Silvey. A general class of coefficient of divergence of one distribution from another. *Journal of Royal Statistical Society (Serie B)*, 28(1):131–142, 1966.

[2] C. Andújar, P. P. Vázquez, and M. Fairén. Way-finder: Guided tours through complex walkthrough models. *Computer Graphics Forum*, 23(3):499–508, 2004. DOI: 10.1111/j.1467-8659.2004.00781.x

[3] R. Badii and A. Politi. *Complexity. Hierarchical Structures and Scaling in Physics*. Cambridge University Press, 1997.

[4] P. Bekaert. *Hierarchical and Stochastic Algorithms for Radiosity*. PhD thesis, Katholieke Universiteit Leuven, Leuven, Belgium, December 1999.

[5] P. Bekaert, L. Neumann, A. Neumann, M. Sbert, and Y. D. Willems. Hierarchical Monte Carlo radiosity. In G. Drettakis and N. Max, editors, *Rendering Techniques '98 (Proceedings of the 9th Eurographics Workshop on Rendering)*, pages 259–268, New York (NY), USA, June 1998. Springer-Verlag.

[6] V. Blanz, M. Tarr, and H. Bülthoff. What object attributes determine canonical views? *Perception*, 28:575–599, 1999. DOI: 10.1068/p2897

[7] M. R. Bolin and G. W. Meyer. A perceptually based adaptive sampling algorithm. In M. Cohen, editor, *SIGGRAPH '98 Conference Proceedings*, Annual Conference Series, pages 299–309, New York (NY), USA, July 1998. ACM SIGGRAPH, ACM Press.

[8] U. D. Bordoloi and H.-W. Shen. Viewpoint evaluation for volume rendering. In *IEEE Visualization 2005*, pages 487–494, 2005. DOI: 10.1109/VISUAL.2005.1532833

[9] H. Bülthoff, S. Edelman, and M. Tarr. How are three-dimensional objects represented in the brain? *Cerebral Cortex*, 5:247–260, 1995. DOI: 10.1093/cercor/5.3.247

[10] J. Burbea and C. R. Rao. On the convexity of some divergence measures based on entropy functions. *IEEE Transactions on Information Theory*, 28(3):489–495, May 1982. DOI: 10.1109/TIT.1982.1056497

[11] T. M. Caelli. *Visual Perception: Theory and Practice*. Pergamon Press, Oxford, UK, 1981.

[12] P. Castelló, M. Sbert, M. Chover, and M. Feixas. Viewpoint entropy-driven simplification. In *Proceedings of WSCG 2007*, pages 249–256, 2007.

[13] P. Castelló, M. Sbert, M. Chover, and M. Feixas. Viewpoint-based simplification using f-divergences. *Information Sciences*, 178(11):2375–2388, 2008. DOI: 10.1016/j.ins.2008.01.011

[14] P. Castelló, M. Sbert, M. Chover, and M. Feixas. Viewpoint-driven simplification using mutual information. *Computers & Graphics*, 32(3):451–463, 2008. DOI: 10.1016/j.cag.2008.05.005

[15] F. Cazals and M. Sbert. Some integral geometry tools to estimate the complexity of 3D scenes. Research Report 3204, INRIA, July 1997.

[16] P. Christensen. Ambient occlusion, image-based illumination and global illumination. *Photorealistic RenderMan Application Notes, Note 35*, 2002.

[17] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: Measuring error on simplified surfaces. *Computer Graphics Forum*, 17(2):167–174, 1998. DOI: 10.1111/1467-8659.00236

[18] J. Cohen, M. Olano, and D. Manocha. Appearance-preserving simplification. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 115–122, New York, NY, USA, 1998. ACM Press. DOI: 10.1145/280814.280832

[19] M. F. Cohen and D. P. Greenberg. The hemi-cube: A radiosity solution for complex environments. *Computer Graphics (Proceedings of SIGGRAPH '85)*, 19(3):31–40, July 1985. DOI: 10.1145/325165.325171

[20] M. F. Cohen and J. R. Wallace. *Radiosity and Realistic Image Synthesis*. Academic Press Professional, Boston (MA), USA, 1993.

[21] R. Coleman. *Stochastic Processes*. George Allen & Unwin Ltd., London, UK, 1974.

[22] I. E. Commission. Default RGB colour space - sRGB. In *Colour Management in Multimedia Systems*, chapter Part 2.1. IEC, Geneva, Switzerland, 1998.

[23] Computer Graphics Research Group. Renderpark: A photorealistic rendering tool. Software, Department of Computer Science, Katholieke Universiteit Leuven, Leuven, Belgium, 2000.

[24] R. L. Cook, T. Porter, and L. Carpenter. Distributed ray tracing. *Computer Graphics (Proceedings of SIGGRAPH '84)*, 18(3):137–145, July 1984. DOI: 10.1145/964965.808590

[25] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley Series in Telecommunications, 1991. DOI: 10.1002/0471200611

[26] I. Csiszár. Eine Informationsheoretische Ungleichung und ihre Anwendungen auf den Beweis der Ergodizität von Markoffschen Ketten[1]. *Magyar Tudományos Akadémia Közleményei*, 8:85–108, 1963.

[1] An information-theoretic inequality and its application to the proof of the ergodicity of Markov chains.

[27] I. Csiszár and P. C. Shields. Information theory and statistics: A tutorial. *Foundations and Trends in Communications and Information Theory*, 1(4), 2004. DOI: 10.1561/0100000004

[28] L. Darsa and B. Costa. Multi-resolution representation and reconstruction of adaptively sampled images. In *Proceedings of IX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI '96)*, pages 321–328, October 1996.

[29] L. Darsa, B. Costa, and A. Varshney. Navigating static environments using image-space simplification and morphing. In M. Cohen and D. Zeltzer, editors, *1997 Symposium on Interactive 3D Graphics*, pages 25–34. ACM SIGGRAPH, April 1997. DOI: 10.1145/253284.253298

[30] M. A. Z. Dippé and E. H. Wold. Antialiasing through stochastic sampling. *Computer Graphics (Proceedings of SIGGRAPH '85)*, 19(3):69–78, July 1985. DOI: 10.1145/325165.325182

[31] D. M. Endres and J. E. Schindelin. A new metric for probability distributions. *IEEE Transactions on Information Theory*, 49(7):1858–1860, 2003. DOI: 10.1109/TIT.2003.813506

[32] M. Feixas. *An Information-Theory Framework for the Study of the Complexity of Visibility and Radiosity in a Scene*. PhD thesis, Universitat Politècnica de Catalunya, Barcelona, Spain, Desember 2002.

[33] M. Feixas, E. del Acebo, and M. Sbert. Entropy of scene visibility. In *Proceedings of Winter School on Computer Graphics and CAD Systems (WSCG '99)*, pages 25–34, Plzen-Bory, Czech Republic, February 1999.

[34] M. Feixas, M. Sbert, and F. González. A unified information-theoretic framework for viewpoint selection and mesh saliency. *ACM Transactions on Applied Perception*, 6(1):1–23, 2009. DOI: 10.1145/1462055.1462056

[35] D. P. Feldman. A brief introduction to: Information theory, excess entropy and computational mechanics, 2002.

[36] D. P. Feldman and J. P. Crutchfield. Discovering noncritical organization: Statistical mechanical, information theoretic and computational views of patterns in one-dimensional spin systems. Working Paper 98–04–026, Santa Fe Institute, Santa Fe (NM), USA, April 1998.

[37] D. P. Feldman and J. P. Crutchfield. Statistical measures of complexity: Why? *Physics Letters A*, 238(4/5):244–252, 1998. DOI: 10.1016/S0375-9601(97)00855-4

[38] J. Feldman and M. Singh. Information along contours and object boundaries. *Psychological Review*, 112(1):243–252, January 2005. DOI: 10.1037/0033-295X.112.1.243

[39] R. Gal and D. Cohen-Or. Salient geometric features for partial shape matching and similarity. *ACM Transactions on Graphics*, 25(1):130–150, 2006. DOI: 10.1145/1122501.1122507

[40] M. Garland and P. Heckbert. Surface simplification using quadric error metrics. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 209–216. ACM Press/Addison-Wesley Publishing Co., 1997. DOI: 10.1145/258734.258849

[41] M. Garland and P. S. Heckbert. Simplifying surfaces with color and texture using quadric error metrics. In *VIS '98: Proceedings of the conference on Visualization 1998*, pages 263–269, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press.

[42] B. Gärtner. Fast and robust smallest enclosing balls. In *Proceedings of 7th Annual European Symposium on Algorithms (ESA)*, volume 1643 of *Lecture Notes in Computer Science*, pages 325–338. Springer-Verlag, 1999.

[43] A. S. Glassner. An overview of ray tracing. In A. S. Glassner, editor, *An Introduction to Ray Tracing*, pages 1–32. Academic Press Ltd., San Diego (CA), USA, 1989.

[44] A. S. Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufmann Publishers, San Francisco (CA), USA, 1995.

[45] C. González, P. Castelló, and M. Chover. A texture-based metric extension for simplification methods. In *Proc. of GRAPP 2007, Barcelona, Spain*, pages 69–77, 2007.

[46] B. Gooch, E. Reinhard, C. Moulding, and P. Shirley. Artistic composition for image creation. In *Rendering Techniques*, pages 83–88, 2001.

[47] C. M. Goral, K. E. Torrance, D. P. Greenberg, and B. Battaile. Modelling the interaction of light between diffuse surfaces. *Computer Graphics (Proceedings of SIGGRAPH '84)*, 18(3):213–222, July 1984. DOI: 10.1145/964965.808601

[48] S. J. Gortler, P. Schröder, M. F. Cohen, and P. Hanrahan. Wavelet radiosity. In J. T. Kajiya, editor, *Computer Graphics (Proceedings of SIGGRAPH '93)*, volume 27 of *Annual Conference Series*, pages 221–230. ACM, New York (NY), USA, August 1993.

[49] T. Götzelmann, P.-P. Vázquez, K. Hartmann, A. Nürnberger, and T. Strothotte. Correlating text and images: Concept and evaluation. In *Proceedings of Smart Graphics*, volume 4569 of *Lecture Notes in Computer Science*, pages 97–109. Springer-Verlag, 2007. DOI: 10.1007/978-3-540-73214-3_9

[50] P. Grassberger. Toward a quantitative theory of self-generated complexity. *International Journal of Theoretical Physics*, 25(9):907–938, 1986. DOI: 10.1007/BF00668821

[51] R. M. Gray. *Entropy and Information Theory*. Springer-Verlag, New York (NY), USA, 1990.

[52] P. Hanrahan, D. Salzman, and L. Aupperle. A rapid hierarchical radiosity algorithm. *Computer Graphics (Proceedings of SIGGRAPH '91)*, 25(4):197–206, July 1991. DOI: 10.1145/127719.122740

[53] J. Harvda and F. Charvát. Quantification method of classification processes. Concept of structural α-entropy. *Kybernetika*, 3:30–35, 1967.

[54] E. D. Hellinger. Neue Begründung der Theorie der Quadratischen Formen von Unendlichen Vielen Veränderlichen[2]. *Journal für Reine und Angewandte Mathematik*, 136:210–271, 1909.

[55] H. Hoppe. Progressive meshes. *Proceedings of SIGGRAPH 96*, pages 99–108, August 1996. DOI: 10.1145/237170.237216

[56] H. Hoppe. New quadric metric for simplifying meshes with appearance attributes. In *VISUALIZATION '99: Proceedings of the 10th IEEE Visualization 1999 Conference (VIS '99)*, Washington (DC), USA, 1999. IEEE Computer Society.

[57] A. Iones, A. Krupkin, M. Sbert, and S. Zhukov. Fast, realistic lighting for video games. *IEEE Computer Graphics and Applications*, 23(3):54–64, 2003. DOI: 10.1109/MCG.2003.1198263

[58] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998. DOI: 10.1109/34.730558

[59] E. T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106(4):620–630, May 1957. DOI: 10.1103/PhysRev.106.620

[60] G. Ji and H.-W. Shen. Dynamic view selection for time-varying volumes. *Transactions on Visualization and Computer Graphics*, 12(5):1109–1116, 2006. DOI: 10.1109/TVCG.2006.137

[61] J. T. Kajiya. The rendering equation. *Computer Graphics (Proceedings of SIGGRAPH '86)*, 20(4):143–150, August 1986. DOI: 10.1145/15886.15902

[62] M. H. Kalos and P. A. Whitlock. *The Monte Carlo Method*. John Wiley & Sons Inc., 1986. DOI: 10.1002/9783527617395

[63] Z. Karni and C. Gotsman. Spectral compression of mesh geometry. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 279–286, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co. DOI: 10.1145/344779.344924

[64] A. Keller and S. Heinrich. Quasi-Monte Carlo methods in computer graphics, Part I: The QMC-buffer. Technical Report 242/94, University of Kaiserslautern, Kaiserslautern, Germany, 1994.

[65] Y. Kim and A. Varshney. Saliency-guided enhancement for volume visualization. *Transactions on Visualization and Computer Graphics*, 12(5):925–932, 2006. DOI: 10.1109/TVCG.2006.174

[2] A new foundation of the theory of quadratic forms of infinite many variables.

[66] D. Kirk and J. Arvo. Unbiased variance reduction for global illumination. In *Proceedings of the 2nd Eurographics Workshop on Rendering*, pages 153–156, May 1991.

[67] P. Kohlmann, S. Bruckner, A. Kanitsar, and M. E. Gröller. Livesync: Deformed viewing spheres for knowledge-based navigation. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1544–1551, 2007. DOI: 10.1109/TVCG.2007.70576

[68] A. N. Kolmogorov. On the Shannon theory of information transmission in the case of continuous signals. *IRE Transactions on Information Theory*, 2:102–108, 1956. DOI: 10.1109/TIT.1956.1056823

[69] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:76–86, 1951. DOI: 10.1214/aoms/1177729694

[70] H. Landis. Renderman in production. In *Course notes of ACM SIGGRAPH*, 2002.

[71] C. H. Lee, A. Varshney, and D. W. Jacobs. Mesh saliency. *ACM Transactions on Graphics*, 24(3):659–666, 2005. DOI: 10.1145/1073204.1073244

[72] M. E. Lee, R. A. Redner, and S. P. Uselton. Statiscally optimized sampling for distributed ray tracing. *Computer Graphics (Proceedings of SIGGRAPH '85)*, 19(3):61–67, July 1985. DOI: 10.1145/325165.325179

[73] W. Li. On the relationship between complexity and entropy for Markov chains and regular languages. *Complex Systems*, 5(4):381–399, 1991.

[74] P. Lindstrom and G. Turk. Image-driven simplification. *ACM Transactions on Graphics*, 19(3):204–241, 2000. DOI: 10.1145/353981.353995

[75] A. Lu, R. Maciejewski, and D. S. Ebert. Volume composition using eye tracking data. In *Proceedings of EuroVis 2006*, pages 655–662, 2006.

[76] D. P. Luebke and B. Hallen. Perceptually-driven simplification for interactive rendering. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, pages 223–234, London, UK, 2001. Springer-Verlag.

[77] M. McCool and E. Fiume. Hierarchical Poisson disk sampling distributions. In *Proceedings of Graphics Interface '92*, pages 94–105, May 1992.

[78] S. Melax. A simple, fast, and effective polygon reduction algorithm. *Game Developer*, pages 44–48, Nov. 1998.

[79] D. P. Mitchell. Generating antialiased images at low sampling densities. *Computer Graphics (Proceedings of SIGGRAPH '87)*, 21(4):65–72, July 1987. DOI: 10.1145/37402.37410

[80] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, New York (NY), USA, 1995.

[81] K. Mühler, M. Neugebauer, C. Tietjen, and B. Preim. Viewpoint selection for intervention planning. In *Proceedingss of Eurographics/ IEEE-VGTC Symposium on Visualization*, pages 267–274, 2007.

[82] T. Nishita and E. Nakame. Continuous tone representation of 3-D objects taking account of shadows and interreflection. *Computer Graphics (Proceedings of SIGGRAPH '85)*, 19(3):23–30, July 1985. DOI: 10.1145/325165.325169

[83] A. B. Novikoff. *Integral Geometry as a Tool in Pattern Perception*, pages 347–368. Pergamon Press, Elmsford (NY), USA, 1962.

[84] R. Ohbuchi and M. Aono. Quasi–Monte Carlo rendering with adaptive sampling. Technical report, Tokyo Research Laboratory, IBM Japan Ltd., Tokyo, Japan, 1996.

[85] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Shape distributions. *ACM Transactions on Graphics*, 21(4):807–832, October 2002. DOI: 10.1145/571647.571648

[86] D. L. Page, A. F. Koschan, S. R. Sukumar, B. Roui-Abidi, and M. A. Abidi. Shape analysis algorithm based on information theory. In *IEEE International Conference on Image Processing (Proceedings of ICIP'03)*, volume 1, pages 229–232, September 2003.

[87] J. Painter and K. Sloan. Antialiased ray tracing by adaptive progressive refinement. *Computer Graphics (Proceedings of SIGGRAPH '89)*, 23(3):281–288, July 1989. DOI: 10.1145/74334.74362

[88] S. Palmer, E. Rosch, and P. Chase. Canonical perspective and the perception of objects. *Attention and Performance IX*, pages 135–151, 1981.

[89] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, New York (NY), USA, 1984.

[90] M. Pardo and I. Vajda. On asymptotic properties of information-theoretic divergences. *IEEE Transactions on Information Theory*, 49(7):1860–1868, 2003. DOI: 10.1109/TIT.2003.813509

[91] K. Pearson. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *Philosophical Magazine*, V(1):157–175, 1900.

[92] M. S. Pinsker. *Information and Stability of Random Variables and Processes*. Izdatel'stvo Akademii Nauk SSSR, Moscow, Russia, 1960.

[93] D. Plemenos and M. Benayada. Intelligent display techniques in scene modelling. new techniques to automatically compute good views. In *International Conference GraphiCon'96*, 1996.

[94] O. Polonsky, G. Patanè, S. Biasotti, C. Gotsman, and M. Spagnuolo. What's in an image? *The Visual Computer*, 21(8-10):840–847, 2005. DOI: 10.1007/s00371-005-0326-y

[95] K. Pulli, M. F. Cohen, T. Duchamp, H. Hoppe, L. Shapiro, and W. Stuetzle. View-based rendering: Visualizing real objects from scanned range and color data. In J. Dorsey and P. Slusallek, editors, *Rendering Techniques '97 (Proceedings of the 8th Eurographics Workshop on Rendering)*, pages 23–34, New York (NY), USA, June 1997. Springer-Verlag.

[96] W. Purgathofer. A statistical method for adaptive stochastic sampling. *Eurographics '86: Proceedings of the European Conference and Exhibition*, 11(2):157–162, August 1986.

[97] R.-D. Reiss. *Approximate Distributions of Order Statistics: With Applications to Nonparametric Statistics*. Springer-Verlag, New York (NY), USA, 1989.

[98] A. Rényi. On measures of entropy and information. In *Proc. Fourth Berkeley Symp. Math. Stat. and Probability' 60*, volume 1, pages 547–561, Berkeley (CA), USA, 1961. University of California Press.

[99] J. Rigau. *Information-Theoretic Refinement Criteria for Image Synthesis*. PhD thesis, Universitat Politècnica de Catalunya, Barcelona, Spain, November 2006.

[100] J. Rigau, M. Feixas, P. Bekaert, and M. Sbert. View-dependent information theory measures for pixel sampling and scene discretization in flatland. In *Proceedings of Spring Conference on Computer Graphics '01*, pages 173–180, Los Alamitos (CA), USA, April 2001. IEEE Computer Society. DOI: 10.1109/SCCG.2001.945352

[101] J. Rigau, M. Feixas, and M. Sbert. New contrast measures for pixel supersampling. In J. Vince and R. Earnshaw, editors, *Advances in Modeling, Animation and Rendering (Proceedings of CGI '02)*, pages 439–451, London, UK, July 2002. Springer-Verlag London Limited.

[102] J. Rigau, M. Feixas, and M. Sbert. Entropy-based adaptive sampling. In *Graphics Interface*, pages 149–157. Canadian Information Processing Society, A. K. Peters Ltd., June 2003.

[103] J. Rigau, M. Feixas, and M. Sbert. Refinement criteria based on *f*-divergences. In P. H. Christensen and D. Cohen-Or, editors, *Rendering Techniques 2003 (14th Eurographics Symposium on Rendering)*, pages 260–269, New York (NY), USA, June 2003. Association for Computing Machinery.

[104] J. Rigau, M. Feixas, and M. Sbert. Shape complexity based on mutual information. In *International Conference on Shape Modeling and Applications (SMI '05)*, pages 355–360, Los Alamitos (CA), USA, June 2005. IEEE Computer Society. DOI: 10.1109/SMI.2005.42

[105] J. Rigau, M. Feixas, and M. Sbert. A generalised-mutual-information-based oracle for hierarchical radiosity. In Y. Shi, G. Albada, J. Dongarra, and P. M. Sloot, editors, *Computational Science - ICCS 2007, 7th International Conference*, volume 4488 (II) of *Lecture Notes in Computer Science*, pages 105–113. Springer-Verlag, May 2007.

[106] R. Y. Rubinstein. *Simulation and the Monte Carlo Method*. John Wiley & Sons Inc., New York (NY), USA, 1981. DOI: 10.1002/9780470316511

[107] L. A. Santaló. *Integral Geometry and Geometric Probability*. Cambridge University Press, 1976.

[108] M. Sbert. An integral geometry based method for fast form-factor computation. *Computer Graphics Forum (Proceedings of Eurographics '93)*, 12(3):409–420, 1993. DOI: 10.1111/1467-8659.1230409

[109] M. Sbert. *The Use of Global Random Directions to Compute Radiosity. Global Monte Carlo Methods*. PhD thesis, Universitat Politècnica de Catalunya, Barcelona, Spain, November 1996.

[110] M. Sbert, A. Brusi, R. F. Tobler, and W. Purgathofer. Random walk radiosity with generalized transition probabilities. Research Report IIiA–98–07–RR, Institut d'Informàtica i Aplicacions, Universitat de Girona, Girona, Spain, 1998.

[111] M. Sbert, D. Plemenos, M. Feixas, and F. González. Viewpoint quality: Measures and applications. In *Computational Aesthetics 2005 - First Eurographics Workshop on Computational Aesthetics in Graphics, Visualization and Imaging (CAGVI '05)*, pages 185–192, Aire-la-Ville, Switzerland, May 2005. Eurographics Association.

[112] C. Schlick. An adaptive sampling technique for multidimensional ray tracing. In *Proceedings of the 2nd Eurographics Workshop on Rendering*, pages 48–56, May 1991.

[113] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, July, October 1948. DOI: 10.1145/584091.584093

[114] B. D. Sharma and D. P. Mittal. New non-additive measures of entropy for a discrete probability distribution. *Journal of Mathematical Sciences (India)*, 10:28–40, 1975.

[115] B. D. Sharma and I. J. Taneja. Entropy of type (α, β) and other generalized measures in information theory. *Metrika*, 22(1):205–215, 1975. DOI: 10.1007/BF01899728

[116] P. Shirley. A ray tracing method for illumination calculation in diffuse-specular scenes. In *Proceedings of Graphics Interface '90*, pages 205–212, Toronto (Ontario), Canada, May 1990. Canadian Information Processing Society.

[117] F. X. Sillion and C. Puech. *Radiosity and Global Illumination*. Morgan Kaufmann Publishers, San Francisco (CA), USA, 1994.

[118] M. Simmons and C. H. Séquin. Tapestry: A dynamic mesh-based display representation for interactive rendering. In B. Péroche and H. Rushmeier, editors, *Rendering Techniques 2000 (Proceedings of the 11th Eurographics Workshop on Rendering)*, pages 329–340, New York (NY), USA, June 2000. Springer-Verlag.

[119] N. Slonim and N. Tishby. Agglomerative information bottleneck. In *Proceedings of NIPS-12 (Neural Information Processing Systems)*, pages 617–623. MIT Press, 2000.

[120] N. Slonim and N. Tishby. Document clustering using word clusters via the information bottleneck method. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 208–215. ACM Press, 2000. DOI: 10.1145/345508.345578

[121] B. E. Smits, J. Arvo, and D. Salesin. An importance-driven radiosity algorithm. *Computer Graphics (Proceedings of SIGGRAPH '92)*, 26(2):273–282, July 1992. DOI: 10.1145/142920.134080

[122] D. Sokolov, D. Plemenos, and K. Tamine. Methods and data structures for virtual world exploration. *The Visual Computer*, 22(7):506–516, 2006. DOI: 10.1007/s00371-006-0025-3

[123] L. Szirmay-Kalos. *Monte Carlo Methods in Global Illumination*. Institute of Computer Graphics, Vienna University of Technology, Vienna, Austria, 2000.

[124] S. Takahashi, I. Fujishiro, Y. Takeshima, and T. Nishita. A feature-driven approach to locating optimal viewpoints for volume visualization. In *IEEE Visualization 2005*, pages 495–502, 2005. DOI: 10.1109/VISUAL.2005.1532834

[125] R. Tamstorf and H. W. Jensen. Adaptive sampling and bias estimation in path tracing. In J. Dorsey and P. Slusallek, editors, *Rendering Techniques '97 (Proceedings of the 8th Eurographics Workshop on Rendering)*, pages 285–295, New York (NY), USA, June 1997. Springer-Verlag.

[126] I. J. Taneja. Bivariate measures of type α and their applications. *Tamkang Journal of Mathematics*, 19(3):63–74, 1988.

[127] I. J. Taneja. On generalized information measures and their applications. In *Advances in Electronics and Electron Physics*, volume 76, pages 327–413. Academic Press Ltd., 1989.

[128] M. Tarr, H. Bülthoff, M. Zabinski, and V. Blanz. To what extent do unique parts influence recognition across changes in viewpoint? *Psychological Science*, 8(4):282–289, 1997. DOI: 10.1111/j.1467-9280.1997.tb00439.x

[129] N. Tishby, F. C. Pereira, and W. Bialek. The information bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

[130] F. Topsøe. Some inequalities for information divergence and related measures of discrimination. *IEEE Transactions on Information Theory*, 46(4):1602–1609, 2000. DOI: 10.1109/18.850703

[131] C. Tsallis. Possible generalization of Boltzmann-Gibbs statistics. *Journal of Statistical Physics*, 52(1/2):479–487, 1988. DOI: 10.1007/BF01016429

[132] C. Tsallis. Generalized entropy-based criterion for consistent testing. *Physical Review E*, 58:1442–1445, 1998. DOI: 10.1103/PhysRevE.58.1442

[133] C. Tsallis. Entropic nonextensivity: A possible measure of complexity. *Chaos, Solitons, & Fractals*, 13(3):371–391, 2002. DOI: 10.1016/S0960-0779(01)00019-4

[134] P. P. Vázquez. *On the Selection of Good Views and its Application to Computer Graphics*. PhD thesis, Universitat Politècnica de Catalunya, April 2003.

[135] P. P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich. Viewpoint selection using viewpoint entropy. In T. Ertl, B. Girod, G. Greiner, H. Niemann, and H.-P. Seidel, editors, *Proceedings of Vision, Modeling, and Visualization 2001*, pages 273–280, Stuttgart, Germany, November 2001.

[136] P.-P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich. Automatic view selection using viewpoint entropy and its applications to image-based modelling. *Computer Graphics Forum*, 22(4):689–700, 2003. DOI: 10.1111/j.1467-8659.2003.00717.x

[137] P. P. Vázquez, M. Feixas, M. Sbert, and A. Llobet. A new tool for obtaining good views for molecules. In D. Ebert, P. Brunet, and I. Navazo, editors, *Proceedings of VisSym '02 (Eurographics-IEEE TCVG Symposium on Visualization)*, pages 183–188, May 2002.

[138] P. P. Vázquez, M. Feixas, M. Sbert, and A. Llobet. Realtime automatic selection of good molecular views. *Computers & Graphics*, 30(1):98–110, 2006. DOI: 10.1016/j.cag.2005.10.022

[139] S. Verdú. Fifty years of Shannon theory. *IEEE Transactions on Information Theory*, 44(6):2057–2078, October 1998. DOI: 10.1109/18.720531

[140] I. Viola, M. Feixas, M. Sbert, and M. E. Gröller. Importance-driven focus of attention. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):933–940, 2006. DOI: 10.1109/TVCG.2006.152

[141] T. Whitted. An improved illumination model for shaded display. *Communications of the ACM (Graphics and Image Processing)*, 23(6):343–349, June 1980. DOI: 10.1145/358876.358882

[142] N. Williams, D. Luebke, J. D. Cohen, M. Kelley, and B. Schubert. Perceptually guided simplification of lit, textured meshes. In *SI3D '03: Proceedings of the 2003 symposium on Interactive 3D graphics*, pages 113–121, New York, NY, USA, 2003. ACM Press. DOI: 10.1145/641480.641503

[143] E. Wong and J. Steppe. Invariant recognition of geometric shapes. In S. Watanabe, editor, *Methodologies of Pattern Recognition*, pages 535–546. Academic Press, New York (NY), USA, 1969.

[144] H. Yamauchi, W. Saleem, S. Yoshizawa, Z. Karni, A. G. Belyaev, and H.-P. Seidel. Towards stable and salient multi-view representation of 3d shapes. In *IEEE International Conference on Shape Modeling and Applications*, pages 265–270, 2006. DOI: 10.1109/SMI.2006.42

[145] Y. Yang and A. Barron. Information theoretic determination of minimax rates of convergence. *Annals of Statistics*, 27:1546–1599, 1999. DOI: 10.1214/aos/1017939142

[146] R. W. Yeung. *Information Theory and Network Coding*. Information Technology: Transmission, Processing and Storage. Springer, 2008.

[147] E. Zhang and G. Turk. Visibility-guided simplification. In *VIS '02: Proceedings of the conference on Visualization 2002*, pages 267–274, Washington, DC, USA, 2002. IEEE Computer Society. DOI: 10.1109/VISUAL.2002.1183784

[148] S. Zhukov, A. Iones, and G. Kronin. An ambient light illumination model. In *Rendering Techniques*, pages 45–56, 1998.

Author Biographies

MATEU SBERT

Mateu Sbert is a professor in Computer Science at the University of Girona, Spain. He received an M.Sc. in Theoretical Physics (1977) at the University of Valencia, an M.Sc. in Mathematics (Statistics and Operations Research, 1983) at U.N.E.D. University (Madrid), and a Ph.D. in Computer Science at the U.P.C. (Universitat Politècnica de Catalunya, 1997, Best Ph.D. Award). Mateu Sbert's research interests include the application of Monte Carlo and Information Theory techniques to Computer Graphics and Image Processing. He has authored or co-authored more than one hundred and fifty papers in his areas of research, served as a member of program committee in Spanish and international conferences, and participated in several European and Spanish research projects, leading the VIth European Framework Gametools project. Mateu Sbert coorganized the Dagsthul Seminars "Stochastic Methods in Rendering" and "Computational Aesthetics in Graphics, Visualization and Imaging."

MIQUEL FEIXAS

Miquel Feixas is an associate professor in Computer Science at the University of Girona, Spain. He received an M.Sc. in Theoretical Physics (1979) at the UAB (Universitat Autònoma de Barcelona) and a Ph.D. in Computer Science at the UPC (Universitat Politècnica de Catalunya, 2002). His research is focused on the application of Information Theory techniques to Radiosity, Global Illumination, Viewpoint Selection, Scientific Visualization, Image Processing, Medical Imaging, and Computational Aesthetics. He has authored or co-authored more than fifty papers in his areas of research. He acted as a reviewer for conferences in the field of computer graphics and image processing. He has participated in several European and Spanish research projects.

JAUME RIGAU

Jaume Rigau is an associate professor in Computer Science at the University of Girona (UdG, Spain). He received an M.Sc. in Computer Science and a Ph.D. at the Technical University of Catalonia (UPC, Spain). His research is focused on the application of Information Theory to Computer Graphics and Image Processing. He has co-authored several papers in these areas of research and participated in European research projects and joint actions.

MIGUEL CHOVER

Miguel Chover is an Associate Professor at the Universitat Jaume I de Castellón, Spain. His research focuses on interactive computer graphics, computer games, and Web3D. His current work includes level of detail modelling, simplification algorithms, rendering natural phenomena, and texturing techniques. He received his M.Sc. degree in Computer Science in 1992 and a Ph.D. in Computer Science in 1996 from the Universidad Politécnica de Valencia, Spain. He is member of Eurographics.

IVAN VIOLA

Ivan Viola is an Associate Professor at University of Bergen, and a scientific adviser at Christian Michelsen Research (CMR), Bergen, Norway. He received an M.Sc. in 2002 and Ph.D. in 2005 from Vienna University of Technology, Austria. His research is focused on application of illustrative visualization for communication of complex scientific data. Viola co-authored several scientific works published in international journals and conferences such as IEEE TVCG, IEEE Visualization, and EuroVis and acted as a reviewer and IPC member for conferences in the field of computer graphics and visualization (e.g., IPC IEEE Visualization 2009, EuroVis 2009). He is member of Eurographics, NorSIGD, IEEE Computer Society, VGTC.

Index