SPRINGER BRIEFS IN COMPUTER SCIENCE

André Bigand · Julien Dehos
Christophe Renaud
Joseph Constantin

Image Quality Assessment of Computer-generated Images

Based on Machine Learning and Soft Computing

Springer

SpringerBriefs in Computer Science

More information about this series at http://www.springer.com/series/10028

André Bigand · Julien Dehos
Christophe Renaud · Joseph Constantin

Image Quality Assessment of Computer-generated Images

Based on Machine Learning and Soft Computing

Springer

André Bigand
LISIC
Université du Littoral Côte d'Opale
Calais Cedex
France

Julien Dehos
Université du Littoral Côte d'Opale
Dunkirk
France

Christophe Renaud
Université du Littoral Côte d'Opale
Dunkirk
France

Joseph Constantin
Faculty of Sciences II
Lebanese University
Beirut
Lebanon

ISSN 2191-5768 ISSN 2191-5776 (electronic)
SpringerBriefs in Computer Science
ISBN 978-3-319-73542-9 ISBN 978-3-319-73543-6 (eBook)
https://doi.org/10.1007/978-3-319-73543-6

Library of Congress Control Number: 2018932548

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

The measure of image (videos) quality remains a research challenge and a very active field of investigation considering image processing. One solution consists of providing a subjective score to the image quality (according to a reference or without reference) obtained from human observers. The setting of such psycho-visual tests is very expensive (considering time and human organization) and needs clear and strict proceedings. Algorithmic solutions have been developed (objective scores) to avoid such tests. Some of these techniques are based on the modeling of the Human Visual System (HVS) to mimic the human behavior, but they are complex. In the case of natural scenes, a great number of image (or video) quality databases exist that makes possible the validation of these different techniques. Soft computing (machine learning, fuzzy logic, etc.), widely used in many scientific fields such as biology, medicine, management sciences, financial sciences, plant control, etc., is also a very useful cross-disciplinary tool in image processing. These tools have been used to establish image quality and they are now well known.

Emerging topics these last years concern **image synthesis**, applied in virtual reality, augmented reality, movie production, interactive video games, etc. For example, unbiased global illumination methods based on stochastic techniques can provide photo-realistic images in which content is indistinguishable from real photography. But there is a price: these images are prone to noise that can only be reduced by increasing the number of computed samples of the involved methods and consequently increasing their computation time. The problem of finding the number of samples that are required in order to ensure that most of the observers cannot perceive any noise is still open since the ideal image is unknown.

Image Quality Assessment (IQA) is well known considering natural scene images. Image quality (or noise evaluation) of computer-generated images is slightly different, since image generation is different and databases are not yet developed. In this short book, we address this problem by focusing on **visual perception of noise**. But rather than use known perceptual models, we investigate the use of **soft computing approaches** classically used in the Artificial Intelligence (AI) areas such as full-reference and reduced-reference metrics. We propose to use

such approaches to create a machine learning model based on learning machines such as SVMs and RVMs in order to be able to predict which image highlights perceptual noise. We also investigate the use of **interval-valued fuzzy sets** as no-reference metric. Learning is performed through the use of an example database which is built from experiments of noise perception with human users. These models can then be used in any progressive stochastic global illumination method in order to find the visual convergence threshold of different parts of any image.

The short book is organized as follows: after a brief introduction (Chap. 1), Chap. 2 describes the Monte Carlo methods for image synthesis we use, and then chapter briefly describes the visual impact of rendering on image quality and the interest of a noise model. In Chap. 4, image quality evaluation using SVMs and RVMs is introduced and in Chap. 5 new learning algorithms that can be applied with interesting results are presented. Chapter 6 introduces an original method obtained from the application of fuzzy sets entropy. Finally, the short book is summarized with some conclusions in Chap. 7.

The goal of this book is to present an emerging topic, that is to say IQA for computer-generated images, to students (and others) practitioners of image processing and related areas such as computer graphics and visualization. In addition, students and practitioners should be familiar with the underlying techniques that make this possible (basics of image processing, machine learning, fuzzy sets). This monograph will be interesting for all people involved in image generation, virtual reality, augmented reality, and all new trends emerging around these topics.

Calais Cedex, France — André Bigand
Dunkirk, France — Julien Dehos
Dunkirk, France — Christophe Renaud
Beirut, Lebanon — Joseph Constantin

Acknowledgements

Experiments presented in this book (Chaps. 2 and 3) were carried out using the CALCULCO computing platform, supported by SCoSI/ULCO (Service COmmun du Système d'Information de l'Université du Littoral Côte d'Opale), and the open-source renderer PBRT-v3 by Matt Pharr, Wenzel Jakob, and Greg Humphreys (http://pbrt.org).

Contents

About the Authors

André Bigand (IEEE Member) received a Ph.D. from the University Paris 6 and the HdR degree in from the Université du Littoral Côte d'Opale (ULCO, France). He is currently senior associate professor in ULCO since. His current research interests include uncertainty modeling and machine learning with applications to image processing and image synthesis (particularly noise modeling and filtering). He is currently with the LISIC Laboratory (ULCO). He is author and coauthor of scientific papers in international journals and books or communications to conferences with reviewing committee. He has years of experience teaching and lecturing. He is a visiting professor at UL—Lebanese University—where he teaches "machine learning and pattern recognition" in research master STIP. E-mail: bigand@lisic.univ-littoral.fr. Website: http://www-lisic.univ-littoral.fr/~bigand/.

Joseph Constantin obtained an M.S. in Software Engineering and Systems Modeling from the Lebanese University in 1997 and Ph.D. in Automatic and Robotic control from the Picardie Jules Verne University, France, in 2000. Since 2001, he has been a senior associate professor at the Lebanese University, Faculty of Sciences and a researcher in the Applied Physics Laboratory of the Doctoral School of Sciences and Technology at the Lebanese University. His current research interests are in the fields of machine learning, image processing, robot dynamics and control, diagnosis systems, and biomedical engineering.

Christophe Renaud is Full Professor of Computer Science at Université du Littoral Côte d'Opale (ULCO). He received a Ph.D. in 1993 from the University of Lille and the HdR degree in 2002 from ULCO. His current research interests focus on photo-realistic rendering and on image processing and artificial intelligence applied to rendering techniques. He currently develops collaborations in the area of digital humanities with art historians and psychologists.

Julien Dehos is an Associate Professor of computer science at Université du Littoral Côte d'Opale (ULCO). His research interests include image synthesis,

image processing, and artificial intelligence. He received an engineer's degree from the ENSEIRB school and M.S. from the University Bordeaux 1 in 2007, and his Ph. D. from ULCO in 2010.

Chapter 1
Introduction

Image Quality Assessment (IQA) aims to characterize the visual quality of an image. Indeed, there are many sources of image degradation, for example, optical distortion, sensor noise, compression algorithms, etc., so IQA is useful to evaluate the perceived quality of an image or to optimize an imaging process. IQA has been well studied for natural-scene images (captured by a camera) but there is far less work for computer-generated images (rendered from a virtual scene). This book aims to review the recent advances in Image Quality Assessment for computer-generated images.

1.1 Natural-Scene Images, Computer-generated Images

Natural-scene images are obtained by sampling and digitizing the light coming from a natural scene, with a sensor (CCD, CMOS, etc.). Many aspects are important to obtain "good quality" images: lighting conditions, optical system of the camera, sensor quality, etc. An exhaustive presentation about those topics is given in (Xu et al. 2015). The authors present the methods involved in subjective visual quality and in objective visual quality assessment. Particularly, they also present image and video quality databases which are very important to compare the obtained scores, and they address the interest of machine learning for IQA. So, we will not consider these topics once more and we recommend the reader to consult this presentation if necessary.

High-quality computer-generated images are obtained from computer simulations of light transport in virtual 3D scenes. Computing such a photo-realistic image requires to model the virtual scene precisely: light sources, object geometries, object materials, virtual camera, etc. It also requires to use a physically based rendering algorithm which accurately simulates the light propagation in the virtual scene and the light–matter interactions. Today, the vast majority of the physically based renderers

A. Bigand et al., *Image Quality Assessment of Computer-generated Images*,
SpringerBriefs in Computer Science, https://doi.org/10.1007/978-3-319-73543-6_1

are based on stochastic methods. Path tracing (Kajiya 1986) is a core rendering algorithm which generates many random paths from the camera to a light source, through the virtual scene. Since the paths are chosen randomly, the light contribution can change greatly from one path to another, which can generate high-frequency color variations in the rendered image (Shirley et al. 1996) known as perceptual noise. The Monte Carlo theory ensures that this process will converge to the correct image when the number of sample paths grows; however, this may require a great number of paths and a high computation time (typically hours per image). Thus, to render an image in an acceptable time, it is important to compute a number of paths as small as possible. However, it is difficult to predict how many sample paths are really required to obtain a "good quality" image or which random paths are the best for increasing the convergence rate. Moreover, it is even difficult to determine if a rendered image is sufficiently converged.

To summarize, the main differences between natural-scene images and computer-generated images (for IQA) are the following:

- Since perceptual noise is intrinsic to the image generation process, a computer-generated image is converged when no perceptual noise is noticeable in the final image.
- Image databases for computer-generated images are limited and costly to obtain (psycho-visual index obtained from human observers).
- Noise features are the most important image features to consider for computer-generated images.

The final use of computer-generated images is to be seen by human observers who are generally very sensitive to image artifacts. The Human Visual System (HVS) is endowed with powerful performances but is a very complex process. Consequently, perception-driven approaches were proposed to determine if rendered image are converged. The main idea of such approaches is to replace the human observer by a vision model. By mimicking HVS, such techniques can provide important improvements for rendering. They can be used for driving rendering algorithms to visually satisfactory images and to focus on visually important features (Mitchell 1987; Farrugia and Péroche 2004; Longhurst et al. 2006). HVS models provide interesting results but are complex, still incomplete, and difficult to set up, and generally require relatively long computation times. Therefore, the methods presented in this book focus on the use of a new **noise based perceptual index** to replace psycho-visual index in the perception-driven model assessment. Perceptual noise is considered from a machine learning point of view (noise features) or a soft computing point of view (fuzzy entropy used to set up noise level).

1.2 Image Quality Assessment Models

Image quality assessment models are very important to characterize the visual quality of an image. For example, they are of great interest for image compression (JPEG models) and natural image characterization (Lahoudou et al. 2010). In the literature,

IQA models are usually classified into three families (see (Lahoudou et al. 2011; Beghdadi et al. 2013) for a brief review of IQA and machine learning):

- *Full-reference models* that use the original version of the image for estimating the quality of the processed version. These models are the most used methods to evaluate image quality (for example, the well-known PSNR and SSIM). They are easy to compute in real time and correlated with human subjective appreciation but require a reference image. Unfortunately, these models are not applicable for computer-generated images since the final reference image is not already known during the image generation process.
- *No-reference models* that evaluate the quality of images without access to reference images. Some recent papers (Ferzli and Karam 2005; Zhang et al. 2011) proposed no-reference quality assessment methods with good results but limited to JPEG images. Other methods were proposed for computer-generated images with some success (Delepoulle et al. 2012) but a complete framework has to be yet defined.
- *Reduced-reference models* that analyze the processed image using some relevant information to calculate the quality of the result image. This model seems to be particularly interesting for our study as we will show in the following of the book.

In the last decade, numerous IQA methods for computer-generated images have been proposed but the resulting models are limited in practice and they are still under investigation. Currently, the classical model to characterize image quality remains psycho-visual experiments (Human in the loop experiment (Faugeras 1979)).

1.3 Organization of the Book

In this book, we assume that the reader is familiar with the basic aspects of machine learning and image processing, and we only focus on the visual quality assessment of computer-generated images using soft computing. We present recent techniques to assess if such a photo-realistic computer-generated image is noisy or not, based on full-reference, reduced-reference, and no-reference image quality methods, using learning machines and fuzzy sets. These techniques make it possible to set up uncertainty brought by perceptual noise affecting the image synthesis process. Note that we mainly focus on grayscale images, or more precisely the "L" component of "Lab" color images, since noise mainly affects this component, (Carnet et al. 2008).

In Chap. 2, we introduce image rendering to the reader. We present the basic notions of light transport and the equations that formalize rendering. We then recall the Monte Carlo method and detail the path-tracing algorithm which is the core of many renderers currently used in the computer graphics industry.

In Chap. 3, we study the visual impact of the rendering process on the quality of the rendered image. We present experimental results obtained from a path-tracing renderer and illustrate the influence of several parameters (virtual scene and rendering parameters) on the visual quality of the rendered image.

Chapter 4 introduces image quality evaluation using full-reference methods. We present a conventional way to obtain noise attributes from computer-generated images and also introduce the use of deep learning to automatically extract them. We then present how to use Support Vector Machines (SVM) and Relevance Vector Machines (RVM) as image quality metrics.

Chapter 5 introduces image quality evaluation using reduced-reference methods. We present Fast Relevance Vector Machines (FRVM) and explain image quality evaluation using FRVM and inductive learning. Both methods are then compared on experimental results.

Chapter 6 introduces no-reference methods using fuzzy sets. We present the Interval-Valued Fuzzy Set (IVFS) and an entropy based on IVFS. We then detail an image noise estimation method which uses IVFS and presents promising experimental results obtained with computer-generated images.

In conclusion, Chap. 7 summarizes the important notions presented in this book and gives some perspectives.

References

Beghdadi A, Larabi M, Bouzerdoum A, Iftekharuddin K (2013) A survey of perceptual image processing methods. Sig Process Image Commun 28:811–831

Carnet M, Callet PL, Barba D (2008) Objective quality assessment of color images based on a generic perceptual reduced reference. Sig Process Image Commun 23(4):239–256

Delepoulle S, Bigand A, Renaud C (2012) A no-reference computer-generated images quality metrics and its application to denoising. In: IEEE intelligent systems IS'12 conference, vol 1, pp 67–73

Farrugia J, Péroche B (2004) A progressive rendering algorithm using an adaptive perceptually based image metric. Comput Graph Forum 23(3):605–614

Faugeras O (1979) Digital color image processing within the framework of a human visual model. IEEE Trans ASSP 27:380–393

Ferzli R, Karam L (2005) No-reference objective wavelet based noise immune image sharpness metric. In: International conference on image processing

Kajiya J (1986) The rendering equation. Comput Graph ACM 20(4):143–150

Lahoudou A, Viennet E, Haddadi M (2010) Variable selection for image quality assessment using a neural network based approach. In: 2nd European workshop on visual information processing (EUVIP), pp 45–49

Lahoudou A, Viennet E, Bouridane A, Haddadi M (2011) A complete statistical evaluation of state of the art image quality measures. In: The 7th international workshop on systems, signal processing and their applications, pp 219–222

Longhurst P, Debattista K, Chalmers A (2006) A GPU based saliency map for high-fidelity selective rendering. In: AFRIGRAPH 2006 4th international conference on computer graphics. Virtual reality, visualisation and interaction in Africa, pp 21–29

Mitchell D (1987) Generating antialiased images at low sampling densities. In: Proceedings of SIGGRAPH'87, New York, NY, USA, pp 65–72

Shirley P, Wang C, Zimmerman K (1996) Monte Carlo techniques for direct lighting calculations. ACM Trans Graph 15(1):1–36

Xu L, Lin W, Kuo CCJ (2015) Visual Quality Assessment by Machine Learning, vol 28. Springer Brief, London

Zhang J, Ong S, Thinh M (2011) Kurtosis based no-reference quality assessment of jpeg2000 images. Sig Process Image Commun 26(1):13–23

Chapter 2
Monte Carlo Methods for Image Synthesis

2.1 Introduction

Image synthesis (also called rendering) consists in generating an image from a virtual 3D scene (composed of light sources, objects, materials, and a camera). Numerous rendering algorithms have been proposed since the 1970s: z-buffer (Catmull 1974), ray tracing (Whitted 1980), radiosity (Goral et al. 1984), path tracing (Kajiya 1986), and Reyes (Cook et al. 1987)…

Physically based rendering algorithms (also called photo-realistic rendering algorithms) try to satisfy the physical rules describing the light transport. These algorithms are commonly used to generate high-quality images (see Fig. 2.1), for example, in the cinema industry, and include path tracing, photon mapping (Jensen 2001), bidirectional path tracing (Lafortune and Willems 1993; Veach and Guibas 1994), and metropolis light transport (Veach and Guibas 1997)…

In this book, we only consider the path-tracing algorithm since it is widely used in modern renderers and is the basis of many other rendering algorithms. In this chapter, we present the fundamental notions of light transport, which physically describes rendering. Then, we present the Monte Carlo method, which is the core computing method used in physically based rendering algorithms. Finally, we detail the path-tracing algorithm.

2.2 Light Transport

2.2.1 Radiometry

Radiometry is the science of measurement of electromagnetic radiation, including visible light. It is particularly useful for describing light transport and rendering algorithms (Nicodemus et al. 1977; Glassner 1994; Jensen 2001; Pharr and Humphreys

A. Bigand et al., *Image Quality Assessment of Computer-generated Images*,
SpringerBriefs in Computer Science, https://doi.org/10.1007/978-3-319-73543-6_2

Fig. 2.1 Physically based algorithms can render high-quality images from a virtual 3D scene

2010; Jakob 2013). The notations used in this chapter are mainly inspired from Eric Veach's PhD thesis (Veach 1997).

2.2.1.1 Radiant Flux

Radiant flux (Φ) is the quantity of energy per unit of time (watt):

$$\Phi = \frac{\mathrm{d}Q}{\mathrm{d}t} \quad [W] \tag{2.1}$$

Radiant flux measures the light received or emitted by a point of the scene (see Fig. 2.2).

2.2.1.2 Radiance

Radiance (L) is the flux per unit of area and per unit of projected solid angle (watt per square meter per steradian):

$$L(x \to x') = \frac{\mathrm{d}^2\Phi(x \to x')}{G(x \leftrightarrow x')\mathrm{d}A(x)\mathrm{d}A(x')} \quad [W.m^{-2}.sr^{-1}], \tag{2.2}$$

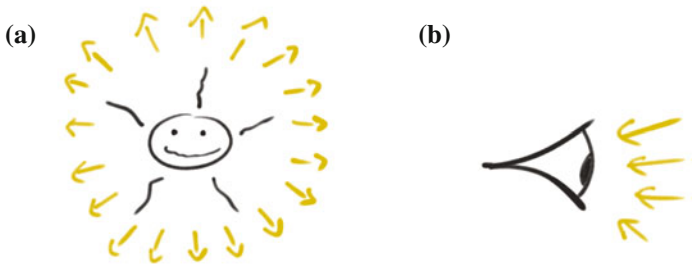


Fig. 2.2 The radiant flux is the quantity of light emitted from a point or received by a point

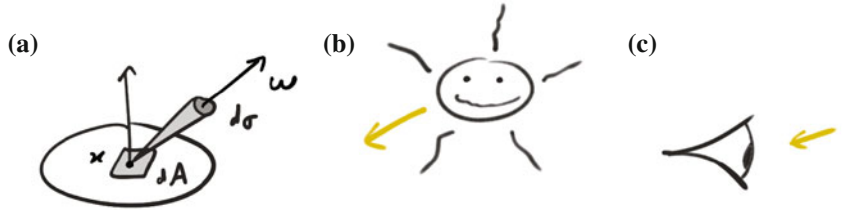


Fig. 2.3 Radiance is the flux emitted or received through a beam in a given direction

where G is the geometric function between the emitting surface and the receiving surface. The notation $x \rightarrow x'$ indicates the direction of light flow. The notation $G(x \leftrightarrow x')$ indicates a symmetric function.

Radiance measures the flux received or emitted by a point through a beam (see Fig. 2.3). It is particularly useful for describing light transport in a scene.

2.2.1.3 Bidirectional Reflectance Distribution Function

The Bidirectional Reflectance Distribution Function (BRDF) f_s describes the ratio of radiance reflected from an incoming direction to an outgoing direction:

$$f_s(x \rightarrow x' \rightarrow x'') = \frac{\mathrm{d}L(x' \rightarrow x'')}{L(x \rightarrow x')G(x \leftrightarrow x')\mathrm{d}A(x)} \quad [sr^{-1}] \tag{2.3}$$

The BRDF is useful for defining how a material reflects light (see Fig. 2.4).

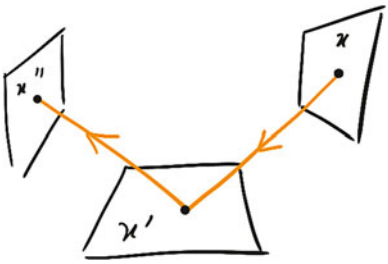


Fig. 2.4 The BRDF describes how a material reflects light from an incoming direction $(x \to x')$ toward an outgoing direction $(x' \to x'')$

2.2.2 *Formulation of Light Transport*

Using the previous radiometric quantities, we can formulate light transport, from the sources of the scene to the camera, and thus synthesize an image. Note that light transport can be formulated from light sources to camera as well as from camera to light sources, since it satisfies energy conservation.

2.2.2.1 Measurement Equation

Rendering consists in computing the radiance received by each pixel of the camera. The intensity I of a given pixel is defined by the measurement equation:

$$I = \int_{\mathscr{M} \times \mathscr{M}} W_e(x' \to x'') L(x' \to x'') G(x' \leftrightarrow x'') \mathrm{d}A(x') \mathrm{d}A(x''), \tag{2.4}$$

where $\mathscr{M}$ is the set of all points in the scene and W_e the response of the camera. The measurement equation simply states that the intensity of a pixel is the sum of the radiances from all points x' of the scene to all points x'' on the pixel (see Fig. 2.5).

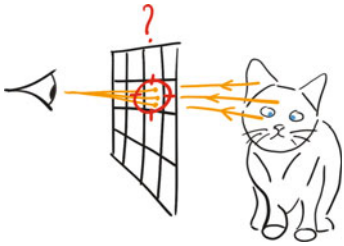


Fig. 2.5 The intensity of a pixel can be computed using the measurement equation, i.e., the integral of radiance from scene points to pixel points

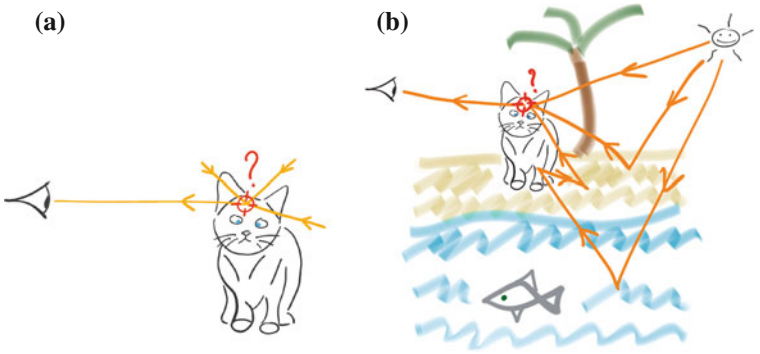


Fig. 2.6 The rendering equation defines how the light is reflected from all incoming directions to an outgoing direction (**a**). It can be applied recursively for incoming directions to fully compute light transport in the scene (**b**)

2.2.2.2 Rendering Equation

The measurement equation describes how a point x' of the scene contributes to the intensity of a pixel at a point x''. To synthesize an image, we also have to compute the radiance from the scene point x' toward the pixel point x'', which is described by the rendering equation:

$$L(x' \to x'') = L_e(x' \to x'') + \int_{\mathcal{M}} f_s(x \to x' \to x'') L(x \to x') G(x \leftrightarrow x') \mathrm{d}A(x), \tag{2.5}$$

where L_e is the light emitted at point x' (light source). Thus, the radiance received by x'' from x' is the sum of two terms: the light emitted by x' toward x'' and the light coming from all points x of the scene and reflected at x' toward x'' (see Fig. 2.6a).

Thus, we can compute the light at x' using the rendering equation. However, this requires to compute the light coming from other points x, i.e., to compute the rendering equation recursively at these points (see Fig. 2.6b).

2.3 Monte Carlo Integration

The measurement Eq. 2.4 and the rendering Eq. 2.5 are well-defined integral equations. However, they are difficult to solve using analytic solutions or deterministic numerical solutions, due to the complexity of the integrands and the high number of dimensions. Stochastic methods, such as Monte Carlo integration, are more suitable for computing such equations. Monte Carlo integration is the core of many physically based rendering algorithms such as path tracing.

2.3.1 Monte Carlo Estimator

Monte Carlo integration aims at evaluating the integral:

$$I = \int_{\Omega} f(x) \mathrm{d}\mu(x), \tag{2.6}$$

where $\mathrm{d}\mu$ is a measure on the domain Ω. This integral can be estimated by a random variable I_N:

$$I_N = \frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)}, \tag{2.7}$$

where X_1, ..., X_N are points of Ω sampled independently using the density function p. We can show the validity of this estimator by computing the expected value of I_N:

$$\begin{aligned} E[I_N] &= E\left[\frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)}\right] \\ &= \frac{1}{N} \sum_{i=1}^{N} E\left[\frac{f(X_i)}{p(X_i)}\right] \\ &= \frac{1}{N} \sum_{i=1}^{N} \int_{\Omega} \frac{f(x)}{p(x)} p(x) \mathrm{d}\mu(x) \\ &= \int_{\Omega} f(x) \mathrm{d}\mu(x) \\ &= I, \end{aligned} \tag{2.8}$$

using the linearity and the definition of expected value.

Thus, Monte Carlo integration converges to the correct solution. Moreover, it is simple to implement since it only requires to evaluate f and to sample points according to p. Finally, integrating high-dimensional functions is straightforward and only requires to sample all dimensions of the domain.

2.3.2 Convergence Rate

The variance of the Monte Carlo estimator decreases linearly with the number of samples:

$$\begin{aligned} V[I_N] &= V\left[\frac{1}{N}\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)}\right] \\ &= \frac{1}{N^2}V\left[\sum_{i=1}^{N}\frac{f(X_i)}{p(X_i)}\right] \\ &= \frac{1}{N^2}\sum_{i=1}^{N}V\left[\frac{f(X_i)}{p(X_i)}\right] \\ &= \frac{1}{N}V\left[\frac{f(X)}{p(X)}\right] \end{aligned} \tag{2.9}$$

Hence the standard deviation:

$$\begin{aligned} \sigma[I_N] &= \sqrt{V[I_N]} \\ &= \sqrt{\frac{1}{N}V\left[\frac{f(X)}{p(X)}\right]} \\ &= \frac{1}{\sqrt{N}}\sigma\left[\frac{f(X)}{p(X)}\right] \end{aligned} \tag{2.10}$$

Thus, the RMS error converges at a rate of $\mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$. This convergence rate is slow (increasing the number of samples by a factor of four only reduces the integration error by a factor of two) but it is not affected by the number of dimensions.

2.3.3 Variance Reduction Using Importance Sampling

Many variance reduction techniques have been proposed to improve convergence rate of Monte Carlo methods. One of them, importance sampling, is classically implemented in physically based renderer.

The basic idea of the importance sampling technique is to sample important regions of the domain with a higher probability. Ideally, we would choose a density function p proportional to f:

$$p(x) \propto f(x) \tag{2.11}$$

which leads to a zero-variance estimator, i.e., constant for all samples X:

$$\frac{f(X)}{p(X)} = c \tag{2.12}$$

In practice, we cannot choose such a density function p since the required constant c is the value we are trying to compute. However, variance can be reduced by choosing a density function which has a shape similar to f. In physically based renderers, density functions are carefully implemented by considering the position of light sources and the reflectance of materials.

2.4 Path Tracing

2.4.1 Random Walk

Using the measurement Eq. 2.4, we can compute a pixel by integrating radiance coming from all directions. To compute radiance in a given direction, we can trace a light ray in this direction until an object is reached and compute the reflected radiance using the rendering Eq. 2.5. However, this equation requires to integrate radiance coming from all directions. This means that we have to trace many rays (for all these directions) and that we have to repeat this process recursively each time one of these rays reaches an object (i.e., tracing new supplementary rays). This naive approach has a huge memory cost and is unfeasible in practice.

The basic idea of the path-tracing algorithm is to randomly sample only one direction for evaluating the rendering equation. Thus, we can sample a path $x_1, \ldots, x_k$ from the camera to a light source and compute the contribution of this path to the pixel value (see Fig. 2.7). This can be seen as a random walk, which means we can estimate the value of a pixel by randomly sampling many paths X_i and by computing the mean value of the contributions:

$$f(X_i) = W_e(x_1, x_2) \left[\prod_{k=2}^{K-1} f_s(x_{k+1}, x_k, x_{k-1}) G(x_k, x_{k-1}) \right] L_e(x_K, x_{K-1}) G(x_K, x_{K-1}) \quad (2.13)$$

2.4.2 The Path-Tracing Algorithm

The path-tracing algorithm has been proposed by James T. Kajiya in Kajiya (1986). This algorithm implements a random walk for solving the rendering equation. It is currently used in many physically based renderers.

A pseudo-code implementation of path tracing is given in Algorithm 1. As explained previously, the algorithm computes each pixel by randomly sampling paths and computing the mean contribution of the paths for the pixel.

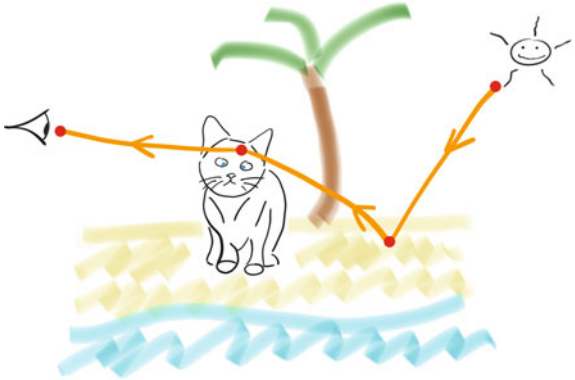


Fig. 2.7 A path (for example $X_i = x_1, x_2, x_3, x_4$) models the light transport from a light source (x_1) to a camera (x_4) after reflection in the scene (x_2 and x_3). The contribution of the path can be computed by developing the rendering equation and the measurement equation: $F(X_i) = L_e(x_1 \rightarrow x_2)G(x_1 \leftrightarrow x_2)f_s(x_1 \rightarrow x_2 \rightarrow x_3)G(x_2 \leftrightarrow x_3)f_s(x_2 \rightarrow x_3 \rightarrow x_4)G(x_3 \leftrightarrow x_4)W_e(x_3 \rightarrow x_4)$

Algorithm 1 : Path Tracing (using N paths per pixel and a probability density function p)

for all pixels in the image **do**
  $I_N \leftarrow 0$ {initialize the computed intensity of the pixel}
  for $i \leftarrow 1$ **to** N **do**
    sample a point x_1 in the pixel
    $P_i \leftarrow p(x_1)$ {initialize the probability of the path}
    $F_i \leftarrow 1$ {initialize the contribution of the path}
    loop
      sample a reflected direction and compute the corresponding point x_k in the scene
      $P_i \leftarrow P_i \times p(x_k, x_{k-1})$
      if x_k is on a light source **then**
        exit loop
      else
        $F_i \leftarrow F_i \times f_s(x_{k+1}, x_k, x_{k-1})G(x_k, x_{k-1})$
      end if
    end loop
    $F_i \leftarrow X_i \times W_e(x_1, x_2)L_e(x_K, x_{K-1})G(x_K, x_{K-1})$
    $I_N \leftarrow I_N + \frac{X_i}{NP_i}$
  end for
  pixel $\leftarrow I_N$
end for

2.4.3 Global Illumination

Algorithm 1 is a straightforward but inefficient implementation of path tracing and can be improved in many ways. A major source of inefficiency stands in the fact that reflected directions are sampled independently from light sources. Indeed, a light source which directly lights a point of the scene is easy to compute and contributes probably greatly to the illumination of the point. On the contrary, light coming

(a) Full lighting **(b)** Direct lighting **(c)** Indirect lighting

Fig. 2.8 Global illumination (**a**) of a scene can be decomposed in direct lighting (**b**) and indirect lighting (**c**). The direct lighting is the light coming from a source to a point and reflected toward the camera. The indirect lighting is the light coming from a source and reflected several times before reaching the camera

indirectly from a source, after several reflections on objects, is difficult to compute and may contribute little to the illumination of the point (see Fig. 2.8) (Nayar et al. 2006).

Thus, a very common optimization implemented in path tracers consists in sampling light sources directly (Vorba and Křivánek 2016): at each intersection point, a ray is sent toward a light source to estimate direct lighting and the path is traced recursively by sampling directions to estimate indirect lighting. This amounts to partitioning the integration domain in the rendering equation, which still gives valid results while improving the convergence speed.

2.5 Conclusion

An image, captured by a camera or seen by the Human Visual System (HVS), is a measure of the light (radiance) propagated in the scene. Photo-realistic image synthesis consists in computing an image from a virtual 3D scene, using physical laws of light transport such as the rendering equation. Current rendering algorithms are based on stochastic methods (Monte Carlo integration, Markov chain) to compute realistic images. Such an algorithm gradually converges to the expected image of the virtual scene but this generally requires a lot of computation time. Many improvements have been proposed to speed up the convergence of the rendering algorithms. The remaining of this book aims to characterize the noise present in rendered images (resulting variance of the rendering algorithm).

References

Catmull EE (1974) A subdivision algorithm for computer display of curved surfaces. Ph.D. thesis, The University of Utah

Cook RL, Carpenter L, Catmull E (1987) The reyes image rendering architecture. In: Proceedings of the 14th annual conference on computer graphics and interactive techniques, SIGGRAPH '87, pp 95–102

Glassner AS (1994) Principles of digital image synthesis. Morgan Kaufmann Publishers Inc., San Francisco

Goral CM, Torrance KE, Greenberg DP, Battaile B (1984) Modeling the interaction of light between diffuse surfaces. In: Proceedings of the 11th annual conference on computer graphics and interactive techniques, SIGGRAPH '84, pp 213–222

Jakob W (2013) Light transport on path-space manifolds. Ph.D. thesis, Cornell University

Jensen HW (2001) Realistic image synthesis using photon mapping. A. K. Peters Ltd., Natick

Kajiya J (1986) The rendering equation. ACM Comput Graph 20(4):143–150

Lafortune EP, Willems YD (1993) Bi-directional path tracing. In: Proceedings of third international conference on computational graphics and visualization techniques (compugraphics '93), Alvor, Portugal, pp 145–153

Nayar SK, Krishnan G, Grossberg MD, Raskar R (2006) Fast separation of direct and global components of a scene using high frequency illumination. ACM Trans Graph 25(3):935–944

Nicodemus FE, Richmond JC, Hsia JJ, Ginsberg IW, Limperis T (1977) Geometric considerations and nomenclature for reflectance. National Bureau of Standards

Pharr M, Humphreys G (2010) Physically based rendering: from theory to implementation, 2nd edn. Morgan Kaufmann Publishers Inc., San Francisco

Veach E (1997) Robust Monte Carlo methods for light transport simulation. Ph.D. thesis, Stanford University

Veach E, Guibas LJ (1994) Bidirectional estimators for light transport. In: Eurographics rendering workshop, pp 147–162

Veach E, Guibas LJ (1997) Metropolis light transport. Comput Graph 31(Annual Conference Series):65–76

Vorba J, Křivánek J (2016) Adjoint-driven russian roulette and splitting in light transport simulation. ACM Trans Graph 35(4):1–11

Whitted T (1980) An improved illumination model for shaded display. Commun ACM 23(6):343–349

Chapter 3
Visual Impact of Rendering on Image Quality

3.1 Introduction

To generate photo-realistic images, modern renderers generally use stochastic algorithms such as the path-tracing algorithm. These algorithms can produce high-quality images but may require a long computation time. Therefore, rendering is generally stopped after a given amount of time and the output image may not be fully converged. In this case, the resulting variance can be seen as noise.

In this chapter, we present the impact of several parameters on the visual quality of the rendered images. More specifically, we study some parameters of the rendering algorithm (path length, number of paths) and the scene complexity (light source, scene geometry, materials). The images presented in this chapter were rendered using the path-tracing algorithm implemented in PBRT (Pharr and Humphreys 2010), a well-known open-source renderer developed for research purposes. Two very classic scenes are studied here, the *Cornell box* scene (Goral et al. 1984) and the *glass egg* scene (Veach and Guibas 1995) (see Fig. 3.1).

3.2 Influence of Rendering Parameters

3.2.1 Path Length

Basically, the path-tracing algorithm traces a path in the scene until a light source is reached. High-length paths are, therefore, costly to compute but they are necessary to compute complex lighting.

Since reflected radiance is smaller than incident radiance, each reflection step reduces the potential contribution of the path so the potential contribution of high-length paths to the final pixel value is generally negligible. Renderers generally set a maximum path length to avoid computing costly high-length paths but this introduces

A. Bigand et al., *Image Quality Assessment of Computer-generated Images*,
SpringerBriefs in Computer Science, https://doi.org/10.1007/978-3-319-73543-6_3

(a) The*Cornell box* scene (diffuse materials + one area light source) is a classic scene for testing rendering with global illumination and color bleeding. Rendered with: max path length = 16, number of paths per pixel = 1024.

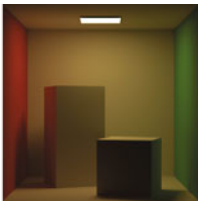
(b) The *glass egg* scene (diffuse materials + one specular object + directional light sources) is a classic scene for testing rendering with global illumination and light caustics. Rendered with: max path length = 32, number of paths per pixel = 524 288.

Fig. 3.1 Reference scenes studied in this chapter, rendered using the PBRT path tracer

(a) Max path length: 1, rendering time: 14min

(b) Max path length: 2, rendering time: 41min

(c) Max path length: 4, rendering time: 72min

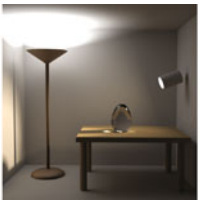


(d) Max path length: 8, rendering time: 86min

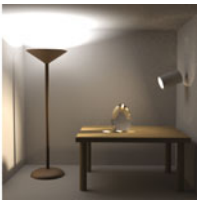
Fig. 3.2 Impact of maximum path length for rendering the *Cornell box* scene. High-length paths are necessary to render complex lighting but increase computation time

(a) Max path length: 1, rendering time: 6h

(b) Max path length: 2, rendering time: 18h



(c) Max path length: 4, rendering time: 40h



(d) Max path length: 8, rendering time: 45h

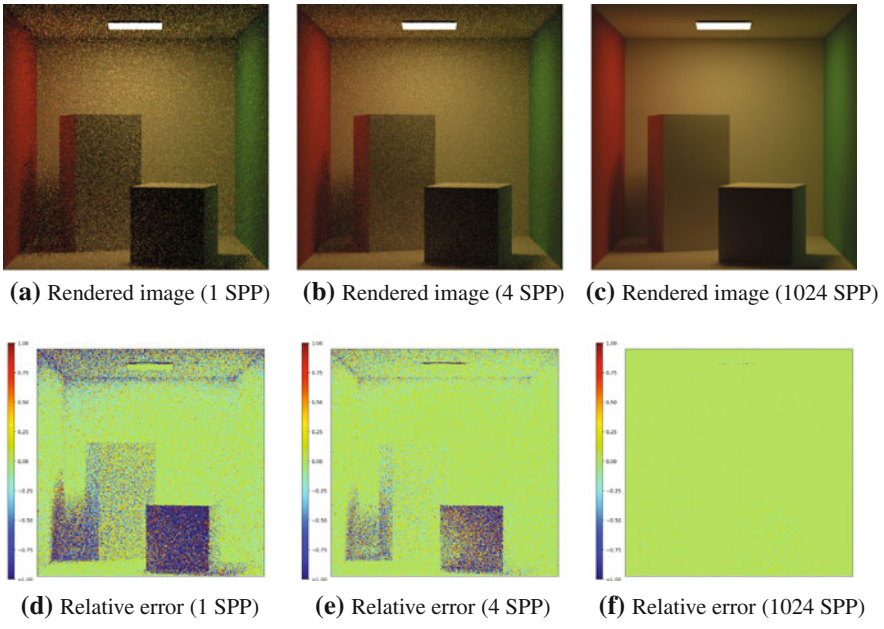
Fig. 3.3 Impact of maximum path length for rendering the *glass egg* scene. Light transport is quite complex in this scene, so the bias introduced by setting a low value for the maximum path length is clearly visible

a bias (see Figs. 3.2 and 3.3). To reduce this bias, renderers also implement Russian roulette (Arvo and Kirk 1990), an unbiased technique for stopping the tracing of a path with a probability proportional to the potential contribution of the path (a path with a low potential radiance is more likely to be stopped).

3.2.2 *Number of Path Samples*

Stochastic rendering algorithms, like path-tracing, sample many paths progressively and converge to the solution of the rendering equation. However, the convergence is slow so rendering a high-quality image requires a long computation time. Even worse, predicting how many path samples are necessary to compute a satisfactory image is difficult. In fact, determining if a rendered image converged is already a problem.

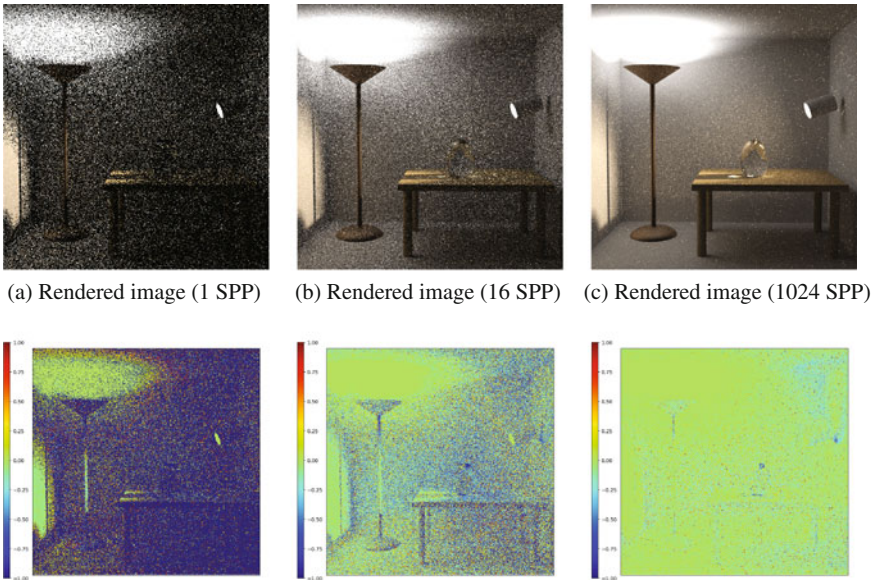
When an image has not fully converged, the resulting variance is visible in the form of noise (see Figs. 3.4 and 3.5). A common problem for renderer users is thus to



(a) Rendered image (1 SPP) **(b)** Rendered image (4 SPP) **(c)** Rendered image (1024 SPP)

(d) Relative error (1 SPP) **(e)** Relative error (4 SPP) **(f)** Relative error (1024 SPP)

Fig. 3.4 Impact of the number of samples per pixel (SPP) on the rendered image for the *Cornell box* scene. When only a few SPP are computed the rendered image is very noisy but this noise is reduced when more samples are computed. We may notice that noise is mainly present in indirectly lit area of the scene, since renderers compute direct lighting separately (as explained in Sect. 2.4.3). Legends for relative error: in blue, the rendered pixel is too dark; in red, the rendered pixel is too bright



(a) Rendered image (1 SPP) (b) Rendered image (16 SPP) (c) Rendered image (1024 SPP)

(d) Relative error (1 SPP) (e) Relative error (16 SPP) (f) Relative error (1024 SPP)

Fig. 3.5 Impact of the number of SPP on the rendered image for the *glass egg* scene. Lighting is complex in this scene so many path samples should be computed. Even with 1024 SPP, the noise is still important and some areas are still underestimated (for instance, highlights and caustics produced by the glass egg)

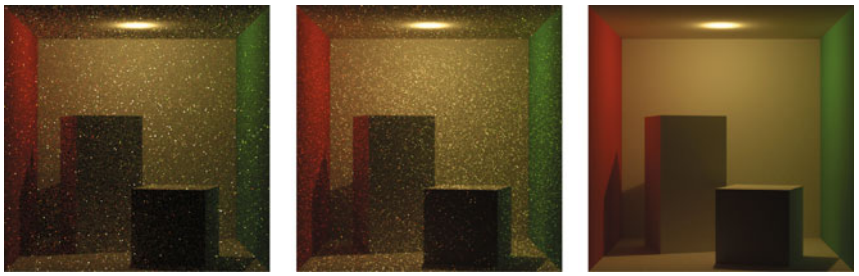
correctly set the number of Samples Per Pixel (SPP), which should be low enough for limiting computation time but high enough for limiting visible noise. In practice, the number of path samples to use for rendering a scene is determined by human experts and this number is generally intentionally overestimated for rendering high-quality images.

3.3 Influence of the Scene

3.3.1 Light Sources

Theoretically, any object can emit light, from any point on its surface and toward any directions. This is implemented with the emissive function L_e in the rendering Eq. (2.5). For most objects, the emissive function can be neglected (pure reflectors). The few objects which really emit light (the light sources) are considered separately to compute rendering more efficiently (using the direct lighting sampling technique explained in Sect. 2.4.3).

Renderers classically implement several light source models. An area light source is an object which emits light diffusely (uniformly in all directions) over its surface.



(a) 1 SPP **(b)** 1024 SPP **(c)** 131 072 SPP

Fig. 3.6 A point-light source is more difficult to renderer using the path-tracing algorithm, since this light model has no area. This results in a higher variance so the convergence is slower (131 072 SPP) than for the reference scene (1024 SPP, see Fig. 3.4)

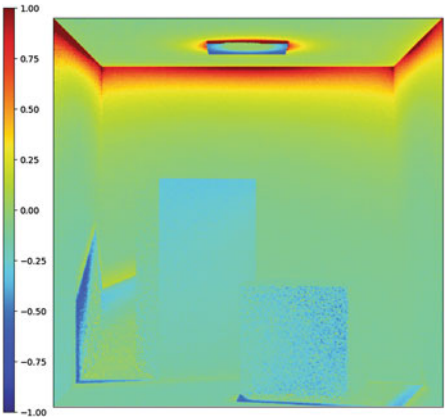


Fig. 3.7 Relative difference between rendered images when replacing the area light source with a point-light source, in the *Cornell box* scene. Legend: in blue, the modified scene is darker than the reference scene; in red, it is brighter

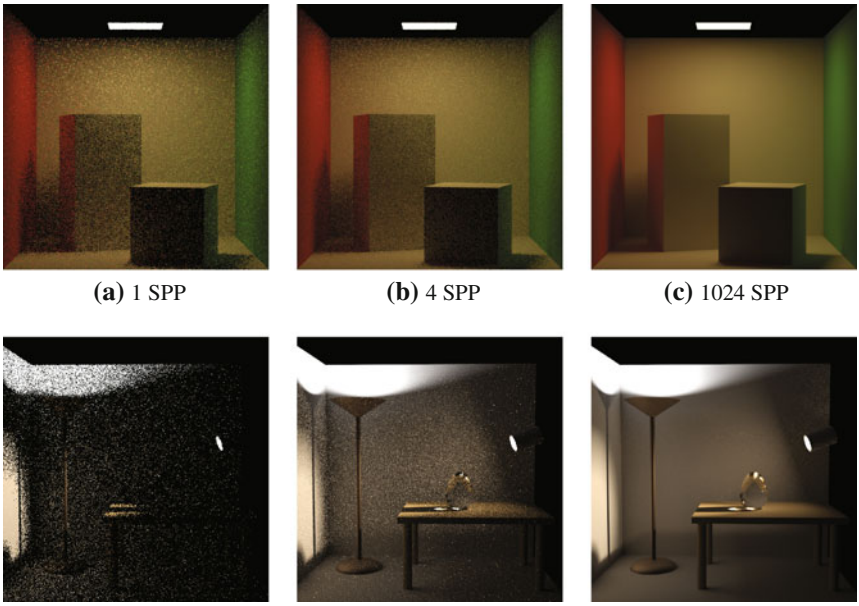
This is a general model which is very frequently used in physically based renderers. An omnidirectional point-light emits light from a point toward all directions uniformly. This model is a simplification of a small light source quite far away (for example a light bulb) and is often used in real-time renderers. A directional light source emits light in a single direction (from an infinitely far away position). This model is classically used to simulate sunlight.

Light models are part of the scene description and, therefore, impact the rendered image. Moreover, each light model should be implemented specifically in the rendering algorithm, which also impacts the rendered image. For example, replacing the area light source by a point-light source in the *Cornell box* scene changes soft shadows to hard shadows (see the shadows produced by the boxes in Figs. 3.1a, 3.6c, and 3.7). This also greatly impacts the visible noise in the rendered images: the area light source is easily found by path-tracing hence the fast convergence, whereas the point-light source is hard to find using path tracing (a point has no area) hence the slow convergence (see Fig. 3.6).

3.3.2 Scene Geometry

With physically based rendering, all objects can contribute to the global illumination of the scene, which means an object can influence the light reflected by another object in the rendered image (see Fig. 2.4.3). This is particularly important in a closed interior scene, since all the light emitted by light sources will finally reach the camera. On the contrary, in an opened outdoor scene most of the sunlight is reflected by objects toward the sky and will never reach the camera.

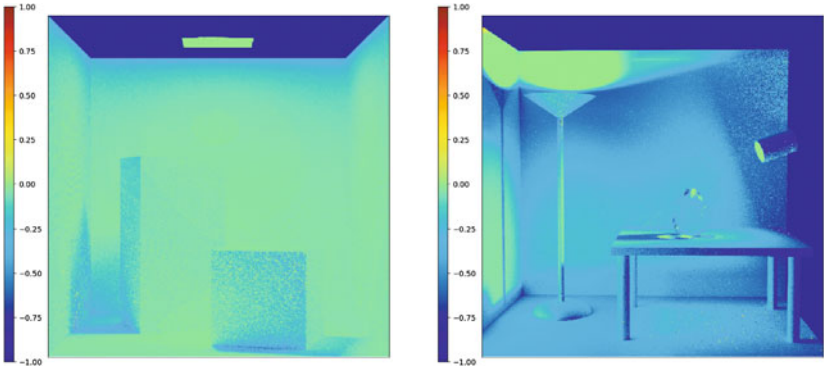
Thus, if we remove some parts of a scene (for instance, the ceiling or some walls of an interior scene), the scene is more opened and indirect lighting is less important in the rendered image since a large part of the emitted light can escape the scene (see Fig. 3.9). This escaping light can just be discarded for the rendering computation but the remaining indirect light still has to be computed so the corresponding variance is not significantly reduced (see Fig. 3.8).



(a) 1 SPP **(b)** 4 SPP **(c)** 1024 SPP

(d) 1 SPP **(e)** 1024 SPP **(f)** 524 288 SPP

Fig. 3.8 Impact of the geometry on rendered images. Top row: removing the ceiling in the *Cornell box* scene. Bottom row: removing the ceiling and the right wall in the *glass egg* scene. Making a scene more opened has no significant impact on rendering convergence (rendering requires the same number of SPP)



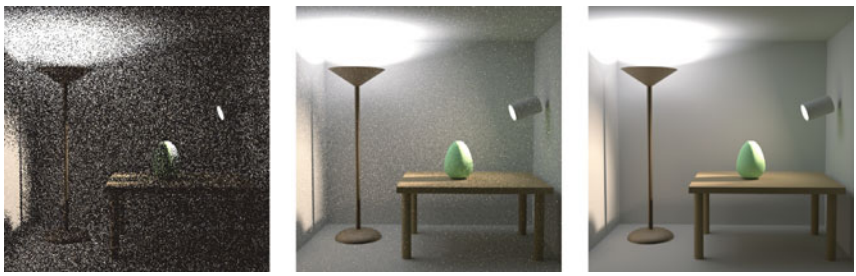
(a) The *Cornell box* scene when removing the ceiling. Indirect lighting in the shadows of the boxes is reduced but it is still important since the scene is still mostly closed.

(b) The *glass egg* scene when removing the ceiling and a wall. The indirect lighting is greatly reduces here since the scene is now quite opened and since the ceiling greatly contributes to global illumination in the original scene.

Fig. 3.9 The relative difference between rendered images when removing some objects of the scene

3.3.3 Materials

A material describes how an object reflects (or transmits) light. Real materials can be very complex so renderers generally propose a few models, which can be combined to describe more complex materials. Material models are generally an intermediate



(a) 1 SPP *(b) 1024 SPP* *(c) 524 288 SPP*

Fig. 3.10 Material models greatly impact the rendered image and may impact the convergence. With a diffuse egg, the color-bleeding on the right-side wall converges slowly. In the original scene, the corresponding caustics are also slow to converge but the small caustics on the table converge fast

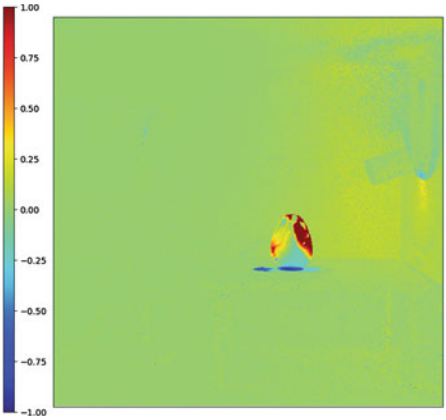


Fig. 3.11 The relative difference between rendered images when using a diffuse material in the *glass egg* scene. The specular highlights on the egg is replaced by a smooth diffuse highlight and the caustics on the table is replaced by color-bleeding on the right-side wall

between two particular models: the specular model and the diffuse model. The specular model reflects light in one specific direction (e.g., mirrors) and the diffuse model reflects light in all directions uniformly (e.g., rough materials).

Materials greatly contribute to global illumination. A diffuse material tends to smoothly spread the color of the object onto neighboring objects (color-bleeding). A specular material tends to show highlights on the surface of the object and to produce sharp caustics onto neighboring objects (see Figs. 3.11 and 3.10). If the caustics produced by specular objects have a limited area, they may be faster to converge than large color-bleeding produced by diffuse objects.

3.4 Conclusion

Stochastic rendering algorithms, such as path-tracing, generate theoretically correct images with an infinite number of paths of a potentially infinite length, i.e., an infinite rendering time. To render an image in a finite time, the number of paths and the maximum path length are two parameters that the user has to set before launching the rendering. As a result, the missing computation can be seen in the final image as rendering noise (high-frequency "grainy" variations) and artifacts (black or under-illuminated areas). Limiting the number of paths introduces a remaining variance, which can be seen as rendering noise in the final image. Limiting the maximum path length introduces a bias, which can be seen as artifacts in the final image. These problems are more important in scenes with complex lighting conditions such as multiple reflection (close scene), caustics (specular materials), small light sources, etc.

To analyze the quality of the final image, it would be interesting to have a model of the noise introduced by a rendering algorithm. However, such a model can be difficult to obtain and is specific to the considered algorithm whereas many algorithms are implemented in modern renderers. In the remaining of this book, we use some

machine learning techniques to characterize the visual image quality of a synthetic image containing rendering noise. We study several approaches, assuming we have the full reference image, a reduced reference, or no reference.

References

Arvo J, Kirk D (1990) Particle transport and image synthesis. SIGGRAPH Comput Graph 24(4):63–66

Goral CM, Torrance KE, Greenberg DP, Battaile B (1984) Modeling the interaction of light between diffuse surfaces. In: Proceedings of the 11th annual conference on computer graphics and interactive techniques, SIGGRAPH '84, pp 213–222

Pharr M, Humphreys G (2010) Physically based rendering: from theory to implementation, 2nd edn. Morgan Kaufmann Publishers Inc., San Francisco

Veach E, Guibas L (1995) Optimally combining sampling techniques for monte carlo rendering. In: Proceedings of the 22nd annual conference on computer graphics and interactive techniques, SIGGRAPH '95, pp 419–428

Chapter 4
Full-Reference Methods and Machine Learning

This chapter introduces the application of machine learning to Image Quality Assessment (IQA) in the case of computer-generated images. The classical learning machines, like SVMs, are quickly remained and RVMs are presented to deal with this particular IQA case (noise features learning). A recently performed psycho-visual experiment provides psycho-visual scores on some synthetic images (learning database), and comprehensive testing demonstrates the good consistency between these scores and the quality measures we obtain. The proposed measure has also been compared with close methods like RBFs, MLPs, and SVMs and gives satisfactory performance.

4.1 Image Quality Metrics Using Machine Learning Methods

Image quality measures are very important to characterize the visual quality of an image. For example, they are of great interest for image compression (JPEG models) and natural image characterization (Lahoudou et al. 2010). The Image Quality Assessment (IQA) can be provided by subjective and objective evaluation. Subjective evaluation such as Mean Opinion Score (MOS) is truly definitive but is time consuming and expensive. Objective evaluation is defined by mathematical definitions such as the well-known metrics Mean Squared Error (MSE), Mean Absolute Error (MAE), Peak Signal-to-Noise Ratio (PSNR), and Structural SIMilarity (SSIM). Generally, objective evaluation metrics are based on the pixel-to-pixel difference between two compared images and are easy to compute with low-computational complexity. These metrics are used in the remaining of this chapter but they are not very well matched to visual quality perception so other models have been provided; for example, learning machines were adopted to compute MOS.

A. Bigand et al., *Image Quality Assessment of Computer-generated Images*,
SpringerBriefs in Computer Science, https://doi.org/10.1007/978-3-319-73543-6_4

As presented in the introduction, image quality assessment models are usually classified into three model families in the literature. Considering machine learning, full-reference models (and reduced-reference models) are linked to the supervised (or semi-supervised) learning process of learning machines. In supervised learning, the machine is defined by its inputs and its corresponding desired outputs (labels). In semi-supervised learning, a reduced database is used; this will be presented in Chap. 5. Finally, no-reference models evaluate the quality of images without access to reference images (i.e., unsupervised learning, where no labels are provided to the learning task). This method has been used with some success (Delepoulle et al. 2012), and will be presented in Chap. 6.

The method for establishing IQA using machine learning can be summarized as follows:

- First step: Prepare a subjective database
- Second step: Set noise features
- Third step: Compute these features for the next stage of the process
- Fourth step: Predict visual image quality Q with the noise features (Q = quality = f(noise features)). Compare the predicted results to the normal MOS to set up the learning machines (ANNs, RBFs, SVMs, RVMs)
- Fifth step: Evaluate the model's performance.

As previously presented, the main goal of global illumination methods is to produce *synthetic images with photorealistic quality*. They are generally based on the Path Tracing method (Kajiya 1986), where stochastic paths are generated from the camera point of view towards the 3D scene. Since the paths are randomly chosen, the light contribution can change greatly from one path to another, which generates high-frequency color variations in the rendered image (Shirley et al. 1996). The Monte Carlo theory, however, ensures that this process will converge to the correct image when the number of samples (the paths) grows. But no information is available about the number of samples that are really required for generating a *visually* satisfactory image. Consequently and due to the high-computational cost of global illumination algorithms, perception-driven approaches were proposed. The main idea of such approaches is to replace the human observer by a vision model. By mimicking the HVS, such techniques can provide important improvements for rendering. They can be used for driving rendering algorithms to visually satisfactory images and to focus on visually important features (Mitchell 1987; Farrugia and Péroche 2004; Longhurst et al. 2006). In Cui et al. (2012), a summary on the research of image quality assessment methods is proposed when using an artificial neural network.

This chapter focuses on the use of a new perceptual index to replace psychovisual index in the perception-driven model. We present a full-reference image quality metric, based on Relevance Vector Machine (RVM), to model the uncertainty brought by noises affecting the image synthesis. In Tipping (2003), Tipping studies RVM and introduces the principles of Bayesian inference in a machine learning context, with a particular emphasis on the importance of marginalization for dealing with uncertainty. RVM is applicable for regression and classification and provides sparse Bayesian models interesting in image processing. Here, we focus on grayscale images

and use RVM for image quality evaluation (in fact, we use the "L" component of "Lab" color images, since noise only affects the "L" component (Carnet et al. 2008a)). In this chapter, we first describe the experimental setup we use, then we present some feature extraction methods and image quality metrics based on SVM and RVM.

4.2 Experimental Setup

Unbiased Global Illumination (G.I.) methods use randomly chosen paths for sampling the illumination of visible objects. This process generates *stochastic noise*, which is perceptible by any human observer. Image denoising techniques, used *a posteriori*, are largely present in the literature (Heinonen and Neuvo 1987; Arakawa 1996; Bigand and Colot 2010). Noise models and noise estimation from images are, however, more difficult. Anyway, these models are based on theoretical models of noise like additive white noise, procedural noise functions (Lagae et al. 2010), etc. However in G.I. algorithms, noise is not additive and arises from an unknown random distribution function. To our knowledge, there is no existing model able to detect and to quantify stochastic visible noise in an image. We detail in the following the different steps of our approach for solving this problem by using a new image quality measure.

4.2.1 Overview

Our goal is to mimic the human visual detection of noise by way of a full-reference image quality measure. So, it is necessary to provide to the proposed model some examples of what human judgment consider to be noisy images or noiseless ones. After validation on all these examples, the method will generate a model (function $y = Q = f(x)$, x is the noise feature vector and y is IQA) that will then be used on images that have to be analyzed.

4.2.2 Data Acquisition

The model is built on data corresponding to images of globally illuminated scenes. We used (as a first approach) a Path Tracing with next event algorithm (Shirley et al. 1996), which computes several images from the same point of view by adding successively N new samples[1] equally for each pixel. For each scene and each point of view, we thus have several images available, the first ones being strongly noisy and the last ones being converged. The images were computed at 512×512 resolution,

[1] In the following, we will call sample a stochastic path between the viewpoint and a light source.

Fig. 4.1 Two used reference scenes (Bar, Deskroom)

the number of additional samples between two successive images was $N = 100$ and 12 scenes were used. The largest number of samples per pixel was set at 10.000 which appeared to be sufficient for generating visually converged images. Figure 4.1 presents 2 of these scenes that were used during the validation stage of the model. These scenes highlight different illuminations and various geometrical and textures complexities.

4.2.3 Psycho-visual Scores Acquisition

Because we have to evaluate the noise level present in each generated image, some experiments were necessary in order to specify the noise threshold that is used as stopping criterion in images synthesis. But considering the entire image for noise thresholding has two main drawbacks: on one hand, it requires evaluation methods to work on very large datasets; this has been experimentally shown to reduce their learning efficiency. On the other hand, the noise is generally not equally perceived by human observers through any part of an image; noise thresholds are, thus, different for each location in each image and the use of a global threshold would reduce the efficiency of the approach by requiring the same number of samples to be computed for each pixel of the image. We, thus, defined a very simple protocol in which pairs of images are presented to the observer. One of this image is called the *reference image* and has been computed with $N_r = 10.000$ samples per pixel. The second image, so-called the *test image*, is built as a stack of images, from very noisy ones above to converged ones below: by calling N_i the number of samples in the stack's image i, with $i = 100$ at the top of the stack and $i = max$ at its bottom, we, thus, ensure the property $\forall i \in [100, max[, N_i < N_{i+100} \leq N_r$. Each of these images are opaque and virtually cut into nonoverlapping blocks of size 128×128 over the entire image. For the used 512×512 test images we thus get 16 different blocks (clockwise sorted from 0 to 15, top image of Fig. 4.8) for each of the stack's images. During

the experiments, the observer is asked to modify the quality of the noisy image by pointing the areas where differences are perceived between the current image and its reference one. Each point-and-click operation then causes the selection and the display of the corresponding $i + 100$ level block thus visually reducing noise in this image's subpart. This operation is done until the observer considers that the two images are visually identical. Note that for reducing experiment artifacts this operation is reversible meaning that an observer is able to go down or up into the images stack. The pair of images that is presented to the observer is chosen randomly but we ensure that each pair will be presented two times. Obviously, the block grid is not visible and all the observers worked in the same conditions (same display with identical luminance tuning, same illumination conditions, etc.). The results were recorded for 33 different observers and we computed the average number of samples $\tilde{N}$ that are required for each subimage to be perceived as identical to the reference one by 95% of the observers. We got experimentally $\tilde{N} \in [1441, 6631]$ with often large differences between sub-images of the same image (see Fig. 4.2 for some images we used). So, now we present attempts of automatic image quality index we propose.

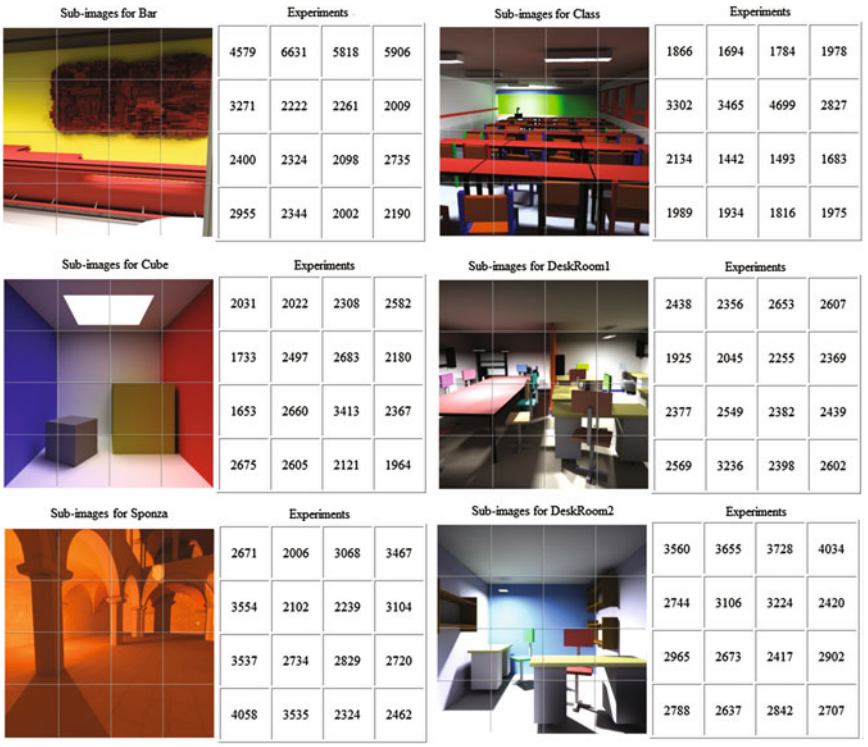


Fig. 4.2 The thresholds obtained for some used scenes

4.3 Noise Features Extraction

4.3.1 Classical Strategies

Feature generation is the iterative process of finding a subset of only pertinent features by removing redundant or interfering ones. It makes it possible to go from a high dimension of images (number of pixels) to a lower dimension (feature vector). Training algorithm after feature generation will be easier and better performances in time and space complexities would be achieved. In this section, we discuss methodologies to extract image noise quality indexes in order to detect stochastic noise in photo-realistic images.

In order to get the best characterization of noise as possible, we convert each image into the frequencies domain using a **noise mask**. The noise mask is a denoising technique commonly used in satellite imagery in order to allows pictures to have enhanced details and to reduce the image noise (Kumar 2014). Since noise only affects L component of Lab color images (Carnet et al. 2008b), first a blurred image is computed by applying to L linear filtering with averaging filters of sizes 3×3 and 5×5 (Makandar and Halalli 2015), linear filtering with Gaussian filters of same sizes and with standard deviations $\sigma \in \{0.5, 1, 1.5\}$, median filters (Dawood et al. 2012) and adaptive Wiener filters (Biswas et al. 2015) of same sizes. The obtained image is also denoised via wavelet analysis since the wavelet coefficients do not follow a Gaussian distribution (Gao et al. 2015; Chang et al. 2000; Gou et al. 2009; Fernandez-Maloigne et al. 2012). Next, the obtained image is denoised via Wavelet analysis. After one stage 2D wavelet decomposition, an input image is decomposed into four sub-bands namely low–low (LL), low–High (LH), high–low (HL), and high–high (HH) sub-bands (Fig. 4.3).

Among these four sub-bands, the LL sub-band contains low-frequency components, while the others three are the high-frequency components. The final image is reconstructed by extracting noise from these components using MATLAB Wavelet Toolbox. We obtain then 13 denoised versions for the image (Constantin et al. 2015), namely L_D in the following. The mean value and the standard deviation are applied to the different denoised versions of the image in order to obtain a total of 26 noise features used as inputs to the learning model. Given an image L, the estimated image

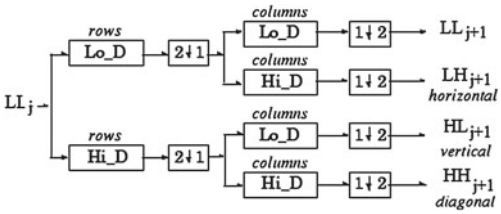


Fig. 4.3 One stage of the 2D-wavelet transform

noise at the pixel location (i, j) is obtained by a pixel wise subtraction between the current image pixel and the denoised one L_D:

$$e(i, j) = |L(i, j) - L_D(i, j)|. \tag{4.1}$$

After some experiments, we find that the mean and the standard deviation give the most significant results from the set of noise quality indexes defined in (Lahoudou et al. 2010, 2011):

$$f^{(1)}(L) = \frac{1}{V \times W} \sum_{i=1}^{V} \sum_{j=1}^{W} e(i, j) \tag{4.2}$$

$$f^{(2)}(L) = \left(\frac{1}{V \times W} \sum_{i=1}^{V} \sum_{j=1}^{W} (e(i, j) - f^{(1)}(L))^2 \right)^{1/2}, \tag{4.3}$$

where $V \times W$ is the dimension of the image matrix L. We then obtain a total of 26 features used as inputs to the learning model. The implementation of the learning dataset in case of feature generation is given by Algorithm 2.

Algorithm 2 *Learning data-set algorithm for noise feature generation*

1: ***Requirement:*** *a set S of scenes where each scene is divided in* 16 *sub-images.*
2: *Read a sub-image* I_{ij} *from the set S of scenes where* $i = 1, 2, ..., P$, $j = 1, 2, ...Q$; *P denotes the number of training scenes and Q denotes the number of sub-images in a scene (In our case* $P = 101$ *and* $Q = 16$*).*
3: *Convert the RGB color sub-image to Lab color sub-image and extract the luminance vector L which has a size of* 128×128 *pixels for each sub-image.*
4: *Extract the* 26 *noise feature vector from the luminance component L.*
5: *Repeat steps 2-4 for all the training sub-images.*
6: *Save the feature vectors with the desired values calculated based on HVS thresholds on the learning data-set.*

4.3.2 Pooling Strategies and Deep Learning Process

In the previous section, we tried to identify the best subset of features to represent noise features in synthetic images. However, these noise features depend on change with different kind of images (specular images, diffuse,etc.) and features extraction techniques. Another way to automatically obtain noise features from images consists of using the deep learning paradigm. In the present section, we propose the use of deep learning (DL) for reverse engineering of noise features that are learned directly from the raw representation of input synthetic images using Convolutional Neural Networks (CNN). It has been used in Lee et al. (2017) to learn leaf features for plant

classification. In the same way, the use of deep learning has been investigated for blind image quality assessment (Bianco et al. 2016).

The CNN model for selecting features of noisy images is based on the model proposed in Krizhevsky et al. (2012). Rather than training a new CNN architecture, we reused the pretrained network, because it is now widely admitted that features extracted from the activation of a CNN trained in a fully supervised manner in large-scale object recognition studies can be repurposed for a novel generic task (Donahue et al. 2014).

Different standard databases are available to test the algorithm's performance with respect to the human subjective judgments. However, image captured using camera devices are usually afflicted by stochastic mixtures of multiples distortions which are not well modeled by synthetic distortions found in existing databases (Liu et al. 2014; Virtanen et al. 2015). In case of scenes with 512 × 512 resolution as in the previous section, the scenes Bar, Class, and Cube consist of diffuse surfaces with complex geometry (Fig. 4.2). In the Deskroom1, DeskRoom2, and Sponza scenes, objects with different shapes and material properties are placed together presenting rich and complicated shading variations under different lighting and viewing conditions. We also used more complex scenes with 800 × 800 resolution (see Figs. 4.4 and 4.5) to improve the learning diversity. The corresponding parameters are presented Table 4.2.

The scene Chess (Fig. 4.4a) includes rich inter-reflection effects between the diffuse and specular surfaces, such as color bleeding and caustics. The new scene Kitchen (Fig. 4.4b) is used to illustrate view-dependent indirect illumination effects caused by strong inter-reflections between specular surfaces. We have also tested the scalability of our method on the four complex scenes Pnd (Fig. 4.4c), BathRoom1, BathRoom2, and DeskRoom (Fig. 4.5a, b and c) which contain different geometrical and textural characteristics. The images are cut into 16 nonoverlapping blocks of sub-images of size 128 × 128 pixels for the scenes with 512 × 512 resolution. We set the maximum number of rays per pixel to 10100 in order to obtain non-distorted copies of the images. The images were generated by adding 100 rays for each pixel between two successive images using path tracing algorithm (Hedman et al. 2016).

Fig. 4.4 Three used scenes with 800 × 800 resolution

Fig. 4.5 Three other used scenes with 800 × 800 resolution

The labeling process selects images computed using diffuse and specular rendering and ask the observers for their qualities. The observers which are from different locations have faced the same display conditions (Constantin et al. 2015). The learning and the testing sets contain images with different percentages of homogeneous regions, edges, and exhibit different light effects. In order to separate the noise from signal, a sub-image is computed by applying to the luminance component different pooling strategies. The feature pooling is performed element by element on each version of the sub-image in a deep learning process using 13 layers. We use Averaging (A1–A2), Gaussian (G1–G6) (Makandar and Halalli 2015), Median (M1–M2) (Dawood et al. 2012), and Wiener (W1–W2) (Biswas et al. 2015) convolutions of depth and spread equal one. Next, the image is denoised using wavelet decomposition (Wav) (Gao et al. 2015) (see Table 4.1). The model is trained using Caffe framework (Jia et al. 2014). Next, the feature concatenation pooling is performed by concatenating the features vectors in a single longer feature vector in order to obtain a total of 26 noise features vector used as input to the perception model. Finally, the prediction pooling is performed for each sub-image of the scene (Table 4.2).

Table 4.1 Feature pooling architecture

Layers	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12	L13
Type	A1	A2	G1	G2	G3	G4	G5	G6	M1	M2	W1	W2	Wav
Size	3 × 3	5 × 5	3 × 3	5 × 5	3 × 3	5 × 5	3 × 3	5 × 5	3 × 3	5 × 5	3 × 3	5 × 5	
Standard deviation			0.5	0.5	1	1	1.5	1.5					
Padding	1	2	1	2	1	2	1	2	1	2	1	2	

Table 4.2 Number of rays per pixel between two successive images and largest number of rays per pixel for the scenes with 800 × 800 resolution

Scenes with 800 × 800 resolution	Number of rays between two successive images	Largest number two successive images
Kitchen	5	700
Chess	50	5000
BathRoom1	10	950
Pnd	50	5000
BathRoom2	10	950
DeskRoom	50	5000

4.4 Image Quality Metrics Based on Supervised Learning Machine

4.4.1 Support Vector Machines

In a previous work, we first applied Support Vector Machines (SVMs) to the perception of noise toward an automatic stopping criterion (Takouachet et al. 2007). Let us recall some basic and well-known properties of SVMs. SVM makes predictions based on the function $y(x)$ defined over the input space, and "learning" is the process of inferring this function (often linked to some parameters of the function, w is the normal vector to the hyperplane, Fig. 4.6). SVM is a non-probabilistic binary linear classifier, used to solve discrimination (or regression) problem, that is, to say decide if a new example belongs to one class or not. Learning process makes it possible the construction of the function $y = f(x)$ using a hyperplane and a (hard or soft) margin to separate learning samples into 2 classes, Class1 and Class2 (+1 or −1, Fig. 4.6). Consider the problem of separating the set of training vectors belonging to two separate classes:

$$D = \{(x_1, y_1), \ldots, (x_N, y_N)\}, x \in \Re^d, y \in \{-1, +1\}, \tag{4.4}$$

where N is the number of images in the learning set. The objective of SVM is to find the optimal hyperplane that can separate the two classes and meanwhile can maximize the margin which is defined as the distance between the nearest point and the hyperplane. It can be formulated as follows:

$$y(x; w) = f(x) = w^T x + b, \tag{4.5}$$

where w and b denote the weight vector and the bias respectively.

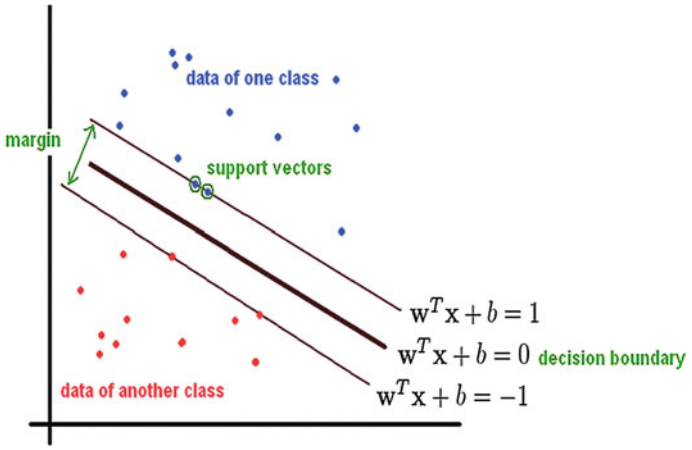


Fig. 4.6 Optimal hyperplane in the two class problem

In addition to performing linear classification, SVMs can efficiently perform a nonlinear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. Popular set of candidates for $y(x)$ is that of the form:

$$y(x; w) = \sum_{i=1}^{M} [w_i.\phi(x)] = \sum_{i=1}^{N} [w_i.K(x, x_i) + b], \tag{4.6}$$

where $\phi(x)$ are basic functions, w_i are weights (adjustable parameters). SVMs are built with basis functions $K(x, x_i)$, named kernel functions. The key feature of the SVM is that, in classification case, its target function attempts to minimize a measure of error on the training set while maximizing the "margin" between the two classes (in the feature space defined by the kernel). This is detailed in the following to introduce the particular case of RVMs. Learning methods like RBF (Radial Basis Function) neural network or multi-layer neural networks (MLP) are particular cases of SVM (they are built with different kernels). So, they will also be considered for comparisons in the following. These techniques were presented in the book (Xu et al. 2015) so they are not remained in this chapter.

Learning methods concern algorithms which are able to automatically improve their results over time. Our goal is to study and develop an artificial intelligent model, which should produce the same answer as a set of observers to the noise problem. That means that we want this model to be able to classify images as noisy or not. In this work, we study the use of SVMs to fit our classification problem. The main drawback of these approaches is that the data used during the learning stage have to be carefully chosen for the model to learn what is expected. Also, they require a huge number of nonredundant labeled images in order to train and to evaluate the perception models. For this reason, we use SVM inductive learning

to find the most informative noisy image for the learning algorithm and to reduce manual labeling workload. SVMs (Ren 2012) are part of a set of supervised learning methods for regression and classification problems. SVMs compute the hyperplane that can correctly separate the data and meanwhile can maximize the margin which is defined as the distance between the nearest point and the hyperplane. The training data that lie closest to the hyperplane are called support vectors. Generally, SVMs allow us to project the original training data to a higher dimensional feature space via Mercer Kernel operator K. We consider the set of classifiers of the form:

$$f(x) = w^T \phi(x) + b, \tag{4.7}$$

where w and b denote weight vector and bias respectively. $\phi(x)$ is a nonlinear mapping function that maps the training data into higher dimensional space, thus making them linearly separable. When $f(x) \geq 0$, we classify the image x as not affected by noise otherwise we classify the image x as noisy. The SVM finds an optimal hyperplane that can correctly separate the two classes of images and meanwhile can maximize the margin which is defined as the distance between the nearest point and the hyperplane. Then, SVM can be specifically stated as follows (Abe 2010):

$$min_{w,\xi_i} \frac{1}{2} w^T w + C \sum_{i=1}^{N} \xi_i \tag{4.8}$$

subject to

$$y_i(w^T \phi(x_i) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, \ldots N \tag{4.9}$$

where ξ_i is a positive variable giving a soft classification boundary. C is a positive regularization constant controlling the degree of penalization of the variable ξ_i. To solve the above optimization problem, we can adopt a Lagrange function in order to obtain the following dual optimization problem:

$$maximize \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \phi(x_i).\phi(x_j) \tag{4.10}$$

subject to

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \text{ and } 0 \leq \alpha_i \leq C, \forall i \tag{4.11}$$

where α_i is the Lagrange multiplier. During the training process, most of the α_i are zero and only the training data that lie on the margin surface or inside the margin area have nonzero α_i are called support vectors. To calculate $\phi(x_i).\phi(x_j)$, we can

use kernel function $K(x_i, x_j) = \phi(x_i).\phi(x_j)$ that verify Mercer's condition. For any test image x_i, its class label is determined through the following:

$$y(x) = sgn\left(\sum_{j=1}^{M} \alpha_i K(x, x_i) + b\right), \tag{4.12}$$

where M is the number of support vectors. However, SVMs are limited to small sizes of images, due to the great number of kernels (neurons) required (the number of support vectors required grows linearly with the size of the training size). Predictions made by SVMs are not probabilistic: this is crucial for classification applications where posterior probabilities of class membership are necessary to adapt to varying class priors and asymmetric misclassification costs (SVMs usually learn texture but not noise!). It is necessary to estimate the error/margin trade-off parameter C and the kernel function $K(x, x_i)$ must satisfy Mercer's condition. Tipping (Tipping 2004) have shown that RVM are interesting in sparse Bayesian learning. Particularly, the number of kernels of RVMs drastically decreases compared to SVMs. The obtained sparsity is interesting to investigate the proposed image quality model.

4.4.2 Relevance Vector Machines

4.4.2.1 Definition

The major motivation of RVM implementation is to remove the above limitations of SVMs. We now recall some results about RVM.

Given a dataset of input–target pairs $\{x_n, t_n\}_{n=1}^{N}$, the targets are samples from the model with additive noise:

$$t_n = y(x_n; w) + \varepsilon_n, \tag{4.13}$$

where ε_n are independent samples from some noise process assumed to be mean-zero Gaussian with variance σ^2. Due to the assumption of independence of the t_n, the likelihood of the complete dataset can be written as

$$p(T \mid W, \sigma^2) = \left(2\pi\sigma^2\right)^{-N/2} exp\left[-\frac{1}{2}\left(\frac{|| \, t - \Phi.W \, ||}{\sigma}\right)^2\right], \tag{4.14}$$

where $T = (t_1, t_2, ...t_N)^T$, $W = (w_0, w_1, ..., w_N)^T$, $\Phi = [\phi(x_1), \phi(x_2), ...\phi(x_N)]$ and $\phi(x_N) = [1, K(x_n, x_1), K(x_n, x_2), ..., K(x_n, x_n)]^T$.

A classical zero-mean Gaussian prior distribution over W is defined as follows:

$$p(W \mid \alpha) = \prod_{i=0}^{M} N(w_i \mid 0, \alpha_i^{-1}), \tag{4.15}$$

where N is a Gaussian distribution over w_i, and where α is the vector of hyperparameters. This prior distribution is favourable to weak weight models.

4.4.2.2 Sparse Bayesian Learning

Having defined the prior, Bayesian inference proceeds by computing the posterior over all unknowns given the data (from Bayes'rule, Tipping 2003):

$$\begin{aligned} & p(T \mid W, \alpha, \sigma^2) \\ & = \frac{p(T \mid W, \sigma^2) p(W \mid \alpha)}{p(T \mid \alpha, \sigma^2)} \\ & = (2\pi)^{-(N+1)/2} \mid \sum \mid^{-1/2} exp\left[-\frac{1}{2}(W-\mu)^T \sum^{-1} (W-\mu) \right] \end{aligned}, \tag{4.16}$$

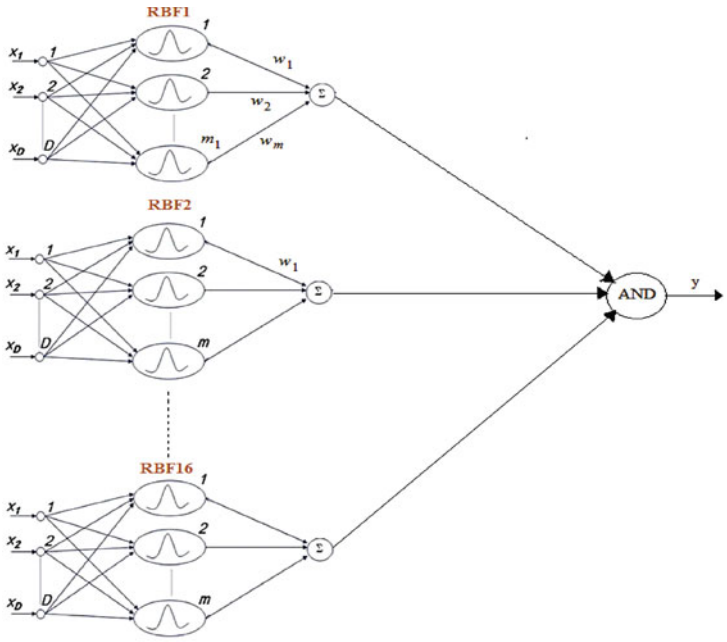
where the posterior covariance and mean are respectively:

$$\begin{aligned} \sum &= (\sigma^{-2} \Phi^T \Phi + A)^{-1} \\ \mu &= \sigma^{-2} \sum \Phi^T T \end{aligned}, \tag{4.17}$$

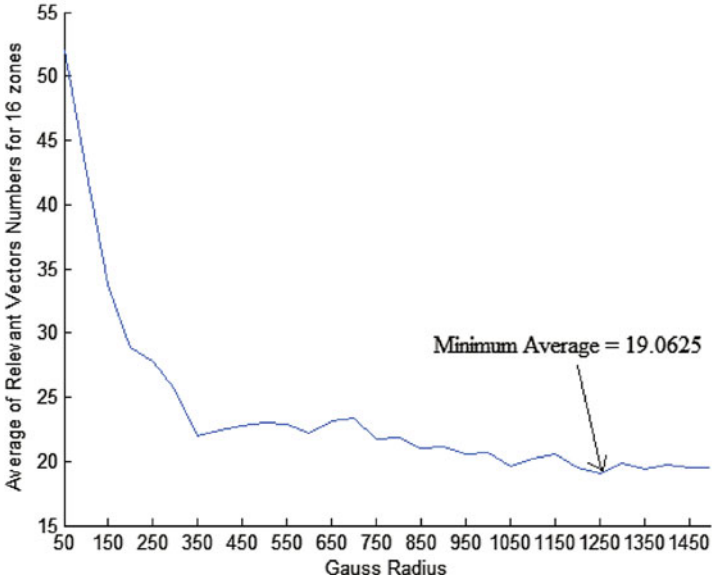
where $A = diag(\alpha_0, \alpha_1, ..., \alpha_N)$.

4.4.2.3 RVM Construction

The aim of the RVM consists in maximization of the marginal probability, that makes it possible the elimination of useless parameters. This property is illustrated in Fig. 4.7, using images of size 512×512, each image is divided in 16 blocks. During learning phase, the input of the network consists in the difference between the processed image and a noisy image (first image created during the synthesis process, with 100 samples). This procedure makes it possible the detection of noise in images, and do not use reference image (which is obviously unavailable). Processed images (60 different images were used) are examples obtained with different noise levels (different number of sample N_i) and un-noisy images. The output of the network is "-1" if the image is considered "noisy" or "-1" if image is considered non-affected with noise (human judgment). The network parameters are optimized using classical cross-validation.



(a) Complete structure of the proposed RVM



(b) Kernel width optimization for minimum relevant vectors

Fig. 4.7 RVM design

4.4.2.4 The Design of the RVM Image Quality Evaluation

We propose to use the previous defined RVM to extract a new noise-level index. The noise-level measure scheme is divided into two steps. In the first one, we perform images learning procedure applied to a block of the processed image I (obtained with Ni samples). Then, in a second step, we use this network to test the resulting RVM.

4.4.2.5 Algorithm

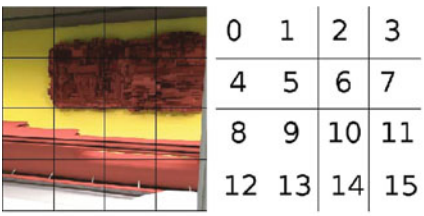
The implementation of image quality evaluation based on RVM is given by the following algorithm:

Algorithm 3 RVM Image quality measure

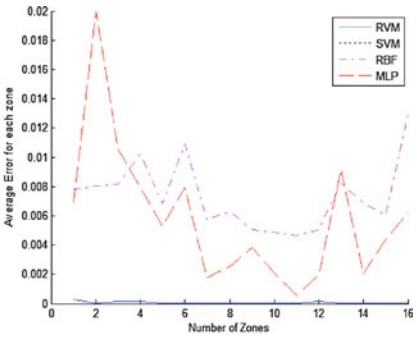
Require: a $M \times N$ gray-level image I, divided in 16 blocks and presented to neural network of Figure 4.6
1: Initialize the values α_i and σ^2
2: **if** Optimisation condition unsatisfied **then**
3: Find values of μ and $\sum$
4: Process values of γ and estimate new values of α and σ^2
5: **else**
6: Suppress weights associated to values $\mu = 0$
7: **end if**
8: Keep the vectors of the RVM associated wih values $\mu \neq 0$
9: Display result (image is noisy or not)

4.4.2.6 Experimental Results with a Synthetic Image

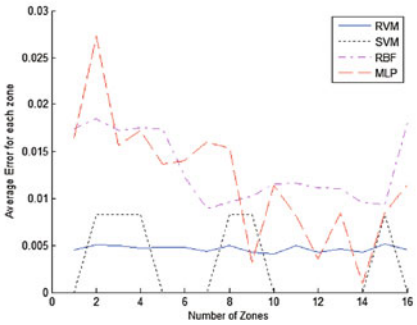
In order to test the performance of the proposed technique, some results obtained with the synthetic image named "Bar", Fig. 4.8, are shown in this section (other images were tested and same behaviors were observed, so they are not presented here due to the lack of space). This image is composed of homogeneous and noisy blocks and is interesting to present some results. We also present the results obtained with RBF, MLP, and SVM ("C" parameter optimized to 512) networks with the same image. The first figure represents the "Bar" image, the second one represents the learning average quadratic error, the third one the test average quadratic error, and the fourth one the number of relevant vectors, support vectors, or hidden neurons for the four implemented methods. These curves exhibit the advantage of using RVM for this task, where RVM performances outperform the other equivalent networks. These results have been generalized to a great variety of computer-generated images (see our website).



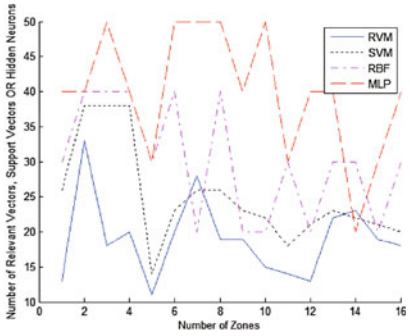
(a) Sub-image numbering



(b) Learning average quadratic error



(c) Test average quadratic error



(d) Number of relevant vectors, support vectors or hidden neurons for the four implemented methods

Fig. 4.8 Original image and compared results of different neural networks

4.5 Conclusion

The central idea of this chapter was to introduce the application of machine learning (and particularly RVMs), to take into account the uncertainty (noise) present at the image synthesis stage, and this idea seems to be very promising. Quality of synthetic images in testing stage is assessed with low error (less than 1%). RVM technique uses dramatically fewer basis functions than a comparable SVM while offering a number of additional advantages. These ones include the benefits of probabilistic predictions (noise detection), automatic estimation of "nuisance" parameters and the use of arbitrary basis functions (non-"Mercer" kernels) and a good generalization despite the small size of the learning database. The biggest advantage for this application is a sparse Bayesian learning that makes it possible to treat complete images with fewer relevant vectors than support vectors of SVMs. Noise features considered as inputs of neural networks (RVMs or others) are another key issue of the proposed method. We have compared conventional extracted features and deep learning extracted features:

the both give about the same (good) results. CNN models are highly dependent on the size and level of diversity of the training set. We think that the training data at our disposal are not sufficient at the moment, but CNN models have to be kept under the hand for the future. Thus we have shown that quality of computer-generated images can be predicted using machine learning associated to noise features. At the moment, we have tested reduced-reference models to better exploit the diversity of the training data at our disposal (next chapter) and a no-reference model (last chapter) for small training datasets. So, that the user can choose the most effective method for the considering problem he has to deal with.

References

Abe S (2010) Support vector machines for pattern classification. Springer Press, Berlin

Arakawa K (1996) Median filters based on fuzzy rules and its application to image restoration. Fuzzy Sets Syst 77:3–13

Bianco S, Celona L, Napoletano P, Schettini R (2016) Predicting image aesthetics with deep learning. In: Blanc-Talon J., Distante C., Philips W., Popescu D., Scheunders P (eds) Lecture Notes in Computer Science, vol 10016, pp 196–202

Bigand A, Colot O (2010) Fuzzy filter based on interval-valued fuzzy sets for image filtering. Fuzzy Sets Syst 161:96–117

Biswas P, Sarkar A, Mynuddin M (2015) Deblurring images using a Wiener filter. Int J Comput Appl 109(7):36–38

Carnet M, Callet PL, Barba D (2008a) Objective quality assessment of color images based on a generic perceptual reduced reference. Sig Process Image Commun 23(4):239–256

Carnet M, Callet PL, Barba D (2008b) Objective quality assessment of color images based on a generic perceptual reduced reference. Image Commun 23(4):239–256

Chang SG, Yu B, Vetterli M (2000) Spatially adaptive wavelet thresholding with context modeling for image denoising. IEEE Trans Image Process 9(9):1522–1531

Constantin J, Bigand A, Constantin I, Hamad D (2015) Image noise detection in global illumination methods based on FRVM. NeuroComputing 64:82–95

Cui X, Shi Z, Lin J, Huang L (2012) The research of image quality assessment methods testing. Physics Procedia 25(25):485–491

Dawood F, Rahmat R, Kadiman S, Abdullah L, Zamrin M (2012) Effect comparison of speckle noise reduction filters on 2d-echocardiographic. World Acad Sci Eng Technol 6(9):425–430

Delepoulle S, Bigand A, Renaud C (2012) A no-reference computer-generated images quality metrics and its application to denoising. In: IEEE intelligent systems IS'12 conference, vol 1, pp 67–73

Donahue J, Jia Y, Vinyals O, Hoffman J, Zhang N, Tzeng E, Darrell T (2014) A deep convolutional activation feature for generic visual recognition. In: Proceedings of international conference on machine learning ICML'14, pp 647–655

Farrugia J, Péroche B (2004) A progressive rendering algorithm using an adaptive perceptually based image metric. Comput Graph Forum 23(3):605–614

Fernandez-Maloigne C, Robert-Inacio F, Macaire L (2012) Digital Color: Acquisition, Perception. Wiley-ISTE

Gao D, Liao Z, Lv Z, Lu Y (2015) Multi-scale statistical signal processing of cutting force in cutting tool condition monitoring. Int J Adv Manuf Technol 80(9):1843–1853

Gou H, Swaminathan A, Wu M (2009) Intrinsic sensor noise features for forensic analysis on scanners and scanned images. IEEE Trans Inf Forensics Secur 4(3):476–491

Hedman P, Karras T, Lehtinen J (2016) Sequential Monte Carlo instant radiosity. In: Proceedings of the 20th ACM SIGGRAPH symposium on interactive 3D graphics and games, pp 121–128
Heinonen P, Neuvo Y (1987) FIR-median hybrid filters. IEEE Trans ASSP 35(6):832–833
Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the 22nd conference on multimedia ACM'14, pp 675–678
Kajiya J (1986) The rendering equation. ACM Comput Graph 20(4):143–150
Krizhevsky A, Sutskever I, Hinton G (2012) Imagenet classification with deep convolutional neural networks. In: Proceedings of NIPS, pp 1097–1105
Kumar P (2014) Satellite image denoising using local spayed and optimized center pixel weights. Int J Electr Comput Eng 4(5):751–757
Lagae A et al (2010) State of the Art in procedural noise functions, vol 28. EG 2010, Eurographics Association, London
Lahoudou A, Viennet E, Haddadi M (2010) Variable selection for image quality assessment using a neural network based approach. In: 2nd european workshop on visual information processing (EUVIP), pp 45–49
Lahoudou A, Viennet E, Bouridane A, Haddadi M (2011) A complete statistical evaluation of state of the art image quality measures. In: The 7th international workshop on systems, signal processing and their applications, pp 219–222
Lee SH, Chan CS, Mayo SJ, Remagnino P (2017) How deep learning extracts and learns leaf feature for plant classification. Pattern Recogn 71:1–13
Liu X, Pedersen M, Hardeberg J (2014) Cid:iq a new image quality database. Image Signal Process 8509:193–202
Longhurst P, Debattista K, Chalmers A (2006) A GPU based saliency map for high-fidelity selective rendering. In: AFRIGRAPH 2006 4th international conference on computer graphics, virtual reality, visualisation and interaction in Africa, pp 21–29
Makandar A, Halalli B (2015) Image enhancement techniques using highpass and lowpass filters. Int J Comput Appl 109(14):21–27
Mitchell D (1987) Generating antialiased images at low sampling densities. In: Proceedings of SIGGRAPH'87, New York, NY, USA, pp 65–72
Ren J (2012) Ann vs. svm: which one performs better in classification of mmcs in mammogram imaging. Knowl. Based Syst 26:144–153
Shirley P, Wang C, Zimmerman K (1996) Monte Carlo techniques for direct lighting calculations. ACM Trans Graph 15(1):1–36
Takouachet N, Delepoulle S, Renaud C (2007) A perceptual stopping condition for global illumination computations. In: Proceedings spring conference on computer graphics 2007, Budmerice, Slovakia, pp 61–68
Tipping ME (2003) Fast marginal likelihood maximization for sparse Bayesian models. In: Bishop CM, Frey BJ (eds) Proceedings of the ninth international workshop on artificial intelligence and statistics
Tipping ME (2004) Bayesian inference: an introduction to principles and practice in machine learning. Adv Lect Mach Learn 3176:41–62 Lecture notes in computer sciences
Virtanen T, Nuutinen M, Vaahteranoksa M, Oittinen P, Hkkinen J (2015) Cid 2013: a database for evaluating no-reference image quality assessment algorithms. IEEE Trans Image Process 24(1):390–402
Xu L, Lin W, Kuo CCJ (2015) Visual quality assessment by machine learning, vol 28. Springer Brief, London

Chapter 5
Reduced-Reference Methods

Reduced-reference image quality assessment needs no prior knowledge of reference image but only a minimal knowledge about processed images. A new reduced-reference image quality measure, based on SVMs and RVMs, using a supervised learning framework and synthetic images is proposed in this chapter. This new metric is compared with experimental psycho-visual data with success and shows that inductive learning is a good solution to deal with small sizes of the databases of computer-generated images. As reduced-reference techniques need only small size of labeled samples, thus the rapidity of the learning process is increased.

5.1 Introduction

We now introduce some new learning algorithms that can be applied with success to the considered application. In many practical global illumination applications, the reference image is not available at the beginning of the algorithm, and a blind quality assessment approach is desirable. However, the no-reference models have a crucial drawback because they are based on theoretical models of noise (Zhang et al. 2011; Delepoulle et al. 2012). In the third type of method, the reference image is only partially available, in the form of a set of extracted features made available as side information to help evaluate the quality of the distorted image. This is referred to as reduced-reference quality assessment (Lahoudou et al. 2010a; Renaud et al. 2011). However, the proposed models provide incomplete results due to the complexity of the internal human vision system. They are very efficient for learning perceptual features, but, and at the same time, they are less efficient for learning noise. Moreover, they require a huge number of labeled images in order to train and evaluate the perception models. In (Constantin et al. 2015a), the authors realized a Fast Relevance Vector Machine (FRVM) architecture for the noise perception model that

A. Bigand et al., *Image Quality Assessment of Computer-generated Images*,
SpringerBriefs in Computer Science, https://doi.org/10.1007/978-3-319-73543-6_5

uses fewer basis functions than a comparable SVM does and as well offering good prediction on the testing images. However, they used only one single scene to train and evaluate the FRVM model without considering an inductive learning algorithm for choosing the most informative images. It is quite clear that using only one learning scene is not enough to train the model in order to obtain a good generalization on different computer-generated images. In case of Spike Neural Network (SNN), the authors realized a simple architecture for the noise model, whereas the learning algorithm is still inefficient to learn a huge number of different images in order to make good perception (Constantin et al. 2015b). Furthermore, the neural dynamic should be evaluated for a certain period of time which makes the computation time for questioning the model very slow. These drawbacks lead to study a more possible biological adapted method based on inductive learning. The inductive learning, in its most general sense, refers to machine learning where the learning algorithm has some degree of control over the images on which it is trained (Guo and Wang 2015; Paiseley et al. 2010). Dornaika et al. developed a semi-supervised feature extraction algorithm for pattern categorization (Dornaika et al. 2016). However, this algorithm, which seeks a linear subspace close to a nonlinear one for feature extraction, was tested only for the classification of a benchmark databases data without considering real-life applications. Ruz improves the performance of the inductive rules family classifiers by presenting a new technique for the presentation order of the training samples which combines a clustering method with a density measure function (Ruz 2016). The main drawback with this approach is that the convergence to a good result could be quite time consuming especially when the training set contains thousands of samples. So, it cannot be applied in case of global illumination, because the inductive learning algorithm should select pertinent samples from a set containing a huge number of unlabeled sub-images. In order to reduce manual labeling workload, we try here to consider active and semi-supervised learning algorithms for image noise detection in global illumination. The active learning algorithm aims at finding the most informative image in each iteration for human labeling, while semi-supervised learning aims at labeling images by the machine itself. It has been shown that active and semi-supervised learning may sometimes overcome the limitations of supervised learning in the fields of classifications (Guo and Wang 2015), characters recognition (Richarz et al. 2014), and computational chemistry (Liu 2014). However, their performance was not investigated in the domain of noise perception for global illumination algorithms.

The main contribution of this work lies in the approach of designing a new inductive learning model based on feature selection in order to detect and to quantify stochastic noise present in global illumination. This algorithm uses SVM semi-supervised learning to select class central images (Guangwu and Minggang 2014) and at the same time selecting during active learning the images close to the classification hyperplane (Guo and Wang 2015). The performance of this algorithm is tested by comparing it with the SVM model on global illumination scenes computed with diffuse and specular rendering (Constantin et al. 2015b). We also make a comparative study between this algorithm and our previous work based on FRVM for image noise detection in global illumination (Constantin et al. 2015a). We demonstrate

that the inductive learning algorithm is powerful to generalize and can effectively perform a good perception for predicting the quality of computer-generated images. The chapter is structured as follows: Sect. 5.2 describes the experimental database we use and Sect. 5.3 explains the inductive inference strategies for global illuminations whereas Sect. 5.4 shows the experimental results. Finally, the work is summarized with some conclusions in Sect. 5.5.

5.2 Fast Relevance Vector Machine

The RVM has a probabilistic Bayesian learning framework and has good generalization capability (see Chap. 4). We now present how to build a fast version of RVM (FRVM). RVM acquires relevance vectors and weights by maximizing a marginal likelihood function. The structure of the RVM is described by the sum of product of weights and kernel functions. A kernel function means a set of basis function projecting the input data into a high- dimensional feature space.

As previously presented, given a dataset of input–target pairs $\{x_n, t_n\}_{n=1}^{N}$, we write the targets as a vector $t = (t_1, ..., t_N)^T$ and express it as the sum of an approximation vector $y = (y(x_1), ..., y(x_N))^T$ and an error vector $\varepsilon = (\varepsilon_1, ..., \varepsilon_N)^T$ (Tipping 2003):

$$t = y + \varepsilon = \phi w + \varepsilon \tag{5.1}$$

where $w = [w_1, w_2, ..., w_M]$ is the parameters vector, $\phi = [\phi_1, ..., \phi_M]$ is the $N \times M$ matrix whose columns comprise the complete set of M basis vectors.

The sparse Bayesian framework makes the conventional assumption that the errors are modeled probabilistically as independent zero-mean Gaussian with variance σ^2. Due to the assumption of independence of the targets t_n, the likelihood of the complete dataset can be written as

$$p(t|w, \sigma^2) = (2\pi)^{-N/2}\sigma^{-N} exp\left(-\frac{\|t - y\|^2}{2\sigma^2}\right) \tag{5.2}$$

Maximizing likelihood estimation of w and σ^2 from (5.2) leads to overfitting. To avoid this, the likelihood function is complemented by a prior over the parameters, which take the following form:

$$p(w|\alpha) = (2\pi)^{-M/2} \prod_{m=1}^{M} \alpha_m^{1/2} exp\left(-\frac{\alpha_m w_m^2}{2}\right), \tag{5.3}$$

where $\alpha = (\alpha_1, ..., \alpha_M)^T$ is a vector of M independent hyperparameters, each one individually controlling the strength of the prior over its associated weight.

Having defined the prior, Bayesian inference proceeds by computing the posterior parameter distribution from Bayes rule as follows:

$$p(w|t,\alpha,\sigma^2) = (2\pi)^{-(N+1)/2}|\Sigma|^{-1/2}exp(-\frac{1}{2}(w-\mu)^T\Sigma^{-1}(w-\mu)), \quad (5.4)$$

where the posterior covariance and mean are, respectively, the following:

$$\mu = \sigma^{-2}\Sigma\phi^T t \quad (5.5)$$

$$\Sigma = (\sigma^{-2}\phi^T\phi + A)^{-1} \quad (5.6)$$

and $A = diag(\alpha_1, ..., \alpha_M)$.

The distinguishing element of Bayesian methods is marginalization, where we attempt to integrate out all nuisance variables. The sparse Bayesian learning is formulated as the maximization with respect to α of the marginal likelihood or equivalently its logarithm:

$$\ell(\alpha) = log(p(t|\alpha,\sigma^2)) = -\frac{1}{2}[Nlog2\pi + log|C| + t^T C^{-1}t] \quad (5.7)$$

with

$$C = \sigma^2 I + \phi A^{-1}\phi^T. \quad (5.8)$$

Values of α and σ^2 that maximize the marginal likelihood cannot be obtained in closed form, and an iterative reestimation method is required. The following approach of (Kim et al. 2006) gives

$$\alpha_i = \frac{\gamma_i}{\mu_i^2} \quad (5.9)$$

$$\sigma^2 = \frac{\|t-\Sigma\mu\|^2}{N-\Sigma_i\gamma_i}, \quad (5.10)$$

where μ_i is the ith posterior mean weight (5.5) and the quantities $\gamma_i = 1-\alpha_i\Sigma_{ii}$ with the ith diagonal element Σ_{ii} of the posterior weight covariance (5.6).

Since many of the hyperparameters tend to infinity during the iterative reestimation, the posterior parameter distribution (5.4) of the corresponding weight becomes highly peak at zero (Tipping 2004). In this optimization process, the vector from the training set that associates with the remaining nonzero weights is called the relevance vector (RV).

The RVM algorithm begins with all the M basis functions included in the model and updated the hyperparameters iteratively. As a consequence of these updates, some basis functions would be pruned and the algorithm would be accelerated but nevertheless the first few iterations would still require $O(M^3)$ computations. The FRVM

algorithm begins with an empty model and adds basis functions to increase the marginal likelihood. For this purpose, the term C can be decomposed as (Tipping 2003)

$$C = C_{-i} + \alpha_i^{-1}\phi_i\phi_i^T, \tag{5.11}$$

where C_{-i} is C with the contribution of ith basis vector removed.

We can write the logarithm $\ell(\alpha)$ as

$$\ell(\alpha) = \ell(\alpha_{-i}) + \frac{1}{2}[log\alpha_i - log(\alpha_i + s_i) + \frac{q_i^2}{\alpha_i + s_i}] = \ell(\alpha_{-i}) + \eta(\alpha_i), \tag{5.12}$$

where $\ell(\alpha_{-i})$ is the marginal likelihood with ϕ_i excluded.

The sparsity and the quality factors are defined as

$$s_i = \phi_i^T C_{-i}^{-1}\phi_i \tag{5.13}$$

$$q_i = \phi_i^T C_{-i}^{-1}t. \tag{5.14}$$

Analysis of $\eta(\alpha_i)$ (Faul and Tipping 2002) shows that $\ell(\alpha)$ has a unique maximum with respect to α_i:

$$\alpha_i = \begin{bmatrix} \frac{s_i^2}{q_i^2 - s_i} & \text{if } q_i^2 > s_i \\ \infty & \text{if } q_i^2 \leq s_i \end{bmatrix}. \tag{5.15}$$

It is relatively straightforward to compute q_i and s_i for all the basis functions including those not currently utilized by the model. However, it is easier to update these values based on Woodbury identity as follows:

$$S_m = \phi_m^T B\phi_m - \phi_m^T B\phi\Sigma\phi^T B\phi_m \tag{5.16}$$

$$Q_m = \phi_m^T B\hat{t} - \phi_m^T B\phi\Sigma\phi^T B\hat{t}, \tag{5.17}$$

where $B = \sigma^{-2}I$, $\hat{t} = \phi_{MP} + B^{-1}(t - y)$ and $\phi_{MP} = \Sigma\phi^T B\hat{t}$ is the point estimate for the parameters which is obtained by evaluating (5.5).

It follows that in the FRVM algorithm, the sparsity and the quality factors are updated as

$$s_m = \frac{\alpha_m S_m}{\alpha_m - S_m} \tag{5.18}$$

$$q_m = \frac{\alpha_m Q_m}{\alpha_m - S_m} \tag{5.19}$$

It is shown that the FRVM algorithm reduces the probability of overfitting and uses a small number of kernel functions (Tipping 2003). However, two main drawbacks of these approaches can be highlighted. On one hand, the data that should be used during

the learning stage has to be carefully chosen, in order for the model to learn what is expected. On the other hand, these kinds of approaches provide us with a black-box model. They give good answers but it is often difficult to know how they learned and exactly what they learned. Next, we briefly remain how to generate the noise feature input vector in order to build the FRVM model for image quality evaluation.

5.3 Image Quality Evaluation (IQE)

5.3.1 IQE Using FRVM

As detailed in the previous chapter for noise feature generation, we first apply image denoising algorithms to an image in order to obtain estimates of the image noise. Given an image L, denoising operation is applied to obtain its denoised version L_D (the dimension of L_D is 13). The estimated image noise at the pixel location (i, j) is obtained by a pixel-wise subtraction between the current image pixel and the denoised one:

$$e(i, j) = |L(i, j) - L_D(i, j)|. \tag{5.20}$$

After some experiments, we find that the mean and the standard deviation give the most significant results from the set of noise quality indexes defined in (Lahoudou et al. 2010b, 2011):

$$f^{(1)}(L) = \frac{1}{V \times W} \sum_{i=1}^{V} \sum_{j=1}^{W} e(i, j) \tag{5.21}$$

$$f^{(2)}(L) = (\frac{1}{V \times W} \sum_{i=1}^{V} \sum_{j=1}^{W} (e(i, j) - f^{(1)}, (L))^2)^{1/2}, \tag{5.22}$$

where $V \times W$ is the dimension of the image matrix L. For each of these 13 components, we extract the noise quality indexes and, therefore, we arrive at a total of 26 features used as input to the SVM learning model.

The experimental dataset is then used for training the FRVM model. In the training protocol, we provided the difference of two sub-images to the model: a sub-image called reference and one of the test sub-images. Ideally, the reference sub-image should be the converged one. But during the use of the model in an iterative global illumination algorithm, the converged image is obviously not available. Thus, the reference image used for learning and noise detection is a quickly ray-traced image of scenes which highlight the same features of the converged one (shadows, textures, reflections, etc.). The implementation of images quality evaluation based on FRVM is given by the marginal likelihood maximization Algorithm 4 (Tipping 2003).

In this approach, the size of globally illuminated scenes is very large and thus training of the learning models is computationally costly. To cope with the problem,

Algorithm 4 *FRVM Image quality algorithm*

1: ***Requirement:*** *a set S of* 512×512 *gray-level scenes, each scene is divided in* 16 *sub-images.*
2: *Apply the different denoising algorithms to the sub-images and save the noise feature vectors with the corresponding desired targets on the learning set.*
3: *Initialize* σ^2 *to some sensible value and the model with a single basis vector* ϕ_i*, setting from (5.15):*

$$\alpha_i = \|\phi_i\|^2/(\|\phi_i^T t||^2/\|\phi_i\|^2 - \sigma^2)$$

All other α_m *are naturally set to infinity.*
4: *Compute* Σ *and* μ *which are scalars initially along with initial values of* s_m *and* q_m *for all M bases* ϕ_m*.*
5: *Select a candidate basis vector* ϕ_i *from the set of all M.*
6: *Compute* $\theta_i = q_i^2 - s_i$*.*
if $\theta_i > 0$ *and* $\alpha_i < \infty$ *then re-estimate* α_i*.*
if $\theta_i > 0$ *and* $\alpha_i = \infty$ *then add* ϕ_i *to the model with updated* α_i*.*
if $\theta_i \leq 0$ *and* $\alpha_i < \infty$ *then delete* ϕ_i *from the model and set* $\alpha_i = \infty$*.*
7: *Update* $\sigma^2 = \|t - y\|^2/(N - M + \sum_{K=1}^{M} \alpha_k \Sigma_{kk})$ *in order to maximize the marginal likelihood defined in (5.7).*
8: *Re-compute* Σ*,* μ *using Eq. (5.5), (5.6) and all* s_m*,* q_m *using Eqs. (5.16)–(5.19) and if converged terminate, otherwise repeat from 5.*

we can conduct feature generation before training, because the complexities of most learning algorithms are proportional to the number of features.

5.3.2 IQE Using Inductive Learning

Learning methods concern algorithms which are able to automatically improve their results over time. Our goal is to study and develop an artificial intelligent model, which should produce the same answer as a set of observers to the noise problem. That means that we want this model to be able to classify images as noisy or not. First, we studied the use of Support Vector Machines (SVMs) to fit the classification problem. The main drawback of these approaches is that the data used during the learning stage have to be carefully chosen for the model to learn what is expected. Also, they require a huge number of nonredundant labeled images in order to train and to evaluate the perception models. For this reason, we use SVM inductive learning (semi-supervised and active learning) to find the most informative noisy image for the learning algorithm and to reduce manual labeling workload. Considering SVMs (Sect. 4.3), for any test image x_i, its class label is determined through the following:

$$y(x) = sgn(\sum_{j=1}^{M} \alpha_i K(x, x_i) + b), \tag{5.23}$$

where M is the number of support vectors.

When employing active learning to work with SVMs, the idea is to take the image that has the lowest confidence as the most informative one. It goes without saying that selecting unlabeled images far from the hyperplane is not helpful since their class membership is known. The most useful images for refining the hyperplane are the unlabeled images near the hyperplane. In order to avoid extra manual labeling images, we utilize a semi-supervised learning algorithm for the remaining large amount of the unlabeled images that are not chosen during the active learning algorithm (Guangwu and Minggang 2014). The proposed algorithm itself selects many class central images to better describe the class distribution and to help SVM-active learning to be more precise in finding the boundary images. In order not to introduce too many labeling errors, the label changing rate is used to ensure the reliability of the predicted labels. In case of global illuminations, whenever the SVM-active learning iterates many times, the unlabeled images should be classified into two classes: the noisy images class (denoted as N) and the reference images class (denoted as R) as follows:

$$R = \{x_i | x_i \in U, f(x_i) > 0\} \tag{5.24}$$

$$N = \{x_i | x_i \in U, f(x_i) < 0\}, \tag{5.25}$$

where U is the set of unlabeled sub-images. For each class, the proposed semi-supervised learning algorithm computes the label changing rates for the unlabeled sub-images and selects the sub-images class central of which the label changing rate is equal to 0 as follows:

$$U_R = \{x_i | x_i \in R, \gamma(x_i) = 0\} \tag{5.26}$$

$$U_N = \{x_i | x_i \in N, \gamma(x_i) = 0\}, \tag{5.27}$$

where $\gamma(x_i)$ refers to the label changing rates of the sub-images x_i. Then, by using U_R and U_N, the two sub-images which are chosen by the learning algorithm from the noisy and the reference classes have the median distance to the current classification hyperplane as follows:

$$x_R = median_{x_i}(d(x_i) | x_i \in U_R) \tag{5.28}$$

$$x_N = median_{x_i}(d(x_i) | x_i \in U_N), \tag{5.29}$$

where $d(x_i)$ is the distance of the image x_i to the current classification hyperplane. Furthermore, by using label changing rate to ensure labeling reliability, datasets U_R and U_N would have a higher confidence on unlabeled sub-images predicted labels. If the label changing rate of a sub-image is equal to zero, it means that this sub-image's predicted label does not change with the adjustment of the hyperplane. Which also means that the sub-image has already been correctly classified in the previous iterations and its predicted label would be very credible. The inductive inference learning for global illuminations is given by Algorithm 5.

Algorithm 5 *Inductive Inference Learning for Global Illumination Method*

Require: *a set S of grayscale scenes, each scene is divided into* 16 *sub-images; M: the iteration times of active learning; N: the iteration times of semi-supervised learning.*

1: Apply the different denoising algorithms to the sub-images and save the noise feature vectors with the corresponding desired targets on the learning set.
2: Split the noise feature vectors into two sets: A small set LA of labeled noise feature vectors with the corresponding desired targets and a large set UL of unlabeled noise feature vectors.
3: $S \leftarrow 0$.
4: $f \leftarrow SVM_{Train}(LA)$.
5: ***while*** $S < N$ ***do***
6: Apply SVM active learning M times by selecting the unlabeled sub-images near to the hyperplane.
7: Apply Semi-Supervised learning for the remaining sub-images and select class centers using equations (5.28) and (5.29).
8: Add by the learning algorithm the predicted targets for class center sub-images.
9: Put the class center sub-images into the set LA and remove them from the set UL.
10: $f \leftarrow SVM_{Retrain}(LA)$.
11: $S \leftarrow S + 1$.
12: ***end while***
13: Apply SVM active learning until convergence.

5.4 Experimental Results and Discussion

5.4.1 Design of the Inductive Model Noise Features Vector

As previously presented (5.2.1), we converted each image into the frequency domain using a noise mask. The noise mask is obtained by computing the difference between quick ray-traced sub-image of scenes and the blurred one. To cope with the problem of training the learning models on very large size of globally illuminated scenes, we estimated the image noise as a pixel-wise subtraction between the current image pixel and the denoised one. The mean value and the standard deviation are applied to the different denoised versions of the image in order to obtain a total of 26 noise features used as inputs to the learning model. In order to validate our results, we plot on the sub-images of the scenes Bar and Chess, the values of mean and the standard deviation versus each sub-image per block for the averaging filter of size 3×3 as shown in Fig. 5.1. We found that the blocks of the scene Bar with 512×512 that the block 5 and blocks 9–16 are in the set of maximum of high-frequency noise; the others blocks are in the set of minimum of high-frequency noise. For the scene Chess with 800×800 resolution, blocks 1–3 and blocks 5–9 are in the set of maximum of high-frequency noise and the others blocks are in the set of minimum of high-frequency noise. The same behaviors are observed by applying the other filters on the other scenes. We can notice the values of these features monotonically change until they reach stable thresholds when the sub-images become noiseless, so they are very important to consider as input to the learning models.

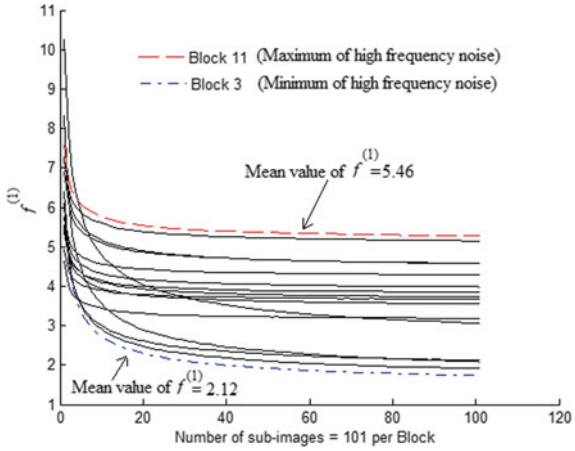


Fig. 5.1 Mean and standard variation per block for the scenes Bar and Chess with 512×512 and 800×800 resolutions

5.4.2 *Inductive SVM Model Selection*

To visually verify the robustness of our model, we first do experiments on the scenes with 512×512 resolutions Bar, Class, Cube, DeskRoom1, Sponza, and DeskRoom2. The input of the SVM model is the luminance vector of size 128×128. The support vector machine model with noise feature (FSVM) uses as input a set of noise quality indexes vector of size 26. The model output is -1 if the image is considered noisy and $+1$ if the image is considered less affected by noise. We choose for SVM radial basis functional kernels (Constantin et al. 2015a). We do a V-times cross-validation with the method of raising the exponent on the scenes Bar, Class, and Cube in order to obtain good precision (Fernandez-Maloigne et al. 2012; Hsu et al. 2003). The sub-images set is split into 303 groups each of size 16 sub-images. Figure 5.2 shows the mean number of support vectors and the precision obtained for different values of the parameter C. It is quite clear that the optimal value of C is equal to 4 when SVM and FSVM are applied. Figure 5.3 shows the variation of the learning precision and the mean number of support vectors with respect to the standard deviation σ for SVM and FSVM. Table 5.1 shows the models of optimal parameters.

Later, we run the inductive learning algorithm by selecting the first 5 blocks of the scene Bar as the initial labeled sub-images. We find that the initial labeled sub-images set has an influence on the choice of sub-images class central and on the

Table 5.1 Optimal parameters for the learning models

Learning model	Optimal standard deviation	Precision %	Mean number of kernels
SVM	50	**96.24**	1750
FSVM	70	90.7	1970

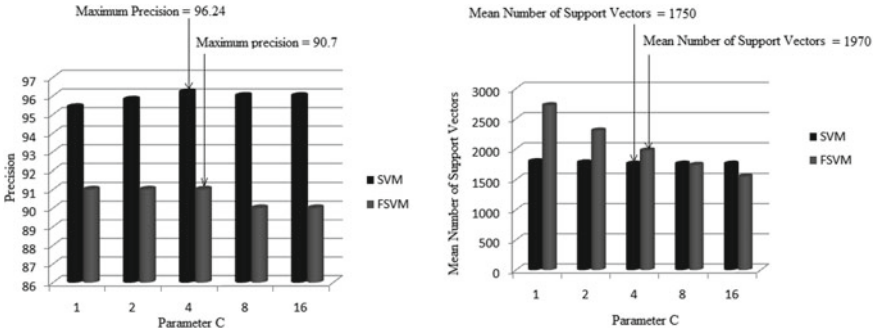


Fig. 5.2 Precision and average number of SVs for different values of the parameter C. Scenes with 512×512 resolution

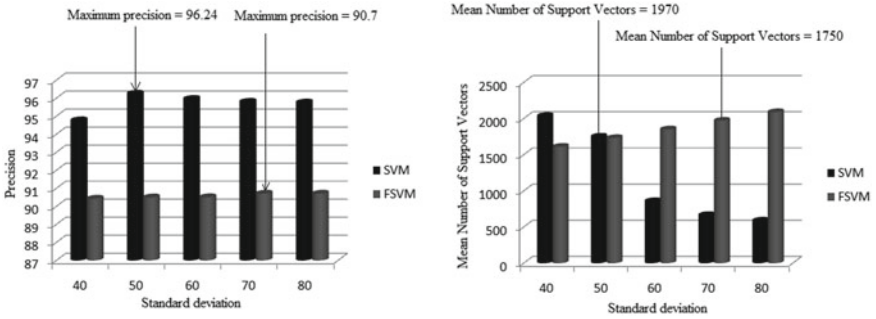


Fig. 5.3 Scenes with 512×512 resolution. Precision and average number of SVs for different values of standard deviation

performance behavior of the inductive learning algorithm as well. That is why, we have chosen the scene Bar which contains different levels of noise in order to get a better precision. The remaining sub-images of the scene Bar and the sub-images of the scenes Class, Cube, DeskRoom1, Sponza, and DeskRoom2 are selected as being the initial unlabeled sub-images. Because the semi-supervised learning algorithm has totally iterated 10 times, 20 sub-images centers are added to the set of labeled sub-images with their predicted labels in case of SVM and FSVM. Here, we find that 20 sub-images centers ($N = 10$) are sufficient for the inductive learning algorithm to achieve a good performance. During active learning, 1091 labeled sub-images are added to the set of labeled images when SVM is used whereas 3212 are added in case FSVM is used in order to obtain a good convergence with a maximum precision equal to 99%. Figure 5.4 shows the variation of precision during the learning process for the SVM and FSVM inductive learning models (SVM-IL and FSVM-IL). In case of SVM-IL, we find that the precision oscillates at the beginning and increases monotonically after querying some pertinent sub-images until it reaches its maximum after 1040 iterations. This might be due to the fact that in this space, the two classes of sub-images overlap severely and are difficult to be classified.

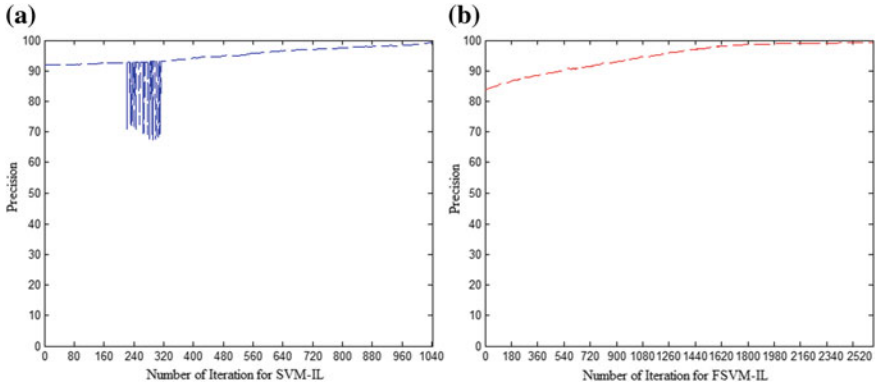


Fig. 5.4 Variation of precision during the iteration of active learning algorithm for SVM-IL (**a**) and FSVM-IL (**b**)

We also find that the number of support vectors is equal to 983 and 2584 for SVM-IL and FSVM-IL respectively. It is clear that the SVM-IL model assures a good convergence by using a number of kernels less than FSVM-IL.

5.4.3 Experiments Using Inductive Learning

To verify the accuracy of our inductive learning models, we first test the inductive learning algorithm on the scenes with 512×512 resolution. We use the sub-images of the scenes Bar and Class to train the SVM and FSVM models. The learning set contains 3232 sub-images computed with different noise levels and de-noised sub-images. Once the training process has been performed, we find that the number of support vectors is equal to 712 and 1560 respectively for SVM and FSVM. It is clear that these models use a number of kernels less than inductive learning models do. This drawback can be solved by applying parallel computation at the hidden layer of the network in order to minimize the required questioning time for the inductive learning models, so they can be used to make online decisions on photo-realistic images scrolling on a video system. Next, the learning models are evaluated on the global illumination scenes Bar, Class, Cube, DeskRoom1, Sponza, and DeskRoom2. Figure 5.5 shows the variation of the mean square error for the optimal architectures which measures the average of the squares of the deviations. That is, the difference between the actual values of the perception models and the desired ones predicted by the HVS (Sect. 5.2). The mean square error range for the learning models is listed in Table 5.2. It is shown that SVM-IL model achieves the best performance on the scenes Class, Cube, DeskRoom1, Sponza, and DeskRoom2. For this model, the mean square error is between 0 and 0.09 for all sub-images. However, the SVM model gives better performance than the SVM-IL only on the scene Bar because it is

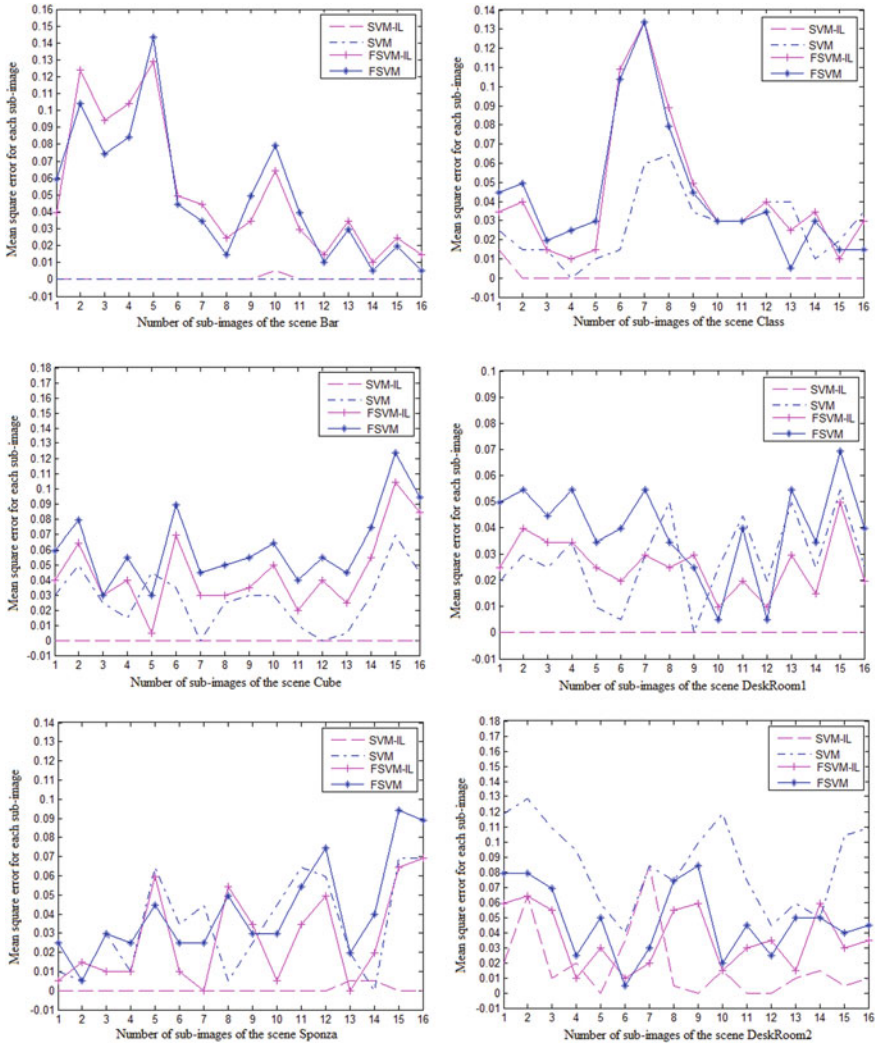


Fig. 5.5 Variation of the mean square error for each sub-image of the scenes with 512 × 512 resolution Bar, Class, Cube, DeskRoom1, Sponza, and DeskRoom2

used as the learning scene for this model. Moreover, this experiment shows that the FSVM-IL model performs better than the FSVM does on the testing scene. It should be highlighted that the SVM-IL and the FSVM-IL models provide mean square errors less than the SVM and the FSVM do respectively. We also make a comparative study between the actual thresholds of the learning models and the desired ones obtained by the human vision system (HVS). Figure 5.6 also shows that the SVM-IL and FSVM-IL models give a good stopping criterion for the global illumination algorithm on

Table 5.2 Mean square error range for each sub-image on the scenes with 512 × 512 Bar, Class, Cube, DeskRoom1, Sponza, and DeskRoom2

Scenes	SVM	SVM-IL	FSVM	FSVM-IL
Bar	0	[0 – 0.01]	[0 – 0.15]	[0 – 0.13]
Class	[0 – 0.07]	[0 – 0.02]	[0 – 0.14]	[0 – 0.14]
Cube	[0 – 0.07]	0	[0 – 0.13]	[0 – 0.11]
DeskRoom1	[0 – 0.06]	0	[0 – 0.07]	[0 – 0.05]
Sponza	[0 – 0.07]	[0 – 0.01]	[0 – 0.1]	[0 – 0.07]
DeskRoom2	[0.04 – 0.13]	[0 – 0.09]	[0 – 0.09]	[0 – 0.07]

all the scenes and achieves a better convergence than the SVM and FSVM can. Moreover, this experiment shows that the SVM-IL's performance is better than the FSVM-IL's.

We also do experiments on the scenes with 800 × 800 resolution computed with diffuse and specular rendering in order to test the learning models capability. The input to SVM is the noise feature vector of size equal to 200 × 200, whereas the FSVM model uses the noise quality indexes vector of size 26 as input. We also do a V-times cross-validation technique by considering the sub-images of the Kitchen and BathRoom1 scenes for the training and the evaluation processes. The set of sub-images is split into 235 groups, each of size 16 sub-images. Figure 5.7 shows the mean number of support vectors and the precision obtained for different values of C. Figure 5.8 shows the variation of the learning precision and the mean number of support vectors with respect to the standard deviation. It is clear that the optimal value of C is equal to 8. The optimal value of standard deviation is equal to 70, the mean number of support vectors is equal to 1770 and the maximum precision is equal to 97.50. In case of FSVM, the set of sub-images is split into 630 groups, each of size 16 sub-images. In this case, the optimal value of C is equal to 8. The optimal value of standard deviation is equal to 120, the mean number of support vectors is equal to 5350, and the maximum precision is equal to 83.22 (Figs. 5.7 and 5.8).

As previously, the initial set of labeled sub-images is initialized to the first 5 blocks of the scene BathRoom1. This scene is used because it contains different levels of noise and, at the same time it gives a good precision of the inductive learning algorithm. The unlabeled set is initialized to the remaining sub-images of the global illumination scenes with 800 × 800 resolution. Later, we apply the inductive learning algorithm with the parameters $N = 10$ and $M = 5$. During the active learning, 2386 labeled sub-images are added to the set of labeled sub-images in case of SVM-IL and 3285 are added in case of FSVM-IL in order to obtain a good convergence with a precision equals to 99% and 93.4% for SVM-IL and FSVM-IL respectively. We find that the number of support vectors is equal to 1402 and 2066 for SVM-IL and FSVM-IL respectively. We train the SVM and the FSVM models on the sub-images of the scenes Kitchen and BathRoom1. The learning set contains 3760 sub-images computed with different noise levels. As a result, we find that the number of support

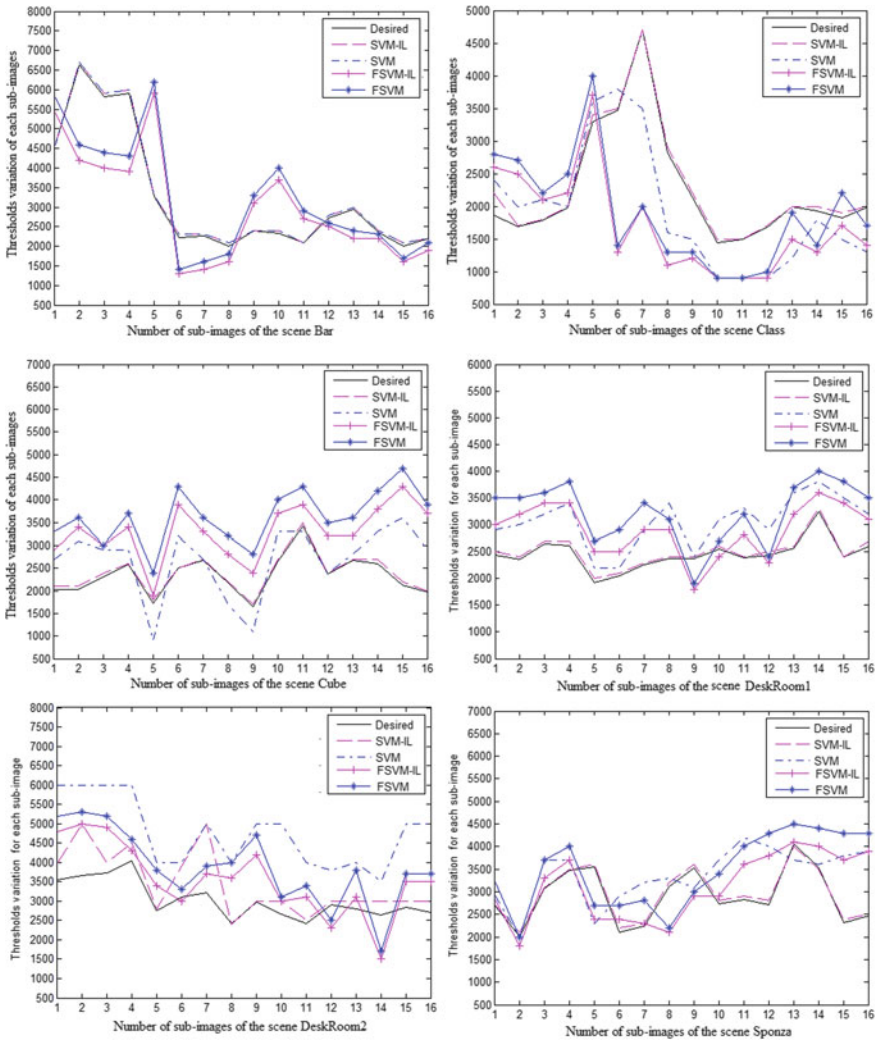


Fig. 5.6 Variation of the actual thresholds of the learning models and the desired ones obtained by the HVS for each sub-image on the scenes with 512 × 512 resolution Bar, Class, Cube, DeskRoom1, Sponza, and DeskRoom2

vectors is equal respectively to 2367 and 1380 for SVM and FSVM. It is shown that SVM-IL model assures a good convergence, using for the learning process, a number of labeled sub-images less than the other learning models. That is why, it uses a number of kernels less than FSVM-IL, SVM, and FSVM models. Next the path tracing algorithm is applied where at each iteration, a different number of paths are added to the un-converged sub-images depending on the selected scene (Sect. 5.2). For each block of the selected scenes, we plot the variation of the mean square

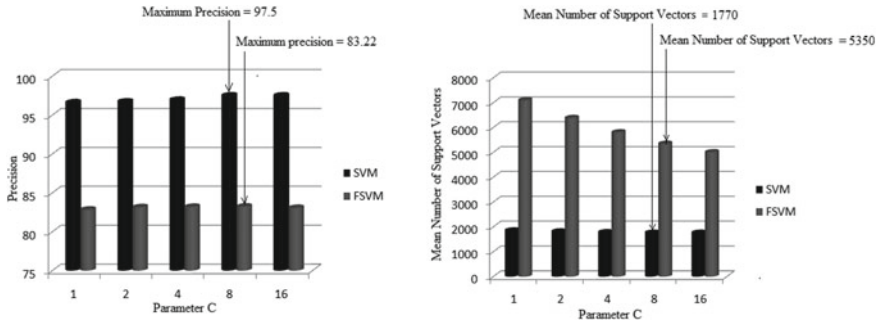


Fig. 5.7 Precision and Average number of SVs for different values of the parameter C in case of scenes with 800 × 800 resolution

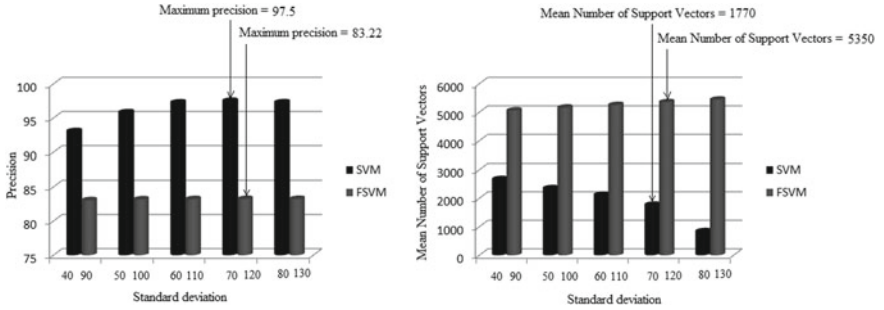


Fig. 5.8 Precision and Average number of SVs for different values of Standard Deviation in case of scenes with 800 × 800 resolution

error (Fig. 5.9), the thresholds of the learning models, as well the desired thresholds obtained by the human vision system (HVS) (Fig. 5.10).

The mean square error range for the learning models are listed in Table 5.3. It is clear that the SVM-IL model achieves the best performance on all the scenes with 800 × 800 resolution. The mean square error is between 0 and 0.03 for the sub-images of the scenes Kitchen, Chess, BathRoom1, Pnd, BathRoom2, and DeskRoom, excepting block 5 of the scene SdbCenter where the SVM-IL model assures same performances as SVM with a mean square error equal to 0.15. It should be also noticed that the FSVM-IL model provides mean square errors less than the FSVM's on all the scenes with 800 × 800 resolution excepting the sub-images of the scene kitchen because it is used as the learning scene for this model. This study also shows that the SVM-IL model assures threshold values similar to those of the Human Vision System (HVS) and gives a good stopping criterion for global illumination algorithms on all the used scenes containing different characteristics and noise levels.

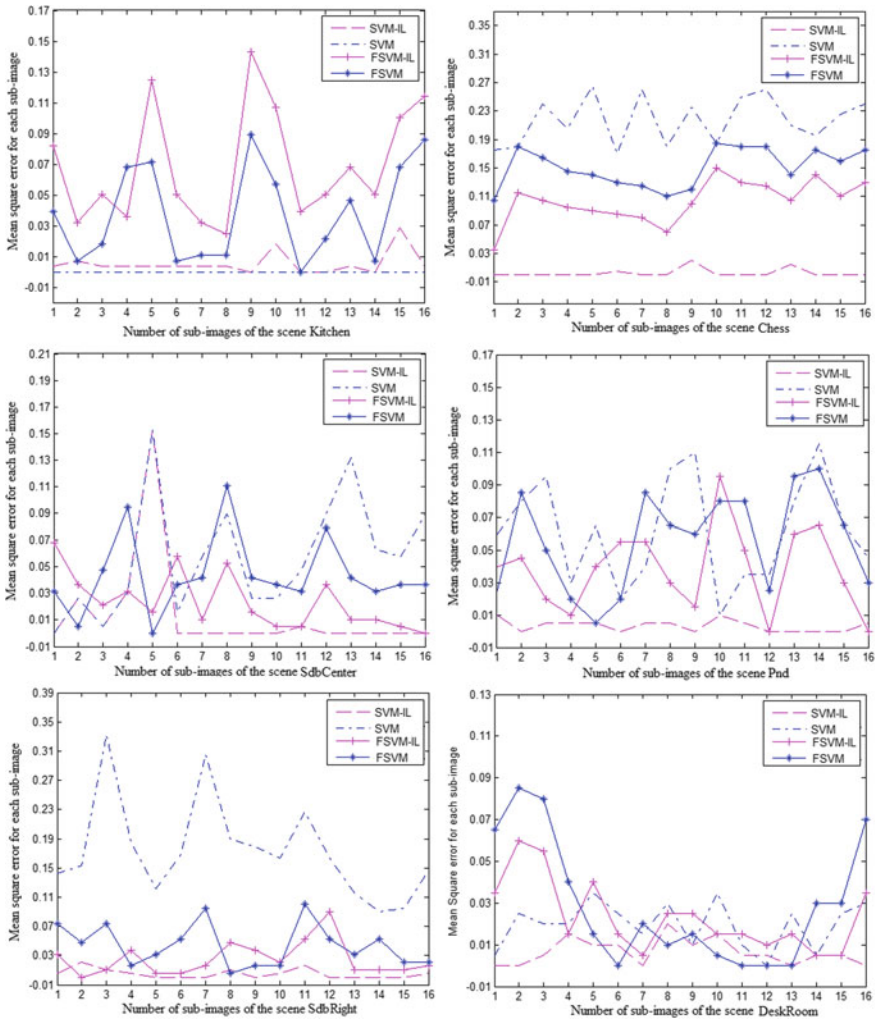


Fig. 5.9 Variation of the mean square error for each sub-image of the scenes with 800 × 800 Kitchen, Chess, BathRoom1, Pnd, BathRoom2, and DeskRoom

5.4.4 Comparison with the Fast Relevance Vector Machine

Previous work proposed fast relevance vector machine (FRVM) models for stochastic noise detection in case of synthetic images with 512 × 512 resolution (Constantin et al. 2015a). The FRVM learning algorithm introduced by Tipping is a probabilistic approach that uses a number of kernel functions less than the SVM. This small number does not necessary satisfy the needed Mercer's condition (Tipping 2003). However, it is typically much slower in convergence than SVM. It was shown that

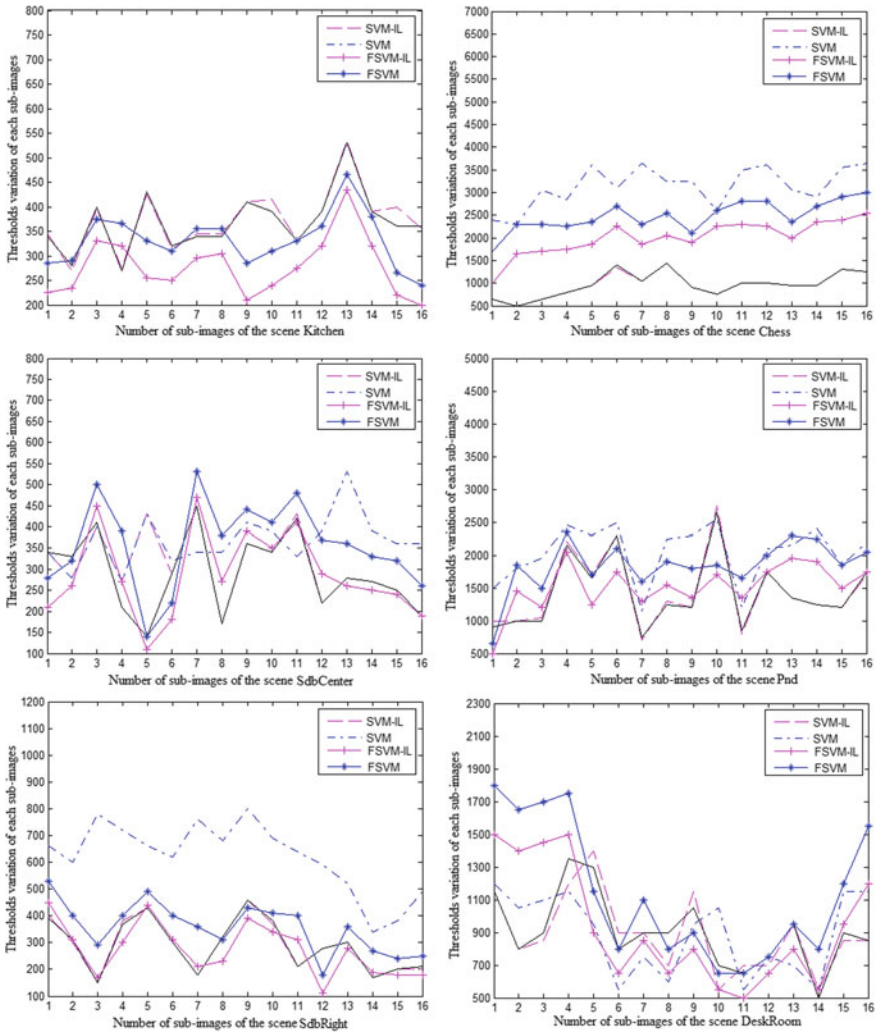


Fig. 5.10 Variation of the actual thresholds of the learning models and the desired ones obtained by the HVS for each sub-image on the scenes with 800 × 800 Kitchen, Chess, BathRoom1, Pnd, BathRoom2, and DeskRoom

the FRVM is much better in performance than the SVM in case of noise perception in global illumination (Constantin et al. 2015a). In order to make a comparative study between the inductive learning and the FRVM models, the scenes Bar and Class are used for the optimization and the learning processes. In fact, considering only one scene as our previous work is definitely not enough for best optimization and evaluation. We use the V-times cross-validation technique in order to find the optimal value of standard deviation σ for FRVM and FFRVM (FRVM with 26 features). The

Table 5.3 Mean square error range for each sub-image on the scenes with 800 × 800 Kitchen, Chess, Pnd, BathRoom1, BathRoom2, and DeskRoom

Scenes	SVM	SVM-IL	FSVM	FSVM-IL
Kitchen	0	[0 − 0.03]	[0 − 0.09]	[0.01 − 0.15]
Chess	[0.15 − 0.27]	[0 − 0.03]	[0.07 − 0.19]	[0.03 − 0.15]
BathRoom1	[0 − 0.15]	[0 − 0.15]	[0 − 0.13]	[0 − 0.07]
Pnd	[0.05 − 0.13]	[0 − 0.01]	[0 − 0.11]	[0 − 0.11]
BathRoom2	[0.07 − 0.35]	[0 − 0.03]	[0 − 0.11]	[0 − 0.11]
DeskRoom	[0 − 0.05]	[0 − 0.03]	[0 − 0.09]	[0 − 0.05]

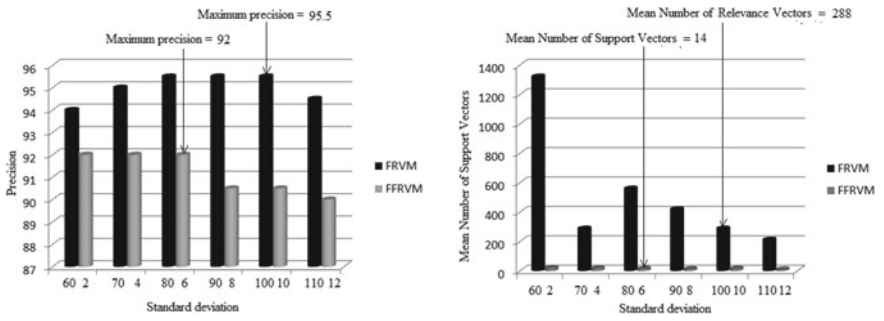


Fig. 5.11 Precision and average number of RVs for different values of Standard Deviation in case of scenes with 512 × 512 resolution

Table 5.4 Optimal parameters for the learning models

FRVM	Optimal standard deviation	Precision %	Mean number of kernels
FRVM	100	**95.5**	288
FFRVM	6	92	14

set of sub-images is split into 202 groups, each of size 16 sub-images (Fig. 5.11). The optimal parameters for FRVM models are shown in Table 5.4.

Then, we plot the actual thresholds of the learning models for each block for the images with 512 × 512 resolution and the desired ones obtained by the HVS. These results show that the inductive model SVM-IL is better in performance than the FRVM and the FFRVM models. Moreover, the FSVM-IL model is better in convergence than FRVM and gives similar results as the FFRVM do. However, the FRVM models use a number of relevance vectors less than the SVM-IL and FSVM-IL models' (Fig. 5.12). So in order to minimize the questioning time of the SVM-IL model during real-time application, a parallel computation using GPU card is a requirement.

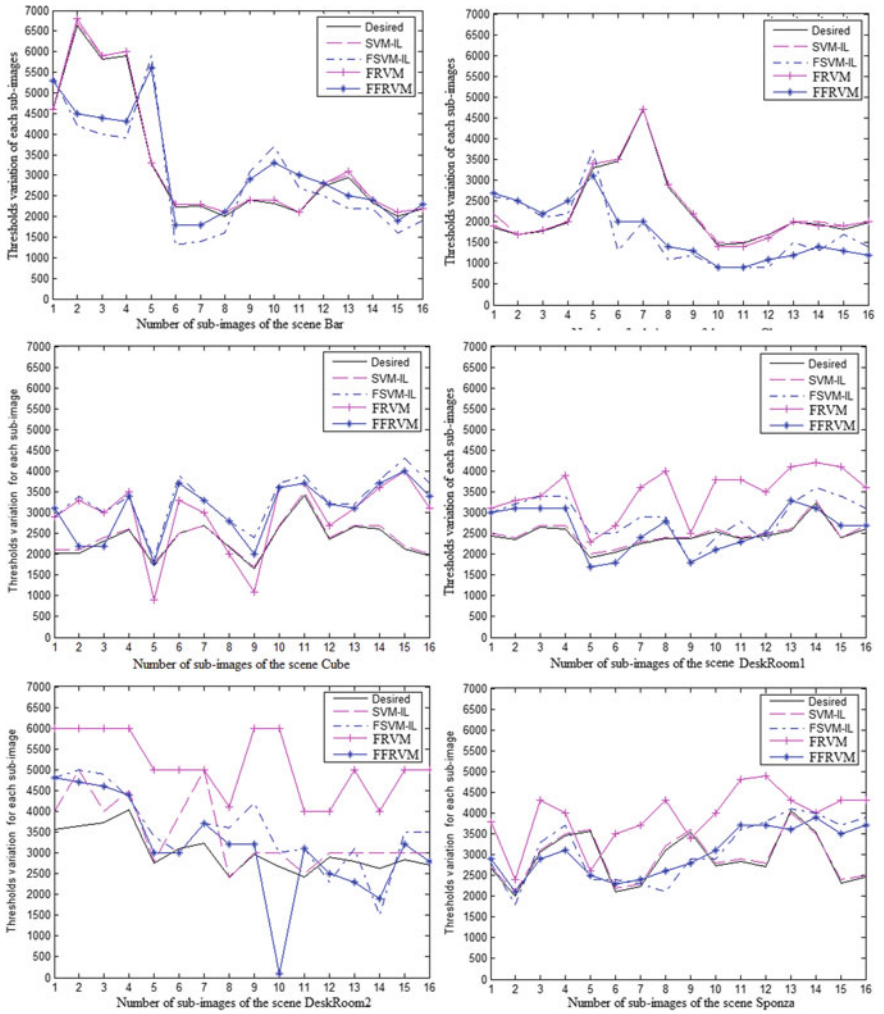


Fig. 5.12 Variation of the actual thresholds of the learning models and the desired ones obtained by the HVS for each sub-image on the scenes with 512×512 resolution Bar, Class, Cube, DeskRoom1, Sponza, and DeskRoom2 in case of inductive learning and FRVM models

5.5 Conclusion

This chapter introduced the application of the inductive learning algorithm for noise perception in the global illumination algorithms. The introduced approach selects the pertinent images for the inductive learning model by using active learning to select class boundary images, and the semi-supervised learning to select class central images. Thus, it can be applied to a large dataset of around $15,400$ images,

and we show that the proposed technique offers high prediction on the testing base when compared with the SVM and FRVM machines. By using an inductive learning machine approach, the learning models improve their performance by minimizing manual labeling and selecting the most pertinent images for the noise perception algorithm. A GPU card using parallel computing can be applied to optimize inductive learning and make online decisions. This new learning technique makes it possible to study how to easily manage more highly indirect lighted environments such as the use of a new noise descriptor. In addition to this, the realization of a new FRVM inductive learning model for global illumination algorithms should be a new research direction. Finally, our approach requires building a relatively small number of images set computed with different rendering algorithms. This work is still difficult to realize because of the time required for scene rendering, but the gain of learning time is important compared with full-reference techniques.

References

Constantin J, Bigand A, Constantin I, Hamad D (2015a) Image noise detection in global illumination methods based on FRVM. NeuroComputing 64:82–95

Constantin J, Constantin I, Rammouz R, Bigand A, Hamad D (2015b) Perception of noise in global illumination algorithms based on spiking neural network. In: The IEEE third international conference on technological advances in electrical, electronics and computer engineering, pp 68–73

Delepoulle S, Bigand A, Renaud C (2012) A no-reference computer-generated images quality metrics and its application to denoising. In: IEEE intelligent systems IS'12 Conference, vol 1, pp 67–73

Dornaika F, Traboulsi YE, Assoum A (2016) Inductive and flexible feature extraction for semi-supervised pattern categorization. Pattern Recognition

Faul AC, Tipping ME (2002) Analysis of sparse Bayesian learning, In: Dietterich TG, Becker S, Ghahramani Z (eds) Advances in neural information processing systems, vol 14, pp 383–389

Fernandez-Maloigne C, Robert-Inacio F, Macaire L (2012) Digital Color: Acquisition. Perception, Wiley-ISTE

Guangwu A, Minggang S (2014) Semi-supervised learning based social image semantic mining algorithm. J Multimed 9(2):245–252

Guo H, Wang W (2015) An active learning-based SVM multi-class classification model. In: Pattern recognition, pp 130–135

Hsu CW, Chang CC, Lin C (2003) A practical guide to support vector classification. National Taiwan University, Technical Report Department of Computer Science

Kim J, Suga Y, Won S (2006) A new approach to fuzzy modeling of nonlinear dynamic systems with noise: rlevance vector learning machine. IEEE Trans Fuzzy Syst. 14(2):222–231

Lahoudou A, Viennet E, Beghdadi A (2010a) Selection of low-level features for image quality assessment by statistical methods. J Comput Inf Technol 2:183–189

Lahoudou A, Viennet E, Haddadi M (2010b) Variable selection for image quality assessment using a neural network based approach. In: 2nd European workshop on visual information processing (EUVIP), pp 45–49

Lahoudou A, Viennet E, Bouridane A, Haddadi M (2011) A complete statistical evaluation of state of the art image quality measures. In: The 7th International workshop on systems, signal processing and their applications, pp 219–222

Liu Y (2014) Active learning with support vector machine applied to gene expression data for cancer classification. J Chem Inf Comput Sci 44(6):1936–1941

Paiseley J, Liao X, Carin L (2010) Active learning and basis selection for kernel-based linear models: a Bayesian perspective. IEEE Trans Signals Process 5(8):2686–2700

Renaud C, Delepoulle S, Takouachet N (2011) Detecting visual convergence for stochastic global illumination. In: Intelligent computer graphics, pp 1–17

Richarz J, Vajda S, Grzeszick R, Fink G (2014) Semi-supervised learning for character recognition in historical archive documents. J Pattern Recogn 47(3):1011–1020

Ruz GA (2016) Improving the performance of inductive learning classifiers through the presentation order of the training patterns. Expert Syst Appl 58:1–9

Tipping ME (2003) Fast Marginal Likelihood Maximization for Sparse Bayesian Models . In: Bishop CM, Frey BJ (eds) Proceedings of the ninth international workshop on artificial intelligence and statistics

Tipping ME (2004) Bayesian inference: an introduction to principles and practice in machine learning. In: Advanced lectures on machine learning, Lecture Notes in Computer Sciences, vol 3176, pp 41–62

Zhang J, Ong S, Thinh M (2011) Kurtosis based no-reference quality assessment of jpeg2000 images. Sig Process Image Commun 26(1):13–23

Chapter 6
No-Reference Methods and Fuzzy Sets

This chapter presents a new noise level estimator, based on interval Type-2 Fuzzy Sets (IT2 FSs) and devoted to computer-generated images. This model can then be used in any progressive stochastic global illumination method in order to estimate the noise level of different parts of any image. A comparative study of this model with a simple test image demonstrates the good consistency between an added noise value and the results from the noise estimator. The proposed noise estimator results have been too compared with full-reference quality measures (or faithfullness measures) like SSIM and give satisfactory performance. This kind of technique is interesting when a low number of learning samples are available, or to obtain a quick convergence of computer-generated images before a more complete treatment.

6.1 Introduction

Preprocessing of digital images with fuzzy filtering techniques often gives good results. In this chapter, we present an interesting application of interval type-2 fuzzy sets for image treatment (quality assessment and noise estimation). Indeed, interval type-2 fuzzy sets are effective to take different kinds of uncertainty into account (imprecision, noise, etc.). The great family of entropies provides some interesting tools to take decision in uncertain environment. The main idea of the presented technique is to use fuzzy entropy to detect noise instead of noise features of image in the case of full and reduced-reference learning methods. Fuzzy entropy is then used to build a new noise estimator, based on interval type-2 fuzzy sets and devoted to natural and computer-generated images. This model can then be used to estimate the noise level of different kinds of noise (impulsive, Gaussian, speckle noise, and perceptual noise in computer-generated images). A comparative study of this model with test images demonstrates the good consistency between an added noise value

A. Bigand et al., *Image Quality Assessment of Computer-generated Images*,
SpringerBriefs in Computer Science, https://doi.org/10.1007/978-3-319-73543-6_6

and the results from the noise estimator. This new framework can be easily extended to other applications like information retrieval, database querying, etc.

Perceptual noise is not well known at the moment. According to us, it is certainly a mixture of different kinds of noises as Gaussian noise, Perlin noise, etc., and image information incompleteness. **Uncertainty** is classically classified using three conceptually distinctive characteristics, fuzziness (blur), randomness, and incompleteness. This classification is interesting for many applications, like sensor management (image processing, speech processing, and time series processing), practical decision-making, and particularly for this work (computer-generated image).

Fuzziness, one of the features of uncertainty, results from the lack of sharp distinction of the boundary of a set and is commonly modeled using Fuzzy Sets (FSs) introduced by Zadeh (1965). The first attempt to quantify the fuzziness was made in 1968 by Zadeh (1968), who introduced fuzzy entropy. Fuzzy entropy is one of the key techniques for fuzziness (vagueness and imprecision) management. Fuzzy sets are interesting tools to evaluate a data element along with the information contained in its neighborhood (Bigand and Colot 2010; Nachtegael et al. 2005; Babu and Sunitha 2011) and they make it possible to manage the imprecision that is present in discrete images. Typical membership functions of fuzzy sets are often generated based on fuzzy entropy.

These techniques consider that measurements have inherent vagueness rather than **randomness**. However, there remain some sources of uncertainty in ordinary (or precise) fuzzy sets (see Mendel and John 2002): in the fuzzy set design (e.g., in symbolic description), in the fact that measurements may be noisy or that the data used to calibrate the parameters of ordinary fuzzy sets may also be noisy. So, since fuzzy sets were introduced by Zadeh (1965), many new approaches treating imprecision and uncertainty (which are naturally present in image processing (Bloch 1996)) were proposed (see Dubois and Prade 2005 for a rapid discussion about some of these theories). Among these, is a well-known generalization of an ordinary fuzzy set, the Interval-Valued Fuzzy Set (IVFS), first introduced by Zadeh (1975). In Bigand and Colot (2010), we have shown that ultrafuzziness index (which is also called IVFS entropy in the sequel) is a particularly adapted tool to detect relevant information in noisy images and makes impulse noise removal from images possible. IVFSs enable to model the lack of knowledge of the expert when constructing the membership function of Fuzzy Sets (FSs) used for image denoising. We have also shown (Bigand and Colot 2012) that this method is efficient to remove speckle noise. We have recently applied it to define a no-reference quality metric of computer-generated images (Delepoulle et al. 2012) (and its application to denoising).

Incompleteness often affects time series prediction (time series obtained from marine data (salinity, temperature, etc.)). Wilbik and Keller recently show that a fuzzy similarity measure exists between sets of linguistic collections (Wilbik and MKeller 2012). They also proved that (Wilbik and Keller 2012) this fuzzy similarity measure is a new metric used to detect abnormalities or state change for huge time series. We have also shown that IVFSs make it a global uncertainty modelization possible for image processing (Bigand and Colot 2010).

In many algorithms devoted to image processing, estimation of noise present in the image is necessary to process the image optimally. This work is mainly devoted to IVFSs to deal with these three dimensions of uncertainty (fuzziness, randomness, and incompleteness). We illustrate the proposed technique through two difficult tasks, Gaussian noise estimation and removal in digital images and the management of global illumination methods.

The main objective of global illumination methods is to produce **synthetic images** with photo-realistic quality. As we saw these methods are generally based on path-tracing theory in which stochastic paths are generated from the camera point of view through each pixel toward the 3D scene.

It is so obvious that automatically measure of image quality is very important to characterize images visual quality. They are of great interest in image compression (JPEG models) and in image synthesis. Image quality metrics are usually categorized into three models in the literature: full-reference (such as the signal-to-noise ratio SNR and structural similarity index measure SSIM (Wang et al. 2004)), no-reference (image quality is estimated without access to reference images (Zhang et al. 2011; Ferzli and Karam 2005; Delepoulle et al. 2012)), and reduced-reference models (Wang and Simoncelli 2005; Li and Wang 2009; Lahoudou et al. 2010). However, the proposed models, which are based on theoretical models of noise, present sensitivity limits in global illuminations.

On the other hand, the fact that synthetic images are generated from noisy to correct images may lead to a noise estimation based method. Noise models have not been heavily studied concerning synthetic images, and they are often difficult to establish concerning CCD cameras images. Olsen (1993) showed that the most reliable estimate is obtained by prefiltering the image to suppress the image structure and then computing the standard deviation value (of white additive noise) from the filtered data. Another possible way to noise estimation is to use interesting results of recent efficient filters, as the one we proposed in (Bigand and Colot 2010). So this chapter focuses on the use of a new noise estimation model to detect and to quantify stochastic noise in a synthetic image. The chapter is structured as follows. Section 6.2 describes the uncertainty detection technique we use that is to say fuzzy entropy. Section 6.3 introduces the design of the IVFS image noise estimation, and Sect. 6.4 shows some experimental results obtained by the estimated noise model. Finally, the work is summarized with some conclusions and perspectives in Sect. 6.5.

6.2 Interval-Valued Fuzzy Sets

The concept of a type-2 fuzzy set was introduced first by Zadeh (1975) as an another possible extension of the concept of an FS. Type-2 fuzzy sets have membership degrees that are themselves fuzzy and have been proved of being capable of enhancing uncertainty handling abilities of traditional type-1 fuzzy sets. So they have been extensively applied to image processing (Sussner et al 2011; Jurio et al. 2011; Bigand and Colot 2010). Let us start with a short review of basics concepts related to IVFS.

Let $S([0, 1])$ denote the set of all closed subintervals of the interval $[0, 1]$, an Interval Type-2 Fuzzy Set (IT2FS or Interval-Valued Fuzzy Set (IVFS): the two terms are equivalent in the following) A in a non-empty and crisp universe of discourse X is a mapping from X to S (Zadeh 1975; Bustince et al. 2009). For each IVFS A, we denote by $\delta_A(x)$ the amplitude of the considered interval ($\delta_A(x) = \mu_{AU}(x) - \mu_{AL}(x)$). So nonspecific evidence (an interval of membership values) for x belonging to a linguistic value A is identified by IVFS.

6.2.1 Uncertainty Representation

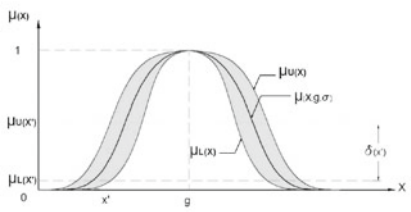
The uncertainty of membership function of a precise FS is modeled using the length of the interval $\delta(x)$ in an IVFS (the longer $\delta(x)$ the more uncertainty), so choice of functions $\mu_U(x)$ and $\mu_L(x)$ is crucial. We proposed to use interval-valued fuzzy sets with the following functions $\mu_U(x)$ and $\mu_L(x)$:

- upper limit: $\mu_U(x) : \mu_U(x) = [\mu(x; g, \sigma)]^{1/\alpha}$, (with $\alpha = 2$),
- lower limit: $\mu_L(x) : \mu_L(x) = [\mu(x; g, \sigma)]^{\alpha}$,

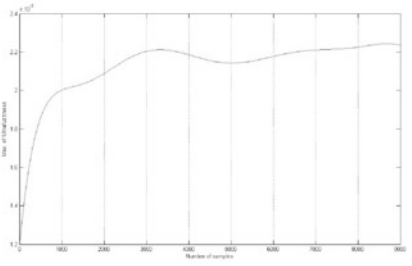
where $\mu(x; g, \sigma)$ is a Gaussian (FS) fuzzy number ($\mu(x; g, \sigma)$ is represented in Fig. 6.1) centered on g and which support is set using a free constant parameter σ ($x \in [0, G - 1]$):

$$\mu(x; g, \sigma) = exp\left[-\frac{1}{2}\left(\frac{x - g}{\sigma}\right)^2\right]. \tag{6.1}$$

Indeed, we know since Babaud et al. (1986) that Gaussian kernel is the only linear filter that gives a consistent scale-space theory.



(a) Membership function of an IVFS



(b) Ultrafuzziness evolution vs. samples number, block 0

Fig. 6.1 IVFS membership function and ultrafuzziness evolution according to the number of paths

6.2.2 IVFS Entropy

The process of selecting the necessary information for image processing must lead here to the correct estimate of the image uncertainty. The present work demonstrates an application of fuzzy set theory to estimate noise power, with the highest accuracy possible (so a good evaluation of uncertainty is essential). The terms *fuzziness degree* (Kaufmann 1975) and *entropy* (Deluca and Termini 1972) provide the measurement of uncertainty in a set and are important issues in fuzzy logic studies. These well-known concepts have been developed in (Bigand and Colot 2010). Nevertheless, the total amount of uncertainty is difficult to calculate in the case of Fuzzy Sets (FSs), and particularly when images (represented using an FS) are corrupted with noise, so we proposed to use the IVFS imprecision degree $Ind(A)$ of an IVFS A in X:

$$Ind(A) = \sum_{i=1}^{q} [\mu_U(x_i) - \mu_L(x_i)] \tag{6.2}$$

It is assumed that an observed image I is defined on a $M \times N$ square lattice, and each pixel $(o,\ p)$ takes a gray-level value $g(o,\ p)$ (with G gray levels $g \in [0,\ G-1]$). For an image subset $I \subseteq X$, the histogram $h(g)$, Tizhoosh (2005) intuitively proved that it is very easy to extend the previous concepts of FS (linear index of fuzziness proposed by Pal and Bezdek (1994)) for IVFS, and to define the (linear) *index of ultrafuzziness* as follows:

$$\Gamma(x) = \frac{1}{M.N} \sum_{g=0}^{G-1} [h(x).(\mu_U(x) - \mu_L(x))] \tag{6.3}$$

The previously defined IVFS is shifted over the interval $[0,\ G-1]$ of the histogram of the image by varying g but keeping σ fixed. Among the numerous frameworks of uncertainty modeling, this last equation seems to be an interesting tool for image processing, especially for noise estimation. The evolution of the global maximum $MAX(\Gamma)$ of this index according to the number of paths of the block 0 (Fig. 6.4) is presented Fig. 6.1. Using the same definition of the linear index of ultrafuzziness, we used this performing index to propose a generalization of Gaussian kernel filtering (Bigand and Colot 2010). In Delepoulle et al. (2011), we illustrated that $\Gamma(x)$ can be used as a global no-reference image metric when the image is clean and was compared with a psycho-visual score with success. In the following, we show that $\Gamma(x)$ increases if the image becomes less and less noisy and may be used in a noise estimation scheme.

6.3 IVFS for Image Noise Estimation

6.3.1 Design of the IVFS Image Noise Estimation

In this presentation, it is assumed that the noise corrupting image I is additive, stationary, and has zero mean (i.e., white noise):

$$I(o, p) = f(o, p) + \eta(o, p), \tag{6.4}$$

where f is the ideal image, I is the observed image (previously defined), and η is the noise component. The goal is to estimate the variance (power) $var(\eta)$ of η.

Many papers in literature present algorithms to estimate the features of noise in digital images (Jolion et al. 1990; Rank et al. 1999; Starck et al 1998; Deng et al. 2007). These methods attack the estimation problem in one of the two ways:

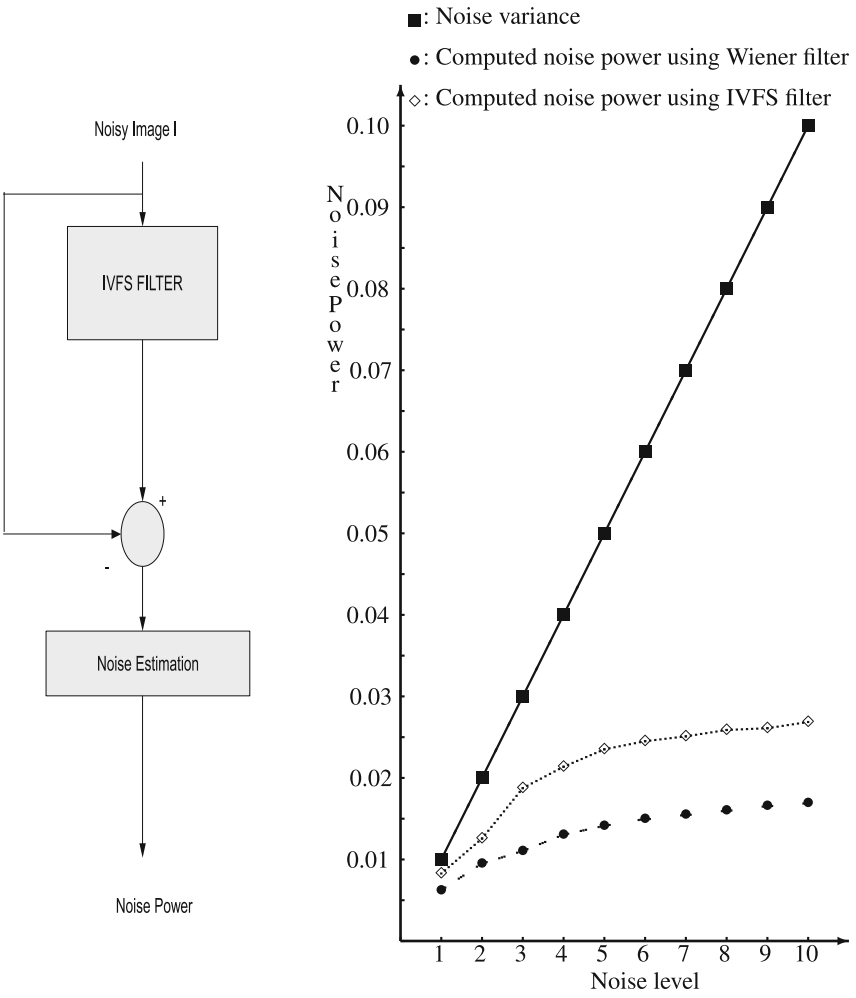
- by filtering I to suppress the image structure and then computing $var(\eta)$ from the filtered data
- by computing $var(\eta)$ from the variance of I in a set of image regions initially classified as showing little structure.

This work presents a method belonging to the first family, starting with the filtering of the image I. In Delepoulle et al. (2011), we defined a new image filter using ultrafuzziness index, to take into account simultaneously the local and global properties of the pixels. In the present work, we show that this index is also effective to measure noise level in computer-generated images and can be used in images noise estimation. By subtracting from I the filtered image J, a measure of the noise at each pixel is computed as presented Fig. 6.2.

6.3.2 Proposed Scheme

Let be a computer-generated image I. The global no-reference image metric taking into account noise levels is divided into three steps.

In the first one, image I is divided into Ki patches I_k (or blocks. A $M \times N$ gray-level image I is divided into Ki nonoverlapping patches ($m \times n$ gray-level patch), for example, a 512×512 image is divided into 16 nonoverlapping blocks of size 128×128). Each patch I_k is then analyzed using the local entropy Γ^k. To be coherent with the overall method, we used the IVFS denoising algorithm we proposed in Bigand and Colot (2010). This procedure is included in Algorithm 6 (this procedure makes it possible to obtain the optimal value σ of the Gaussian fuzzy number (see Eq. 6.1) for each patch). So the method operates like an unsupervised classification method that affects noise-free pixels to K clusters (modes of the image) and noisy pixels to a cluster: "noisy pixels". Let $g(o, p)$ denote the gray-level value of the pixel (o, p) of a $m \times n$ noisy block. The noise detection process results in dividing the



(a) Filtering-based noise estimation

(b) Noise power estimation for simulated noisy images

Fig. 6.2 Noise power estimation scheme

histogram into different zones, according to the maxima of ultrafuzziness Γ_{max}. So pixels are classified into two groups: noise-free pixels (i.e., belonging to one of the K modes) and noisy pixels. Noisy pixels are treated using the classical median filter ("Med", with a 3×3 window). The image restoration appears as follows:

$$g(o, p) = \begin{cases} g(o, p) & \text{if } g(o, p) \text{ is noise-free} \\ Med(o, p) & \text{if } g(o, p) \text{ is noisy.} \end{cases} \tag{6.5}$$

In the second step, once the denoising treatment on each block is made, the new image J is obtained from the concatenation of the K_i patches J_k. In the third step, once image I has been filtered into J, the noise power $var(\eta)$ is easily computed (see Algorithm 6).

6.3.3 Algorithm and Implementation

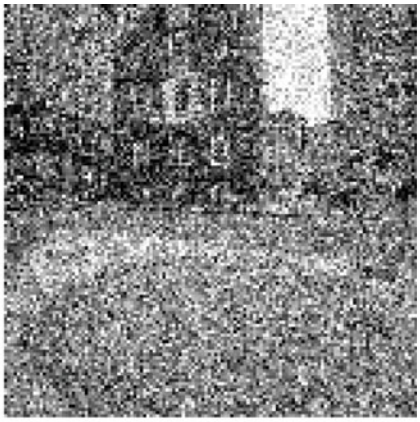
The image noise estimation based on IVFS and measure of ultrafuzziness Γ is given by Algorithm 6.

Algorithm 6 Image quality measure

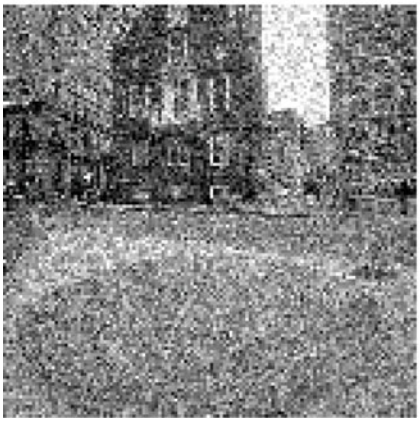
Require: an input noisy $M \times N$ gray-level image I, divided into Ki non-overlapping patches ($m \times n$ gray-level patch) and calculate the local entropy Γ^k for each patch k ($0 < k < Ki$)
1: Select the shape of MF
2: Compute the k-patch image histogram $h(g)^k$ (normalized to 1)
3: Initialize the position of the membership function
4: Shift the MF along the gray-level range
5: $\Gamma^k_{max} \leftarrow 0$
6: **for** each position g **do**
7: Compute $\mu_U(g)$ and $\mu_L(g)$
8: Compute $\Gamma(g)^k = \frac{1}{m \times n} \sum_{g=0}^{L-1} h(g)^k \times [\mu_U(g) - \mu_L(g)]$
9: **if** $\Gamma^k_{max} \leq \Gamma(g)^k$ **then**
10: $\Gamma^k_{max} \leftarrow \Gamma(g)^k$
11: **end if**
12: Keep the value Γ^k_{max} for patch k and optimal value σ
13: **end for**
14: Apply the FS parameter σ to get a denoised patch J_k
15: Iterate the number of the patch: $k + 1$
16: Compute the new image (J)
17: For each denoised patch, keep the local metric $\Gamma(g)^k$. Compute the value Γ for global image J with $\Gamma(g) = \frac{1}{K_i} \sum_{k=1}^{K_i} \Gamma(g)^k$
Ensure: The filtered image J
Ensure: The difference image $I - J$ and the **noise power estimation** $var(\eta)$

The implementation of image noise estimation based on IVFS is illustrated by the following experiment. Let us consider a synthetic test patch (classical "boston" image, see Fig. 6.3), belonging to the Categorical Image Quality (CSIQ) database (Wang et al. 2004) developed at Oklahoma State University. For the sake of brevity, only one representative image is presented here, and white Gaussian noise with different variances σ^2 added to this clean patch (the test patch with white Gaussian noise with $\sigma = 0.1$ is presented in Fig. 6.3). The noisy image is filtered using Wiener filter and IVFS filter and presented in Fig. 6.3. Figure 6.2 presents the evolution of

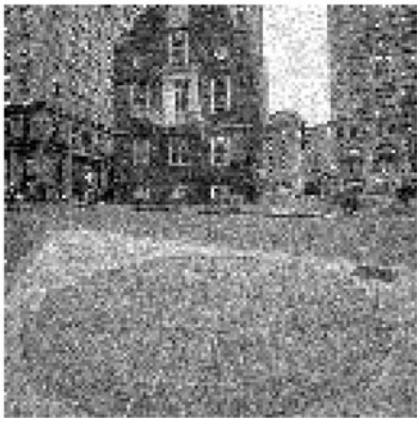
(a) Original Boston Image



(b) Noisy Image



(c) Filtered image using Wiener filter



(d) Filtered image using IVFS

Fig. 6.3 Obtained results with real image "Boston"

computed noise variances using the Wiener estimation method and the estimation method using IVFS filter we propose (100 simulations are carried out with independent noise realizations). From this set of experiments on the test patch, we can see that the value of the noise power $var(\eta)$ drops monotonically as the image content becomes more and more noisy. In other words, it can be thought that $var(\eta)$ is a better indicator of noise power than Wiener estimation (particularly for small noise power). These good results are in the same way that the results we presented in (Bigand and Colot 2010, 2012). They encouraged us to apply this method to the perceptual noise that flaws computer-generated images.

6.4 Experimental Results with a Computer-generated Image

6.4.1 Image Database for Noise Estimation

The model is built on data corresponding to images of globally illuminated scenes. The path-tracing algorithm was used in order to reduce noise (Shirley et al. 1996). This algorithm generates stochastic paths from the camera to the 3D scene. For each intersection of a path with the surface, a direction of reflection or refraction is randomly extracted.

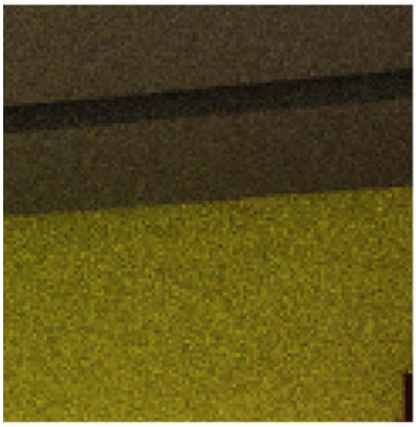
For each scene, several images were obtained, the first one being strongly noisy and the last one being the reference image (Fig. 6.4). Generated images were computed at 512×512 resolutions. The largest number of paths per pixel was set at 10100 which appeared to be sufficient for generating visually converged images. Then, each of these images is opaque and virtually cut into nonoverlapping sub-images of size 128×128 pixels. For the used test image, we thus get 16 different sub-images.

Our goal is to study and develop a noise estimation method devoted to synthetic images. Once noise power is estimated, we aim to propose a new image quality index based on image noise power (instead of classical image features) and compare this one with observed data (psycho-visual score) and classical reference quality metric like SSIM. In order to test the performance of the proposed technique, some results obtained with the computer-generated image named "Bar", Fig. 6.4, are shown in this presentation (other images were tested and same behaviors were observed, so they are not presented here due to lack of space). This image is composed of homogeneous and noisy blocks and is interesting to present some results.

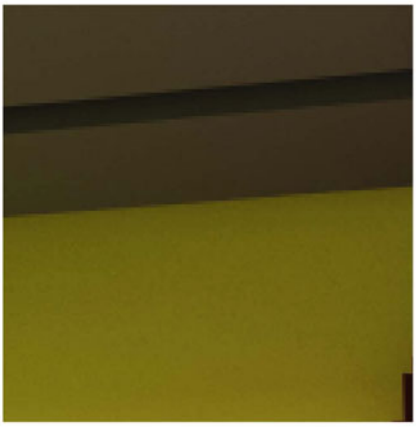
In Fig. 6.5, the first image (middle left) is the noisy block 10, obtained at the beginning of the process, and the second one (middle right) is the result obtained with the IVFS filter.

The **main idea** of the work is the following: synthesis process is started with a great noise level and a low entropy value. So the initial position of the synthesis process is unknown but the observed behavior measured at each iteration of the image synthesis process brings us information. The average information quantity gained at each iteration is entropy. The measured entropy using IVFS, named ultrafuzziness, seems to be an interesting measure of noise level and so supplies a noise power estimation, also used as denoising scheme in the proposed image synthesis process. The performances of the noise power estimation are now illustrated (particularly, noise histograms show the decreasing of noise power).

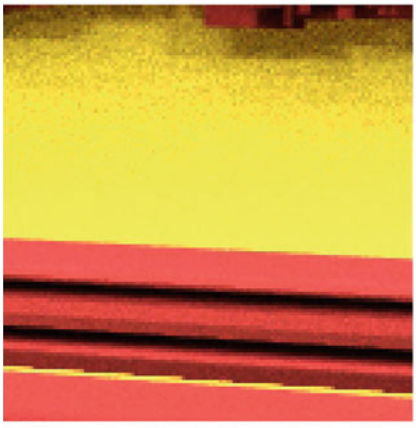
(a) Reference Image "'Bar"'



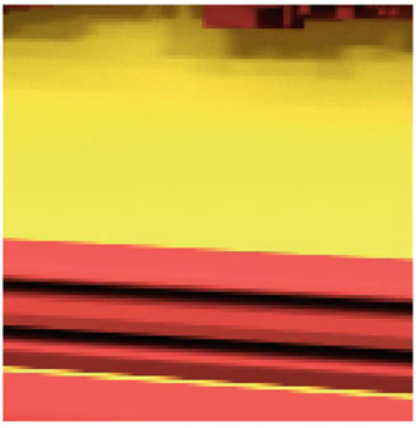
(b) Noisy Image, block 0



(c) Reference Image, block 0

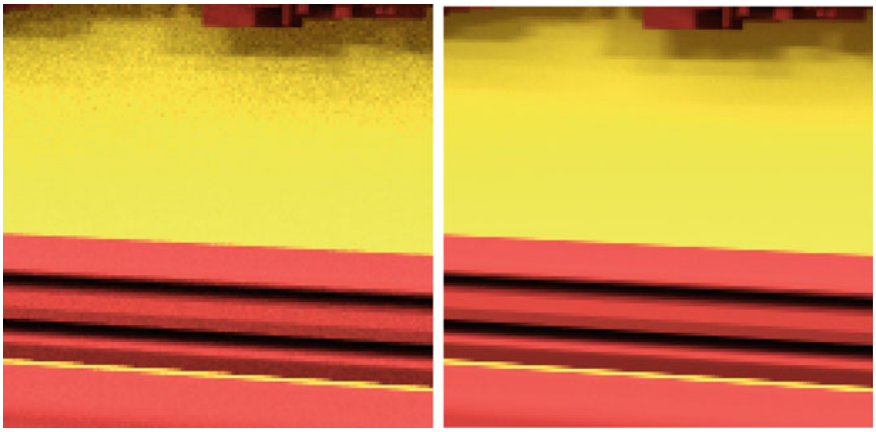


(d) Noisy Image, block 10

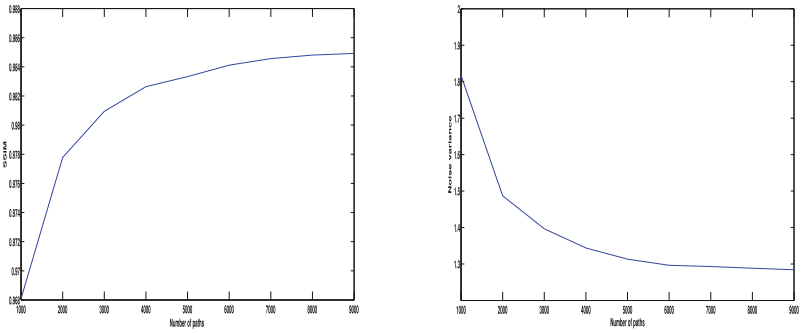


(e) Reference Image, block 10

Fig. 6.4 Original (noisy) and reference images



(a) Noisy Image, block 10 (b) Filtered Image, block 10

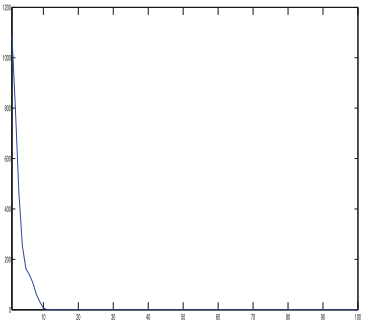


(c) SSIM evolution vs. number of paths (d) Noise variance evolution vs. number of paths

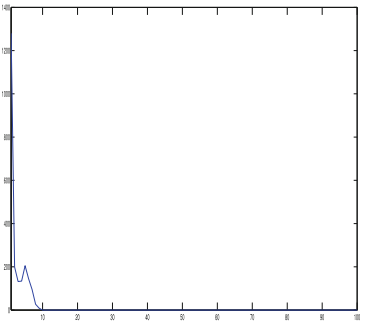
Fig. 6.5 Original and filtered blocks denoising using *IVFSs* and noise evolution

6.4.2 Performances of the Proposed Method

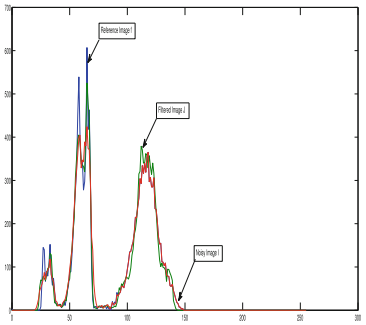
Synthesis process is started with a great noise level and a low entropy value but noise models are not available at the moment for computer-generated images. So a quantitative comparison is difficult to assume. In order to verify the effectiveness of the proposed noise model, a quantitative comparison with SSIM index has been made. The measure of structural similarity for images (SSIM quality index (Wang et al. 2004) has also been computed on the course of paths number as shown in Fig. 6.5. This measure is based on the adaptation of the human visual system to the structural information in a scene. The index accounts for three different similarity measures, namely luminance, contrast, and structure. The closer the index to unity, the better the result. It is easy to see that SSIM index is an opposite behavior according to the noise power measure we propose. This comparison is made at the bottom (c)



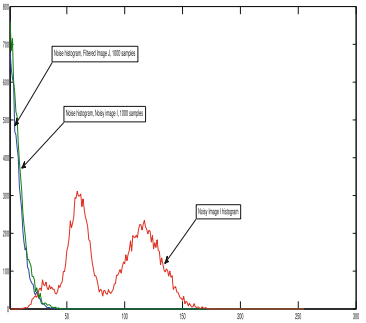
(a) Noise histogram, noisy image with 1000 samples



(b) Noise histogram, reference image



(c) Image histograms



(d) Noise histograms

Fig. 6.6 Synthetic image generation from noise point of view

of Fig. 6.5 for the noisy block 10 presented at the top of Fig. 6.5, and it is easy to verify that this behavior is assumed during image generation (from low sample number to high sample number), that is to say, from high noise level to low noise level (Fig. 6.5d).

To illustrate our work, we also present Fig. 6.6, for the noisy block 0, the noise histogram estimation (for 1000 samples and at the end of the process), the images I, J, f histograms (previously defined, IV), and the noise histograms (at 1000 samples).

We would like to highlight the advantages of the proposed noise power estimation: this method is simple; it is parameter free and avoid additional procedures and training data for parameter ($var(\eta)$) determination.

6.5 Conclusion

Starting from the filter presented in (Bigand and Colot 2010), taking into account the total amount of uncertainty present in computer-generated images using IVFS, a new noise estimation method is presented in this chapter. The method effectively combines image histogram information with the spatial information about pixels of different gray levels using an IVFS entropy technique. The good results we obtain are particularly interesting in image synthesis to model specific noise affecting this kind of images. This new method assumes no "a priori" knowledge of a specific input image, no learning, and no numerous tunable parameters, yet it has good performance compared to a psycho-visual method (Delepoulle et al. 2011). This qualitative noise model seems effective but there remain some open questions to effectively establish a link between the characteristics of noise affecting the image (noise level), and to optimize noise power computation, particularly for high noise level. Computer-generated image noise is not a Gaussian noise. So more extensive investigations on other measures of entropy and the effect of parameters influencing the width (length) of the interval of IVFS are under investigation (to establish a link between this interval and level and type of noise) toward an automatic noise quantification.

References

Babaud J, Witkin A, Baudin M, Duda R (1986) Uniqueness of the Gausssian kernel for scale-space filtering. IEEE Trans PAMI 8:26–33

Babu K, Sunitha K (2011) A new fuzzy Gaussian noise removal method for gray-scale images. IJCSIT 2(1):504–511

Bigand A, Colot O (2010) Fuzzy filter based on interval-valued fuzzy sets for image filtering. Fuzzy Sets Syst 161:96–117

Bigand A, Colot O (2012) Speckle noise reduction using an interval type-2 fuzzy sets filter. In: IEEE 6th international conference 'intelligent systems'

Bloch I (1996) Information combination operators for data fusion: a comparative review with classification. IEEE Trans SMC - Part B 26:52–67

Bustince H, Barrenechea E, Pergola M, Fernandez J (2009) Interval-valued fuzzy sets constructed from matrices: application to edge detection. Fuzzy Sets Syst 160:1819–1840

Delepoulle S, Bigand A, Renaud C (2011) Interval type-2 fuzzy sets based no-reference quality evaluation of synthesis images. In: Proceedings of CGVR'11

Delepoulle S, Bigand A, Renaud C (2012) A no-reference computer-generated images quality metrics and its application to denoising. In: IEEE intelligent systems IS'12 conference, vol 1, pp 67–73

Deluca A, Termini S (1972) A definition of a nonprobabilistic entropy in the setting of fuzzy set theory. Inf Control 20(4):301–312

Deng G, Tay DBH, Marusic S (2007) A signal denoising algorithm based on overcomplete wavelet representations and Gaussian models. Signal Process 87(5):866–876

Dubois D, Prade H (2005) Interval-valued fuzzy sets, possibility and imprecise probability. In: EUSFLAT- LFA 2005

Ferzli R, Karam L (2005) No-reference objective wavelet based noise immune image sharpness metric. In: International conference on image processing

Jolion J, Meer P, Rosenfeld A (1990) A fast parallel algorithm for blind estimation of noise variance. IEEE Trans PAMI 12:216–223

Jurio A, Paternain D, Lopez-Molina C, Bustince H, Mesiar H, Beliakov G (2011) A construction method of interval-valued fuzzy sets for image processing. In: Proceedings 2011 IEEE symposium on advances in type-2 fuzzy logic systems

Kaufmann A (1975) Introduction to the theory of fuzzy set—Fundamental theorical elements, vol 28. Academic Press, New York

Lahoudou A, Viennet E, Beghdadi A (2010) Selection of low-level features for image quality assessment by statistical methods. J Comput Inf Technol 2:183–189

Li Q, Wang Z (2009) Reduced-reference image quality assessment using divisive normalization based image representation. IEEE J Selected Topics Signal Process 3(2):202–211

Mendel J, John RB (2002) Type-2 fuzzy sets made simple. IEEE Trans Fuzzy Syst 10(2):117–127

Nachtegael N, Schulte S, der Weken DV, Witte VD, Kerre E (2005) Fuzzy filters for noise reduction: the case of Gaussian noise. In: Proceedings of International Conference on Fuzzy System

Olsen S (1993) Estimation of noise in Images: an evaluation. CVGIP 55:319–323

Pal N, Bezdek J (1994) Measures of fuzziness: a review and several classes. Van Nostrand Reinhold, New York

Rank M, Lendl M, Unbehauen R (1999) Estimation of image noise variance. In: Proceedings of the IEE visual signal process

Shirley P, Wang C, Zimmerman K (1996) Monte Carlo techniques for direct lighting calculations. ACM Trans Graph 15(1):1–36

Starck JL, Murtagh F, Gastaud R (1998) A new entropy measure based on the wavelet transform and noise modeling. IEEE Trans Circ Syst II 45:1118–1124

Sussner P, Nachtegael M, Esmi E (2011) An approach towards edge-detection and watershed segmentation based on an interval-valued morphological gradient. In: Proceedings of IPCV'11

Tizhoosh H (2005) Image thresholding using type-2 fuzzy sets. Pattern Recogn 38:2363–2372

Wang Z, Simoncelli EP (2005) Reduced reference image quality assessment using a wavelet-domain natural image statistic model. In: Proceedings of SPIE, conference on human vision and electronic imaging X, vol 5666, pp 149–159

Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. Trans Img Proc 13(4):600–612

Wilbik A, Keller JM (2012) A distance metric for a space of linguistic summaries. Fuzzy Sets Syst 208(1):

Wilbik A, MKeller J, (2012) A fuzzy measure similarity between sets of linguistic summaries. IEEE Trans Fuzzy Syst 21(1):

Zadeh L (1965) Fuzzy sets. Inf Control 8:338–353

Zadeh L (1968) Probability measures of fuzzy events. J Math Anal Appl 23

Zadeh L (1975) The concept of a linguistic variable and its application to approximate reasoning. Inf Sci 8:199–249

Zhang J, Ong S, Thinh M (2011) Kurtosis based no-reference quality assessment of jpeg2000 images. Sig Process Image Commun 26(1):13–23

Chapter 7
General Conclusion and Perspectives

7.1 Summary

High-quality computer-generated image generation techniques and the link between these techniques and perceptual noise are presented in the first two chapters (Chaps. 2 and 3). These chapters also present the high-quality images we can obtain with photo-realistic techniques on one hand and the difficulty to obtain a good visual quality assessment (particularly due to the complexity of the algorithms based on Monte Carlo methods) on the other hand.

HVS should be the better way to assess computer-generated images but the complexity of these techniques requires machine learning at the moment. Machine learning, which is a subfield of artificial intelligence (AI) makes it possible to extract knowledge from data. Thus, this is possible to obtain IQA from images, after a learning stage. The main advantages of the presented techniques are the following:

- Time to decide if an image is noisy or not is very short;
- IQA is obtained in a very short time in the same way; and
- If necessary these techniques can be used previously to an HVS assessment that makes it possible a short time to decide about IQA. Then, IQA can be refined in a classical way (MOS for example) to obtain very high-quality images

The main drawback of this approach is that important databases with high number of samples are needed to train learning methods. Classical classifiers (RBFs and SVMs) associated with noise features give good results for IQA (Chap. 4) and we also present RVM classifiers that give very good results due to their ability to deal with uncertainty brought by data. Then, new machine learning techniques (particularly inductive learning) are presented in the following chapter. This technique makes it possible to deal with huge image databases (using active learning and semi-supervised concepts) and large size images. For ending (Chap. 6), we present a no-reference method that gives good result using fuzzy entropy to set up noise level without noise features learning (fuzzy reasoning is another subfield of AI).

A. Bigand et al., *Image Quality Assessment of Computer-generated Images*,
SpringerBriefs in Computer Science, https://doi.org/10.1007/978-3-319-73543-6_7

7.2 Perspectives

A future research way could be using deep learning to better characterize noise features. In fact, Convolutional Neural Networks (CNN) can be used as feature extractor. Some authors have already used CNN to learn and extract classical features with success (leaf features for plant classification, special image features, etc.). This technique is under development to learn useful noise features directly from the color (computer-generated) images. It can be also applied to 3D (or stereoscopic) images, since CNN is less sensitive to the curse of dimensionality. The drawback of this technique is that a huge number of labeled samples are needed (labeled samples are costly to obtain in the case of image synthesis).

Considering 3D computer-generated images, the same questions are encountered as for 2D images. Modeling this kind of images is even more complex than for 2D computer-generated images. 3D databases are not yet sufficient (despite some 3D CAD existing databases). So we have to establish some new datasets to complement the learning base and validate our models. Then, applying learning models based on noise features is also more difficult because 3D noise features seem to depend on the considered point of view. Can we extend the different techniques presented in this book from 2D to 3D images? In particular, new perceptive noise thresholds need to be established. We also need to know if noise depends more on color representation or on luminance, and how to calibrate human tests. Thus, this is a new and interesting challenge. Uncertainty visualization techniques should certainly be of great interest for noise perception and description as for Perlin noise perception in natural-scene images. These techniques have to be investigated in parallel with machine learning in the spacial case of computer-generated images. It can bring a supplementary information crucial to visualize uncertain scalar information in complex 3D scenes. These challenges constitute an important future for the presented techniques.

Virtual reality (production line management in industry are now developed using this technique), augmented reality, and joint audiovisual quality metrics (movie production and interactive video games) are other domains where these techniques should be interesting to apply to other tasks and can be directly used. The presented methodology can be devoted to a large panel of applications in signal processing and image processing, and the authors sincerely hope that this book will be useful for a large number of people, practitioners, and researchers in these domains.