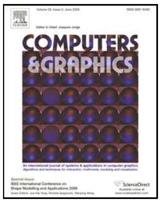
Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/results-in-physics



Perception of noise and Global Illumination: toward an automatic stopping criterion based on SVM

Nawel Takouachet[a], Samuel Delepoulle[b,c], Christophe Renaud[b,c], Nesrine Zoghlami[d], João Manuel R.S. Tavares[e,*]

[a] *Université Abbes Laghrour Khenchela, Algeria*
[b] *Université Lille Nord de France - F-59000 Lille*
[c] *ULCO, LISIC, F-62228 Calais, France*
[d] *Université Tunis EL Manar, Tunis, Tunisie*
[e] *Instituto de Ciência e Inovação em Engenharia Mecânica e Engenharia Industrial, Departamento de Engenharia Mecânica, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal*

ARTICLE INFO

Article history:
Received October 2, 2017

Keywords: Global Illumination, Noise Perception, Learning Methods

ABSTRACT

Unbiased global illumination methods based on stochastical techniques provide photo-realistic images. However, they are prone to noise that can only be reduced by increasing the number of processed samples. The problem of finding the number of samples that are required in order to ensure that most observers cannot perceive any noise is still an open issue. In this article, we address this problem focusing on visual perception of noise. However, rather than using known perceptual models, we investigate the use of learning approaches classically used in the field of Artificial Intelligence. Hence, we propose to use such approaches to create a model which is able to learn which image highlights perceptual noise. The learning is performed through the use of a database of examples based on experimentations of noise perception with human users. This model can then be used in any progressive stochastic global illumination method in order to find the visual convergence threshold of different parts of an input image.

1. Introduction

Stochastic global illumination methods have been proposed for over 20 years in order to provide an accurate simulation of high photo-realistic rendering. These methods are generally based on the Path Tracing method proposed by Kajiya [1], in which stochastic paths are generated from the camera's point of view towards the 3D scene. Since paths are chosen randomly, the gathering of light can change from one path to another generating high frequency color variations through the image [2]. However, the Monte Carlo theory ensures that this process will converge to the correct image when the number of samples, i.e. the paths, grows. However, no information is available about the number of samples required for the image to be visually converged. Indeed, the common final goal of these images is their analyze by the human visual system (HVS) which is very powerful. It requires numerous complex components of the human eye and brain sensitive to different kinds of properties, including contrast, spatial frequencies, shapes and colors. All this information is used by the brain to build a significant representation of what has been perceived. However, the HVS also has sensitivity limits. It carries out a fascinating strategy of compression and sensitivity thresholds. Therefore, we do not perceive equally all components of our environment, some parts of the visual information keep all our attention, while others are automatically and effortlessly ignored. As a consequence of these limits and of the high computational cost of global illumination algorithms, perception-driven approaches have been proposed. The main idea of such approaches is to replace the human observer by a computer vision model. By mimicking the HVS, such techniques can provide important improvements for rendering, and used for driving rendering algorithms to provide

[*] Corresponding author: Tel.: +351-22-508-1487; Tel.: +351-22-508-1445; *e-mail:* tavares@fe.up.pt (João Manuel R.S. Tavares)

visually satisfactory images and focus on visually important areas [3, 4, 5, 6, 7, 8, 9].

Most models based on the HVS provide interesting results, but are complex and still incomplete due to the internal system's complexity and its partial knowledge. Hence, generally, these models require relatively long computation times and are often difficult to use and to parameterize. Therefore, we investigate in this article the use of learning methods in order to improve these models. More specifically, we focus on the use of supervised learning in order to automatically detect the presence of noise in Path Tracing based methods. The learning is performed by using experimental data obtained from human users, which provides a model that can be used in progressive Path Tracing where sets of samples are progressively added. The model built is then interrogated after the computation of each sample set in order to know whether noise is still perceptible.

The remainder of this article is structured as follows. We review previous work on perceptual models and perception-driven rendering techniques in section 2. Then, in section 3, we summarize learning methods and present the one proposed. The different stages of our approach are described in section 4, and section 5 analyzes the problem of finding good inputs for the learning stage. Section 6 presents our results and comparisons with previous approaches. Then, the robustness of our model is investigated in section 7. We conclude the article with a discussion about the work presented and its future perspectives.

2. Perception overview

2.1. Perceptual models

Considerable research efforts have been devoted to understanding and simulating the human visual system behavior. This research showed that the HVS can fail to perceive certain physical inaccuracies and be very particularly sensitive to others [10, 11, 12, 5, 13, 8, 14].

Various perceptual models have been proposed. Some models use perceptual quality metrics that modulate the capacity of the visual system to detect differences between images. For example, the Visible Differences Predictor (VDP) model [10] predicts the probability of detecting differences between two images. It is based on frequency decomposition that extracts visual properties such as sensitivity to contrast and orientations. The Sarnoff Visual Discrimination Model (VDM) [15, 11] is also a well-known image comparison metric based on a set of complex sub-models that simulate several aspects of the human visual behavior. The VDM generates a visible differences map between two images, called Just Noticeable Difference (JND) for images corresponding to a probability of 75% that the differences are perceptible by the HVS.

Another kind of perceptual model is based on visual attention sights. Visual attention is the process of selecting a portion of the available visual information for locating, identifying or understanding objects in an environment. It allows the visual system to process visual inputs preferentially by shifting attention about an image, paying more attention to salient locations and less attention to unimportant regions [12, 16, 5, 9]. Recently, Wang *et al.* [17] proposed a para-boosting classifier that applies several saliency models to generate an improved saliency map.

2.2. Applications

By taking into account one or more characteristics of the human vision system, new perceptually-based techniques have been developed for rendering algorithms. The goal of such as techniques is to perform direct computations to achieve perceptual accuracy. In addition, the exploring of the HVS limits can considerably improve the rendering time by saving calculations in regions where the viewer is not able to detect differences between the refined image and the one built using a standard global illumination method.

Myszkowski [3] used the VDP principle to provide quantitative measures of perceptual convergence by predicting and estimating the perceivable differences between intermediate and final images. A similar approach was proposed by Takouachet and co-authors [18], where the VDP is used for estimating the differences between the initial very noisy image and the successive images of the progressive rendering process. In both approaches, the VDP only operates on the luminance channel and is costly to computed.

Yee [19] has proposed an improved version of the VDP in the same way as Ramasubramanian *et al.* [20], by discarding the orientation computation when calculating the spatial frequencies. These authors also extended the VDP by including the color domain in computing the differences between the images. This new version of the VDP increases the speed over the full VDP, which is especially observed when applied on a large set of images.

Pattanaik *et al.* [21] have introduced a new visual model for realistic tone reproduction. The suggested model is based on a multiscale representation of the luminance, pattern and color processing in the human visual system. The model takes into account changes in threshold visibility, visual acuity and color vision depending on the level of illumination in the scene under analysis.

Farrugia and Peroche [6] proposed a perceptually-based rendering method in which the rendering accuracy needed per pixel is adjusted according to a perceptual adaptive metric based on the Multiscale Model of Adaptation and Spatial Vision suggested in [21]. Various visual attention models [22, 23, 24] have been adapted in order to accelerate the global illumination computation in dynamic environments [10, 11, 15]. These models are used to estimate where computational efforts should be spent during the lighting solution. The rendering systems devote then more time to calculate the observed regions of interest.

Yee *et al.* [4] improved the bottom-up model of visual attention proposed by Itti and Koch in order to accelerate the global illumination computation in dynamic environments. The authors use an initial lighting approximation of the final image and apply the used model of visual attention with the addition of motion to locate important zones in an image. Hence, it is combined spatiotemporal sensitivity with a saliency-map to generate a spatiotemporal error tolerance map. This map, which is designated as Aleph Map, is then used to indicate where the computational efforts should be made during the lighting solution.

2.3. Advantages and drawbacks

Perceptual models are of great interest in Computer Graphics since this field of research is concerned with images that have to be visually analized. However, as aforementioned, the HVS is intricate and composite. Therefore, the majority of the perceptual models that have been applied in Computer Graphics have been simplified and/or modified relatively to the original models and to the "reality".

By taken into account the fact that the value of some parameters of Daly's VDP are unknown, Myszkowski [3] had to initialize and calibrate them in order to be usable in global illumination methods. Yee [25] proposed an abridged VDP version by removing some of the more expensive computations of Daly's algorithm and replacing them with approximations. Similar VDP simplifications were also proposed in [20]. The related original models were validated by neuro-biological and psychophysical studies; however, not all of the adopted simplifications and modifications were validated. For example, Longhurst and Chalmers [26] have shown through experimental results that the VDP does not always give accurate responses. The same issues arise for the saliency models [22, 23, 12], since their use in Computer Graphics has required some additional features, e.g., spatiotemporal sensibility as in [4], or modifications [9].

3. Learning methods

Learning methods, or machine learning, concern algorithms that are able to automatically improve their results over time. This improvement can be performed through results stored in databases, data produced by other programs or even by using the previous outputs of the used learning methods [27]. Over the last 50 years, Artificial Intelligence has provided many algorithms that are able to learn complex problems, including Artificial Neural Networks, Genetic Algorithms, Reinforcement Learning and Bayesian Learning. Recently, some research has been devoted to explore the potential of learning in Computer Graphics applications [28, 29, 30, 31]. Ren et *al.* [29] developed an image-based lighting model using neural networks. The authors applied an Artificial Neural Network (ANN) built from a small set of images to approximate light transport matrix as non-linear function of light source position and pixel coordinates. Nalbach et *al.* [30] proposed a high performance Convolution Neural Network (CNN) based screen space shader. These authors developed the CNN to synthesize and combine various visual features from a pixel-attributes map. In 2017, Satŷlmŷs et *al.* proposed an ANN-based model (CNN) in order to simulate sky illumination conditions with low requirements in terms of computation time and memory. Their model can be trained using either analytical or capture based inputs.

Our main goal in this work was to study and develop a logical component that should produce the same answer as a set of observers present relatively to noise present in images; particularly, the component should be able to classify images as noisy or not. In line with this, we tackled our problem as classification problem. CNN-based learning is particularly suitable for problems with invariant features, but is highly computational demanded. On the other hand, Support Vector Machine (SVM) based classifiers, which fail to learn complex invariances, produce good separator decisions in many classification problems with low computational demands [32, 33]. For this reason, we studied the use of a SVM based classifier as this type of learning techniques appeared to be suitably for our classification problem.

Support vector machines [34] are part of a set of supervised learning methods for regression and classification problems. SVMs compute hyper-planes that provide an optimal separation of data. Linear SVMs are known to be maximum margin classifiers. They also minimize classification errors. A main property of SVMs is their ability to work with large dimensional problems and to find complex separation planes: if the problem is not separable in the current space, the data can be projected in larger spaces where the separation could be easier by using kernel functions.

The advantages of such SVMs based approaches are that they can rely on real cases, meaning that the learning can be performed directly through the use of experimental data. They have also been shown to be robust, which is of great interest in our case when the images to be analyzed are not part of the learning image set. Finally, they are fast to use once the learning has accomplished. However, two main drawbacks of these approaches can be highlighted. On the one hand, the data that should be used in the learning step have to be carefully chosen in order to the model learn what is expected. On the other hand, these kinds of approaches provide a "black-box" model: it gives good answers, but it is often difficult to perceive how it learned and exactly what it learned.

4. Noise perception

Unbiased Global Illumination (GI) methods use randomly chosen paths for sampling the illumination of visible objects. This process generates stochastic noise that is commonly perceptible to human observers. *A posteriori* image denoising techniques are largely present in the literature [35, 36, 37, 38]. However, noise estimate models built from images are more complex. Some approaches exist using global measures such as Signal-to-Noise Ratio (SNR) [39] for quantifying noise in images. SNR is defined as the ratio of the mean pixel value to the standard deviation of the pixel values. Some other models focus on noise estimates [40, 41]. These models are based on basic functions of noise distributions like additive white Gaussian noise used in Information theory to simulate imperfections comes from many natural sources. However, in GI algorithms, noise is a stochastic process that arises from an unknown random distribution function. In 2015, Khademi et *al.* [28] proposed a supervised learning algorithm for fast filtering Monte Carlo noise. The authors trained a multilayer perceptron neural network coupled with a matching filter to learn the complex relationship between the noisy input scene data and the best filter parameters. However, these techniques are purely mathematical and do not take into account any properties of the HVS.

Recently VDP-based approaches [3, 18] have been applied for detecting perceptual noise in progressives GI methods. Even if the VDP is an HVS model, its application remains costly in terms of computation time. Furthermore, the VDP is not devoted to noise perception, it integrates all kind of differences between two images.

To our best knowledge, there is not any model able to detect and quantify stochastic visible noise in an image. The different steps of our approach proposed to overcome this problem based on supervised learning are described in the following section.

4.1. Overview

Our main goal is to mimic the human visual detection of noise in images using a model that has learned to detect this feature. We are interested here in supervised learning; that means that, initially, we have to provide to the model some examples of what we consider to be noisy images and noiseless images. Thus, the first part of our approach is to generate a database of examples and to teach the model about human judgment, i.e. noisy or not noisy image, using the chosen learning method. Hence, based on all imputed training examples, the chose learning method generates a model (left part of Figure 1) that is then used on the images to be analyzed, i.e. classified (right part of Figure 1).

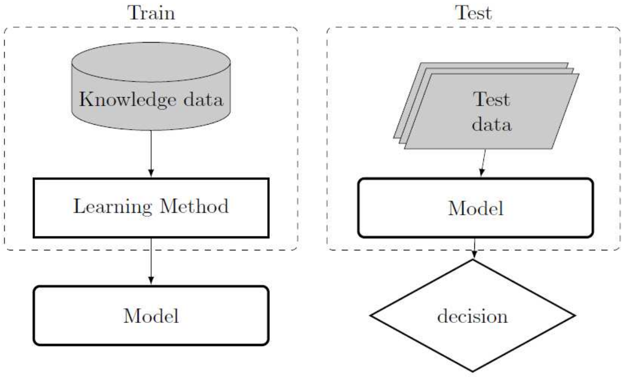


Fig. 1. The two steps of a computational classifier based approach: training the model using representative examples and then using the built model for obtaining a decision.

4.2. Data acquisition

4.2.1. Image dataset

The data used to build our model are images of globally illuminated scenes. We used as a first approach a Path Tracing with next event estimation [42], which computes several images from the same point of view by adding successively N new samples for each pixel. For each scene and each point of view, we thus have several images available, the first ones being very noisy, and the last ones being converged.

In our experiments, the images were computed at 512×512 pixel resolution, 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 to $10,000$, which appeared to be sufficient for generating visually converged images. All used images were tone mapped from the computed high dynamic range (HDR) values to low dynamic range (LDR) images using Reinhards tone mapping operator [43]. Figure 2 presents 6 of the scenes that were used during the model's learning stage. The used images address different illuminations, and various geometrical and texturing complexities. The remaining 6 scenes (see Appendix A) were used for validation purposes as is described later.

4.2.2. Acquiring users' data

Because we have to teach which image is noisy to the learning method, some experiments were necessary in order to determine the visual noise threshold for each image. However, considering the entire image for noise thresholding has two main drawbacks: On the one hand, it requires learning methods to work on very large datasets, which has been experimentally shown to reduce their learning efficiency. On the other hand, the noise is generally not equally perceived by human observers from different parts 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 processed for each pixel in the image. Therefore, we defined a simple protocol in which pairs of images are presented to the observer. One of these images is called the *reference image*, and was computed with $N_r = 10,000$ samples per pixel.

The second image, the so-called *test image*, is built as a stack of images, from very noisy ones above to converged ones below: by calling $N_{t,i}$ the number of samples in the stack's image i, with $i = 0$ at the top of the stack and $i = max = N_r$ at its bottom, we therefore, ensure the property $\forall i \in [0, max] : N_{t,i} < N_{t,i+1} \leq N_r$. Each of these images is opaque and virtually cut into non-overlapping sub-images of size 128×128 pixel. Hence, for our 512×512 pixel test images, we get 16 different sub-images for each of the stack's images.

During the experimentations, 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. Then, each point-and-click operation causes the

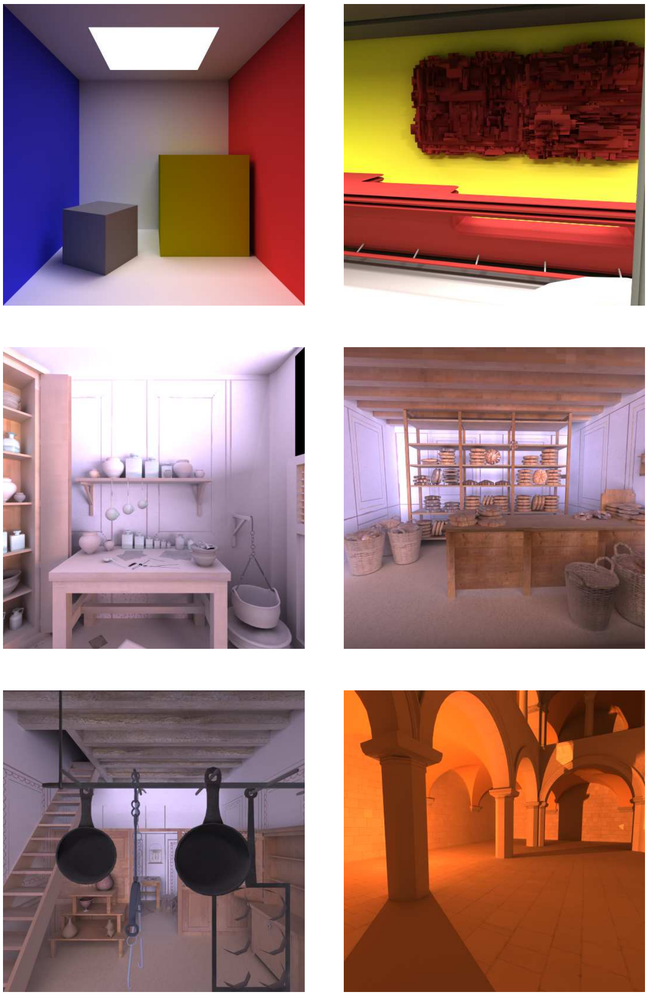
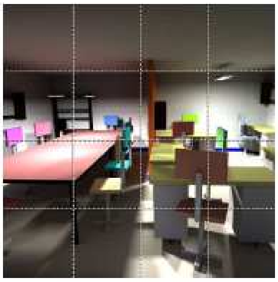


Fig. 2. The six different scenes used in the learning stage. From left to right and top to bottom: Cornell box, Taproom1, Oculist, Baker, Ironmonger, and Sponza. (N.b., the last four scenes are entirely textured.)

selection and the display of the corresponding $t + 1$ level sub-image, reducing visually the noise in this images sub-part accordingly. This operation is done until the observer considers that the two images are visually identical. Note that for reducing experiment artefacts, this operation is reversible, meaning that an observer is able to go down or up into the image's stack. The pair of images that is presented to the observer is chosen randomly, but we ensure that each pair is presented twice. Obviously, the sub-image grid is not visible and all the observers preformed the experiments under the same conditions, including the same display with identical luminance tuning and the same illumination conditions.

The results were recorded for 33 different observers and we computed the average number of samples $\tilde{N}$ that were required for each sub-image to be perceived as identical to the reference one by 95% of the observers. We obtained experimentally $\tilde{N} \in [1441, 6631]$, often with large differences between different parts of the same image, Figure 3.



2438	2356	2653	2607
1925	2045	2255	2369
2377	2549	2382	2439
2569	3236	2398	2602

Fig. 3. On the left, an example of the sixteen 128×128 **pixel sub-images of an image used during the experimentations with human observers; on the right, the grid drawn with the average number of samples required for** 95% **of the observers to consider that the corresponding sub-image is not noisy.**

4.3. Model building process

The experimental dataset can now be used for training a learning method to recognise noise in images. In the training protocol, we experimented image pairs; hence, pairs of sub-images were provided for the method: a sub-image known as *reference* and one of the test sub-images, Figure 4. Then, a third piece of information was combined with the sub-images: a binary value stating whether the two sub-images are considered as identical or not. All pairs of sub-images (S_{ref}, S_n) from the images shown in Figure 2 were thus successively provided for the learning method, with S_{ref} being the reference sub-image and S_n the corresponding potentially noisy sub-image computed with n samples per pixel ($n \in [100, 10000]$ by step of 100). According to the six full images presented in Figure 2 and their 16 sub-images, we were able to provide 9,600 pairs of sub-images for the learning model. This ensured a sufficient training dataset since SVMs are known to be efficient even on small sized example datasets.

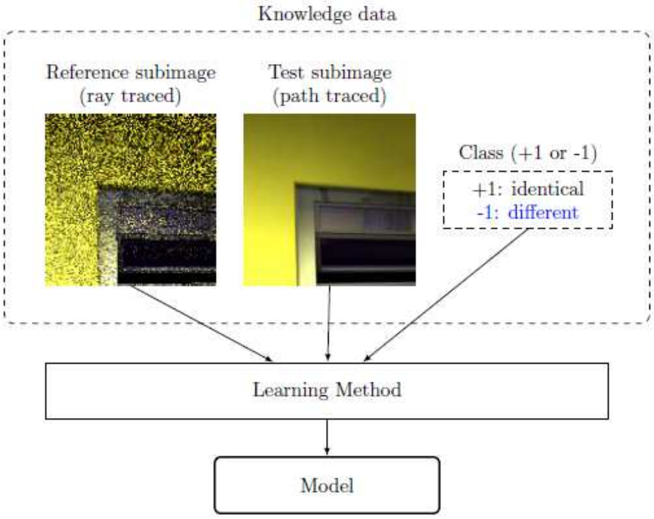


Fig. 4. The training protocol: pairs of sub-images were provided for the learning method with information about the fact that the two sub-images are identical or not from the noise point of view.

Ideally the reference sub-image should be the converged one. However, while using of the model in an iterative GI algorithm, this converged image is obviously not available. Thus, the reference used in both steps, i.e. learning and noise detection, is a quick ray traced image of each scene which highlights the same main features as the converged image: shadows, textures, reflections, etc. Note that the thresholds acquisition step uses the true converged images in order for observers to focus on noise and not on any other differences between ray traced and path traced images.

The library SVM^{light} was used for in our SVM-based experimentations. This library is an implementation of Vapnik's Support Vector Machine [34, 44] for problems of pattern recognition, regression, classification and for the issue of learning a ranking function.

In practice, each sub-image to be used is transformed into a 128 × 128 vector of luminance (one luminance per pixel), with noise being mainly characterized by luminance rather than chromaticities. The two vectors are provided to the SVM with the class the two images belong to: identical or different.

SVMs can be used with different kernels, for example, based on linear, polynomial or radial basis functions (RBF), which are used for projecting data into multidimensional spaces. RBF kernels are defined as: $K(x_i, x_j) = e^{-\gamma||x_i - x_j||^2}, \gamma > 0$, where x_i and x_j represent the values of the input space and C, γ are kernel parameters. It can be noted that parameter C in SVMs does not directly appear in the SVM kernel function. In effect, SVM returns the maximum margin for the linearly separable datasets in the kernel space. By cons, the data may be not linearly separable making the SVM problem unsolvable in the kernel space. To be solved, the nonlinear SVM problem can be formulated as a quadratic optimization problem subject to linear constraints. The idea is to map the datasets to a higher dimensional space using the kernel function, then solving it using some penalty parameter C. The datasets can then be linearly separated by allowing some misclassification error cost C for the samples violating the maximum margin.

This kind of kernel is widely used for the reduced set of parameters that have to be tuned because they provide robust learning models for many non-linear classification problems [45, 46, 47]. Nonetheless, we studied the use of several of those kernels and found that RBF kernels provided the best results using parameters $C = 2$ and $\gamma = 2$. The achieved classification accuracy was of 97.98% with a high number of support vectors, which is a good indicator of the learning model's efficiency.

4.4. Evaluation protocol

Once the training process has been performed, we obtain a model that is expected to identify whether a sub-image is noisy or not. In order to evaluate the capabilities of the model built, we use both training images and certain images that were not used in the training process (see *http://www-lisic.univ-littoral.fr/~delepoulle/CAG2017/cag.html*). We submit a pair of sub-images, designated as reference and test images, to the model which returns the probability P that the two images are identical or different. In our experimentations, we considered that two input images are identical when the model returns $P \geq 90\%$. Hence, the computational results obtained can be compared to those resulting from experimentations with human observers.

In our first approach, we trained the SVM classifier using pairs of the original sub-images without any pre-processing (Figure 4). This first training protocol returned P ¡ 23%, which means that the process failed to efficiently model a separator function. Furthermore, the classification appeared to be done not only based on the noise, but mainly based on the "geometric" content of the input images, i.e. parts of scenes that are "geometrically" similar are classified in the same manner, independently of being noisy or not. For example, some sub-images of the scenes Oculist, Baker and Ironmonger are considered as belonging to the same class, whether the images are noisy or not.

5. Modifying the model's inputs

Unfortunately, the results obtained with our first approach were clearly far from the human perception of noise. The problem is to find a solution to separate noise from other image features in order to be able to provide better information for the learning method. Our solution to make the notion of noise more explicit relatively to other properties of the images was to filter the original images using a noise mask and then re-train the SVM classifier on the resultant noise mask data instead of the original images.

5.1. Converting the images: Noise mask

The noise generated by stochastic GI methods can be characterized by high frequencies between values of adjacent pixels. In order to obtain a better characterization of noise, we propose converting each image into the frequency domain using the *Noise Mask*, which is commonly used in satellite imagery to denoise images [48, 49]. The use of a blur mask on an input image enhance the details and reduced the noise of the image; hence, the suggested process has two steps: Firstly, a blurred image is computed from the original image using a 3 × 3 Gaussian convolution with a convolution coefficient $\sigma \in [0.3, 1.5]$. Then, the corresponding noise mask is obtained by computing the difference between the original image and the blurred one, Figure 5. The noise is reduced by subtracting α times the noise mask from the original image:

$$Image_{\text{new}} = Image_{\text{original}} - \alpha \times Noisemask,$$

with $\alpha \in [5, 50]$ being the mask weight, which in common applications is equal to 30.

In this work, we were not interested in denoising the input images, but rather locating the areas where noise affects the images. Thus, we evaluated the use of the noise mask as the converting tool for the reference and test images that are subject to the learning process.

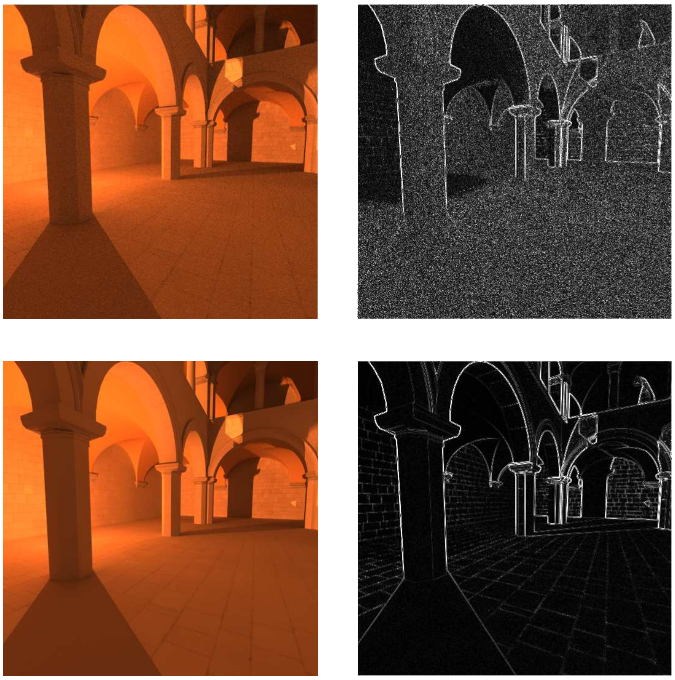


Fig. 5. **Examples of the results provided by the noise mask (right images) applied on an input image (left images): The noise mask preserves the outlines of objects and textures. The first noise mask (first row) was obtained by grouping the noise sub-masks applied on the noisy sub-images computed with 100 samples/pixel. The second noise mask (second row) corresponds to the reference sub-images computed with 10000 samples/pixel.**

5.2. *Experiments*

We applied our approach in a progressive path tracing algorithm where N new samples are added at each iteration to unconverged sub-images (with $N = 100$). The suggested approach uses two input images, the current sampled image, designated as *test image*, and the ray traced one, designated as *reference image*. Only the *test image* is changed progressively depending on the rendering step. At the first iteration of the rendering process, the reference and the test images are calculated and divided into 16 (128 × 128 pixel) sub-images, Figure 6. Then, a noise sub-mask is computed for each of the sub-images. Next, each pair of sub-masks, i.e. the sub-mask from the test sub-image and the sub-mask from the ray traced one (reference sub-image), are subjected to the learning model. Then, the model is asked whether each new sampled sub-image is still noisy. According to the model's answer, we then decide to add new samples or to stop computing for the corresponding sub-image as it is supposed to be visually converged. At the end of the process, the final image is assembled from the 16 converged sub-images. Note that questioning the model is very fast process, requiring only few milliseconds in a common computer.

6. Results

6.1. *Noise thresholds*

The rendering was performed on our 12 test scenes and the number of samples required for each sub-image was recorded. These values could then be compared to the experimental data presented in Section 4.2.2

In Table 1, are indicated the average number for samples per pixel of each entire image, i.e. scene, obtained in the experiments based on human observers and also by the computational model. From the results in Table 1, it can be concluded that the computational model obtained values very close to to experimental thresholds obtained with the human observers both for scenes used in the learning of the model (the first 6 cases) and for scenes that the it never learned with.

Scene	Exp.	Model
Oculist	3278	3287
Cornell box	2344	2300
Taproom 1	3234	3181
Baker	2215	2212
Ironmonger	2385	2381
Sponza	2900	2862
Deskroom1	3030	3012
Deskroom2	2481	2581
Taproom2	2816	2893
Classroom	2255	2300
Draper	2767	2737
Grocer	3168	2968

Table 1. Average number of samples per pixel required for each scene to be perceived as not noisy (exp.: experimental values obtained with human users; model: values obtained with the computational model built).

Figure 7 shows detailed results per sub-image for a scene that has not been used in the learning stage of our model. The first number indicated for each sub-image represents the average number of samples per pixel required for 95% of the human observers to see the sub-image as not noisy. On the other hand, the second number indicated is the number of samples processed by the computational model, which represents the stopping threshold for the iterative path tracing algorithm. These results confirm again that the stopping thresholds are generally very close to corresponding experimental values.

6.2. *Comparison with previous work*

Previous works attempted to discover how to efficiently stop the Path Tracing based methods. In [3] and [18], the authors suggested the VDP for identifying when noise is visually not perceptible. We compared the results of our approach with the ones found by these two approaches. Note that these approaches work by analysing the content of the entire image; hence, for simplicity of comparison, we compute the average number of samples per pixels required by our approach by averaging the thresholds of each sub-image. The results obtained for some of the input scenes by the experiments based on human observers, the suggested computational model, and the referred previous models, are represented in Figure 8.

It can be deduced from Figure 8 that the average number of pixels required by the suggested computational model was always

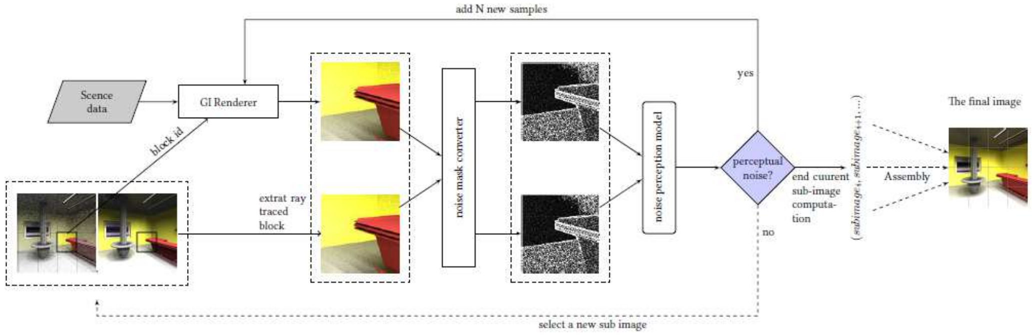
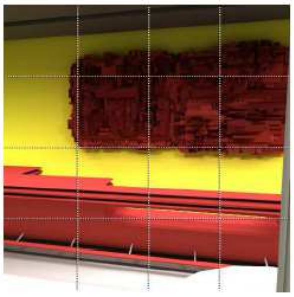


Fig. 6. A schematic representation of a Path Tracing progressive algorithm that includes the detection of noise in each 128×128 **sub-image.**



4579 /4600	6631 /6600	5818 /5700	5906 / 5900
3271 / 3200	2222 /2100	2261 /2100	2009 /1900
2400 /2500	2324 /2200	2098 /2100	2735 /2700
2955 /2900	2344 /2200	2002 /2000	2190 /2200

Fig. 7. The results of the built model on the scene ***Taproom1*****: the 16 sub-images (on the left); the number of samples (on the right) obtained experimentally with human users (on the top) and by the model (on the bottom).**

lower than the ones required by the previous approaches. Furthermore, the suggested model has the advantage of working on sub-images allowing therefore, a better distribution of the computation load. Actually, the approaches proposed in [3] and [18] require the use of the VDP, which is generally more costly to compute than the questioning of the model performed in our approach.

7. Generalization and robustness

Our approach is based on supervised learning, which involves, on one hand, difficulties to really understand what and how the learning occurred and, on the other hand, the use of an inevitably reduced set of examples. Therefore, we have assessed the generalization and robustness of the proposed approach.

7.1. Noise sensitivity

Both learning and tests were initially carried out with a PTWNE method. Hence, a question that can arise is whether the model built is able to identify noisy sub-images when these images are computed with another unbiased method. Therefore, we used our model as a "post-process" of the open-source *LuxRender* [50] software using its Metropolis Light Transport (MLT) [51, 52] approach capabilities. Contrary to path traced algorithms, the Metropolis algorithm avoids repeating the sampling process several times for each pixel. To initialize this algorithm, it is only necessary to trace one or two rays per pixel from the point of departure, i.e. the eye or the light source. Veach and Guibas used the principle of bidirectional path tracing [51, 53] to initialize the sampling, being the rays sampled from the eye and the source simultaneously, but the use of other methods is also possible. Then, new paths are generated by applying modifications, i.e. mutations, to the positions of a current path in order to explore its proximity. Indeed, mutations produce new random sequences of accessible states, i.e. Markov chains, from the current path. Sampling is performed by focusing on regions with high probability density, which correspond to the most important regions presented.

We used the open-source *LuxRender* [50] software according to the computation process described in Figure 6. The only difference is that the Metropolis approach does not ensure a fixed number of samples per pixel, but rather an average number of mutations per pixel for the entire sub-image. The computation loop in Figure 6 was thus slightly modified in order to ask MLT to add an average number of M mutations per pixel (we used $M = 50$ in our tests) for the sub-image under computation. At the beginning of the computation process, the whole image is rendered; then, the learning model is applied independently on each of the sub-images to detect the noise. If a subimage still contains perceptible noise, the MLT algorithm is asked to add an average number of M samples per pixel. Figure 9 presents results that were obtained for a view that was not used in the learning stage of the computacional model nor seen by the human observers before.

Since the experimental threshold values were measured for the PTWNE method, there is no relevant interest in comparing them to the thresholds provided by our approach applied to MLT. Hence, the validation was performed using a new experimentation involving users, as described in section 7.2. The view shown in Figure 9 is part of the set of views that were used during this experiment, whose related thresholds were validated by more than 95% of the users. Similar results were obtained for other views computed with MLT. Note that these results were obtained with the same model previously used in the PTWNE tests, i.e. it was not performed any new training of the computational model using the MLT computed images. This highlights the fact that the proposed model appears to be robust enough to

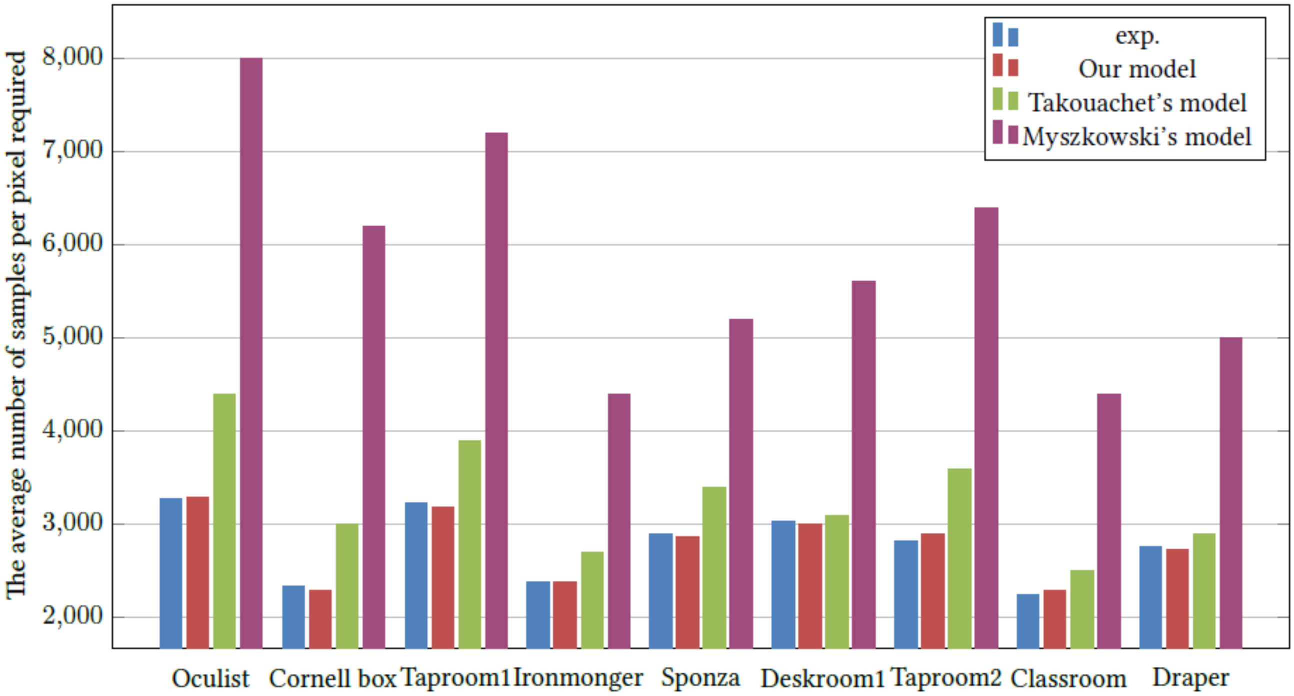
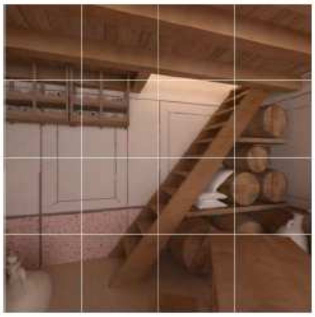


Fig. 8. Average number of samples per pixel required for some scenes to be perceived as not noisy (exp.: experimental values for 95% of the human observers to perceive the image as unnoisy; our model: values obtained with the suggested supervised learning model; Takouachet's and Myszkowski's models: results obtained using the models proposed in [18] and [3], respectively).



3900	3400	5300	5300
4100	3900	5000	5500
3300	3600	4300	6200
3800	4000	4700	5300

Fig. 9. Results of the convergence thresholds obtained by the proposed model when applied to a view of the Grocer scene using the MLT algorithm on the LuxRender software. (The numerical data in the grid shown on the right represent the average number of mutations per pixel used by LuxRender for the corresponding sub-image visible on the left.)

work efficiently with other unbiased algorithms.

7.2. Validation by users

In order to ensure that the images built from the independently computed sub-images are not visually affected and that the goal of finding a correct convergence threshold has been reached, we carried out a second experimentation with human users. The objective was to evaluate the sensitivity of the observers to noise in images assembled from the thresholds obtained by our computational model when it is driven by another unbiased algorithm than PTWE. This sensitivity was then compared to the one obtained with over-converged or very noisy images.

Several pairs of images belonging to three different classes were successively presented to the users:

- the reference class: the two images were the over-converged images computed with 10, 000 samples per pixel for PT images, or an average number of 10, 000 mutations for metropolis based images;
- the test class: one image was selected from the reference images and the other was the corresponding image built from sub-images considered as converged by our model;
- the noisy class: one image was selected from the reference images and the other was built from the sub-images that were obtained for 30% of the convergence threshold, which was found by dividing the number of samples required for convergence.

The pairs of images were presented to the users in random order, and each user was asked to decide whether the two images under comparison are identical or not. A total of 18 × 3 pairs of images were used, 13 pairs computed using the PTWNE method and 5 pairs using the LuxRender software based on the MLT algorithm, and 21 users took part in the experiment. The results obtained are presented in Table 2. From the data in Table 2, one can conclude that 96.8% of the users judged all pairs of the references images as identical, while 94.4% considered that two images from the test class were identical, i.e the images assembled from the thresholds obtained by our computational model and the reference images converged from PTWNE or MLT algorithms. Only 8.2% of the users had perceived the reference image and the noisy corresponding image as identical, which confirms once again the robustness of our model.

Class	Reference	Test	Noisy
identical	96.8 %	94.4 %	8.2 %

Table 2. Average percentage of users who considered as identical two images from the same class.

7.3. User sensitivity

The results presented were obtained based on a training database built from experimentations performed with computer science students. These students do not have any specific knowledge awareness with regard to computer graphics. We were therefore, interested in knowing whether computer arts people would perceive noise in the same sub-images and the difference in accuracy in their perception. Thus, we performed the same experiment, as described in Section 4.2, with students of a computer arts school. The results we obtained show that their sensitivity to noise is greater than of "classical" people. As a consequence, the noise thresholds were higher than those previously obtained. However, the results remain coherent with the previous ones; using a simple regression, we obtained for all the tested sub-images with a correlation coefficient of 0.78:

$$Threshold_{new} = 1.94 \times Threshold_{old} - 346,$$

with $Threshold_{old}$ being the threshold obtained with the experiment described in Section 4.2 and $Threshold_{new}$ the one obtained with computer arts students. This findings allows that final users of our approach can easily adjust the threshold sensitivity of the computational model according to the target audience.

7.4. Computation time

The building of the proposed model requires the computational processing of a large number of sub-images, which took here around 4 hours using the SVM^{light} library. However, this process is done offline and only once, being the resulting model ready to be applied on any computed image. Questioning the built model to know whether a sub-image is still noisy, requires only 0.038 seconds on a Intel Pentium R at 2 Ghz based computer. This is a very low additional computation time, even when repeated several times during the convergence of the underlying algorithm. Another additional requirement of our approach lies in the computation of the ray tracing reference image used. According to the view and the scene, this image requires between 1 and 10 minutes of additional computational time. However, this reference image needs to be computed only once, and its computation time is still moderate when compared to the number of hours required by the PTWNE and MLT algorithms to converge when applied on the same test scenes.

8. Conclusion

Path Tracing based methods provide unbiased and realistic images; however, these methods converge slowly, highlight noise during the convergence process, and should be stopped only when noise is no longer visually perceptible. Methods mimicking the Human Visual System could be interesting for this purpose, but are generally complex and difficult to develop and parametrize. In this article, we investigated a new approach based on a supervised learning technique for simulating noise perception in computed images. Our results are very promising since the stopping values provided by the computational model built are generally very close to the thresholds humanly defined. Additionally, it appears to be relatively robust, as it provides good values even for views that were not part of the learning dataset, or obtained using an illumination algorithm whose results were not learned by the model. Furthermore, it is simple and very fast once the learning step has been done.

Future work will investigate a solution to avoid the use of a ray traced reference image; for example, by using a new noise descriptor instead of the noise mask used here. On the one hand, this will simplify the use of our approach and, on the other hand, this would allow the model to handle highly indirect lighted environments more easily. Indeed, ray tracing images for such environments generally provide dark images whose use as reference images does not lead to efficient results. Another interesting problem to tackle would be on reducing the size of the used sub-images in order to decrease more efficiently the total number of required samples. However, this raises the issue of acquiring the corresponding experimental thresholds, which could be difficult. Our computational model gives good results when used with the PT and Metropolis global illumination algorithms, and we think that it will be also interesting to evaluate its efficiency when applied with other algorithms such as the Bidirectional PT algorithm. To further assess the generalization of our approach, it is required its application on a very large set of test images, which is always difficult to obtain because of the time required for modeling and scene rendering. As a solution to this difficult, we plan to develop a web interface devoted to the exploration of our model. This would allow that anyone can fully experience the results of the model on his/her own images, and also the continuous improvement of the computational model by providing to the model new learning data. Another interesting future work can be the application of CNNs algorithms to generate noise maps quantifying the noise value per pixel of input images.

Acknowledgements

The last author thanks the funding of Project NORTE-01-0145-FEDER-000022 - SciTech - Science and Technology for Competitive and Sustainable Industries, co-financed by "Programa Operacional Regional do Norte" (NORTE2020), through "Fundo Europeu de Desenvolvimento Regional" (FEDER).

APPENDIX

Figure 10 presents the six additional images that were only used here for testing purpose. Note that these images correspond to a total of $16 \times 6 = 96$ sub-images.

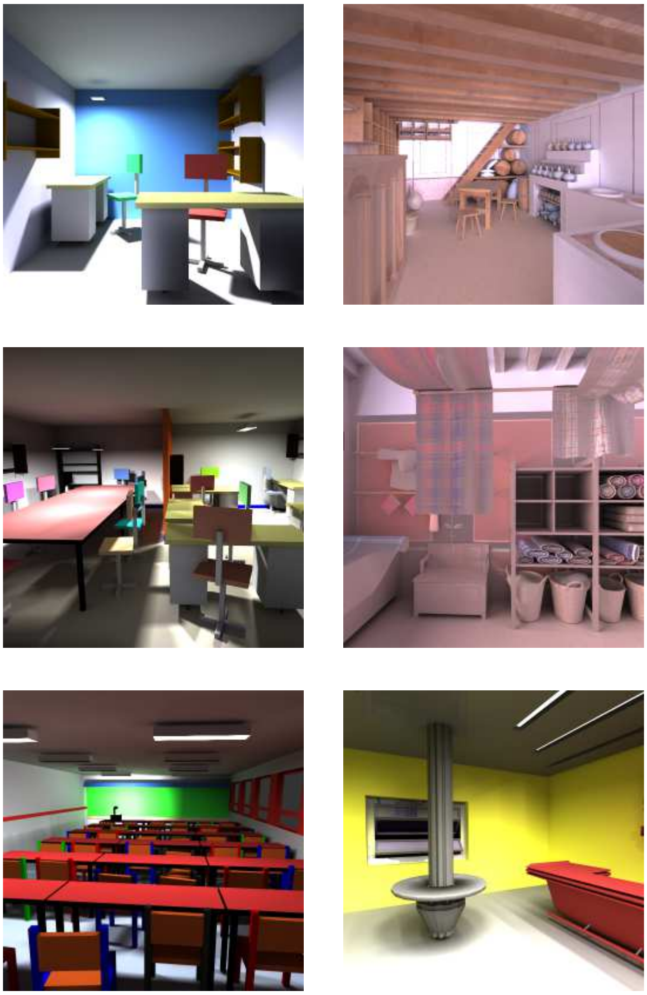


Fig. 10. The six scenes used for testing the learning models.

References

[1] Kajiya, J. The rendering equation. ACM Computer Graphics 1986;20(4):143–150.

[2] Szirmay-Kalos, L. Stochastic methods in global illumination - state of the art report. Tech. Rep. TR-186-2-98-23; Institute of Computer Graphics and Algorithms, Vienna University of Technology; Favoritenstrasse 9-11/186, A-1040 Vienna, Austria; 1998. Human contact: technical-report@cg.tuwien.ac.at.

[3] Myszkowski, K. The visible differences predictor: applications to global illumination problems. In: Eurographics Rendering Workshop. 1998, p. 233–236.

[4] Yee, H, Pattanaik, S, Greenberg, DP. Spatiotemporal sensitivity and visual attention for efficient rendering of dynamic environments. In: ACM Transactions on Graphics. ACM Press; 2001, p. 39–65.

[5] Sundstedt, V, Chalmers, A, Cater, K, Debattista, K. Top-down visual attention for efficient rendering of task related scenes. In: Proceedings of the Vision, Modeling, and Visualization Conference 2004 (VMV 2004), Stanford, California, USA, November 16-18, 2004. 2004, p. 209–216.

[6] Farrugia, JP, Peroche, B. A Progressive Rendering Algorithm Using an Adaptive Perceptually Based Image Metric. Computer Graphics Forum 2004;doi:10.1111/j.1467-8659.2004.00792.x.

[7] Mitchell, DP. Generating antialiased images at low sampling densities. In: SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques. New York, NY, USA: ACM Press. ISBN 0-89791-227-6; 1987, p. 65–72.

[8] Tumblin, J, Rushmeier, H. Tone reproduction for realistic images. IEEE Comput Graph Appl 1993;13(6):42–48. doi:10.1109/38.252554.

[9] Longhurst, P, Debattista, K, Chalmers, A. 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. ACM SIGGRAPH. ISBN 1-59593-288-7; 2006, p. 21–29.

[10] Daly, S. The visible differences predictor: an algorithm for the assessment of image fidelity. In: Digital images and human vision; vol. 4. 1993, p. 124–125.

[11] Sarnoff Corporation, . Sarnoff JND vision model algorithm description and testing. 1997. VQEG.

[12] Niebur, E, Itti, L, Koch, C. Controlling the focus of visual selective attention. In: Hemmen, LV, Domany, E, Cowan, J, editors. Models of Neural Networks IV. Springer Verlag; 2001,.

[13] Niebur, E, Koch, C. Computational architectures for attention. In: Parasuraman, R, editor. The Attentive Brain. MIT Press; 1998, p. 163–186.

[14] Harvey, C, Debattista, K, Bashford-Rogers, T, Chalmers, A. Multi-modal perception for selective rendering. Comput Graph Forum 2017;36(1):172–183. doi:10.1111/cgf.12793.

[15] Lubin, J. A visual discrimination model for imaging system design and evaluation. In: Peli, E, editor. Vision Models for Target Detection and Recognition. World Scientific; 1995, p. 245–283.

[16] Itti, L. Models of bottom-up and top-down visual attention. Ph.D. thesis; 2000.

[17] Wang, J, Borji, A, Kuo, CCJ, Itti, L. Learning a combined model of visual saliency for fixation prediction. IEEE Transactions on Image Processing 2016;25(4):1566–1579.

[18] Takouachet, N, Delepoulle, S, Renaud, C. A perceptual stopping condition for global illumination computations. In: Proc. Spring Conference on Computer Graphics 2007. Budmerice, Slovakia; 2007, p. 61–68.

[19] Yee, H. Perceptual metric for production testing. Journal of Graphics Tools 2004;9(4):33–40. doi:10.1080/10867651.2004.10504900.

[20] Ramasubramanian, M, Pattanaik, SN, Greenberg, DP. A perceptually based physical error metric for realistic image synthesis. In: Rockwood, A, editor. Siggraph 1999, Computer Graphics Proceedings. Los Angeles: Addison Wesley Longman; 1999, p. 73–82.

[21] Pattanaik, SN, Ferwerda, JA, Fairchild, MD, Greenberg, DP. A multiscale model of adaptation and spatial vision for realistic image display. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '98; New York, NY, USA: ACM. ISBN 0-89791-999-8; 1998, p. 287–298. doi:10.1145/280814.280922.

[22] Treisman, AM, Gelade, G. A feature-integration theory of attention. Cognit Psychol 1980;12(1):97–136.

[23] Koch, C, Ullman, S. Shifts in selective visual attention: Towards the underlying neural circuitry. Human Neurobiology 1985;4:219–227.

[24] Niebur, E, Itti, L, Koch, C. Controlling the Focus of Visual Selective Attention. New York, NY: Springer New York. ISBN 978-0-387-21703-1; 2002, p. 247–276. doi:10.1007/978-0-387-21703-1_6.

[25] Yee, H. A perceptual metric for production testing. journal of graphics tools 2004;9(4):33–40.

[26] Longhurst, P, Chalmers, A. User validation of image quality assessment algorithms. In: TPCG '04: Proceedings of the Theory and Practice of Computer Graphics 2004 (TPCG'04). Washington, DC, USA: IEEE Computer Society. ISBN 0-7695-2137-1; 2004, p. 196–202. doi:http://dx.doi.org/10.1109/TPCG.2004.39.

[27] Mitchell, T. Machine Learning. McGraw Hill; 1997.

[28] Kalantari, NK, Bako, S, Sen, P. A Machine Learning Approach for Filtering Monte Carlo Noise. ACM Transactions on Graphics (TOG) (Proceedings of SIGGRAPH 2015) 2015;34(4).

[29] Ren, P, Dong, Y, Lin, S, Tong, X, Guo, B. Image based relighting using neural networks. ACM Trans Graph 2015;34(4):111:1–111:12. doi:10.1145/2766899.

[30] Nalbach, O, Arabadzhiyska, E, Mehta, D, Seidel, H, Ritschel, T. Deep shading: Convolutional neural networks for screen-space shading. CoRR 2016;abs/1603.06078.

[31] Satỳlmỳs, P, Bashford-Rogers, T, Chalmers, A, Debattista, K. A machine-learning-driven sky model. IEEE computer graphics and applications 2017;37(1):80–91.

[32] Huang, FJ, LeCun, Y. Large-scale learning with svm and convolutional nets for generic object categorization. In: Proc. Computer Vision and Pattern Recognition Conference (CVPR'06). IEEE Press; 2006,.

[33] Nagi, J, Caro, GAD, Giusti, A, Nagi, F, Gambardella, LM. Convolutional neural support vector machines: Hybrid visual pattern classifiers for multi-robot systems. In: ICMLA (1). IEEE. ISBN 978-1-4673-4651-1; 2012, p. 27–32.

[34] Vapnik, V. The Nature of Statistical Learning Theory. New York: Springer-Verlag; 1995.

[35] Kim, J, Park, H. Adaptive 3-d median filtering for restoration of an image sequence corrupted by impulse noise. SP:IC 2001;16(7):657–668.

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

[37] Hore, ES, Qiu, B, Wu, HR. An adaptive filter for image denoising using fuzzy inference. In: Signal and Image Processing (SIP 2003), Proceedings of the IASTED International Conference, August 13-15, 2003, Honolulu, HI, USA. 2003, p. 479–484.

[38] Priyanka, K, Versha, R. A brief study of various noise model and filtering techniques. Journal of Global Research in Computer Science 2013;4(4):166–171.

[39] Corsini, G, Mossa, A, Verrazzani, L. Signal-to-noise ratio and autocorrelation function of the image intensity in coherent systems. sub-rayleigh and super-rayleigh conditions. IEEE Transactions on Image Processing 1996;5(1):132–141.

[40] Starck, JL, Murtagh, F. Automatic noise estimation from the multiresolution support. PASP 1998;110:193–199.

[41] Starck, JL, Murtagh, F, Gastaud, R. A new entropy measure based on the wavelet transform and noise modeling. IEEE Transactions on Circuits and Systems II 1998;45.

[42] Shirley, P, Wang, C, Zimmerman, K. Monte carlo techniques for direct lighting calculations. ACM Transactions on Graphics 1996;15(1):1–36.

[43] Reinhard, E, Stark, M, Shirley, P, Ferwerda, J. Photographic tone reproduction for digital images. In: Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '02; New York, NY, USA: ACM. ISBN 1-58113-521-1; 2002, p. 267–276. doi:10.1145/566570.566575.

[44] Joachims, T. Estimating the generalization performance of a SVM efficiently. In: Langley, P, editor. Proceedings of ICML-00, 17th International Conference on Machine Learning. Stanford, US: Morgan Kaufmann Publishers, San Francisco, US; 2000, p. 431–438.

[45] C. Huang, LSD, Townshend, JRG. An assessment of support vector machines for land cover classification. International Journal of Remote Sensing 2002;23:725–749(25).

[46] Melgani, F, Bruzzone, L. Classification of Hyperspectral Remote Sensing Images With Support Vector Machines. IEEE Transactions on Geoscience and Remote Sensing 2004;42:1778–1790.

[47] Liu, Q, Zhang, Y, Hu, Z. Extracting positive and negative association classification rules from rbf kernel. In: ICCIT '07: Proceedings of the 2007 International Conference on Convergence Information Technology. Washington, DC, USA: IEEE Computer Society. ISBN 0-7695-3038-9; 2007, p. 1285–1291.

[48] Russ, JC. The Image Processing Handbook. CRC Press; 1992.

[49] Dragesco, J. High resolution astrophotography. Cambridge: Cambridge University Press; 1995.

[50] Luxrender. 2011. URL: http://www.luxrender.net/.

[51] Veach, E, Guibas, LJ. Metropolis light transport. Computer Graphics 1997;31(Annual Conference Series):65–76.

[52] Hoberock, J, Hart, JC. Arbitrary importance functions for metropolis light transport. In: Computer Graphics Forum; vol. 29. Wiley Online Library; 2010, p. 1993–2003.

[53] Veach, E. Robust monte carlo methods for light transport simulation. Ph.D. thesis; Stanford, CA, USA; 1998. AAI9837162.