

Physics-Motivated Features for Distinguishing Photographic Images and Computer Graphics

Tian-Tsong Ng, Shih-Fu Chang
Jessie Hsu, Lexing Xie
Department of Electrical Engineering
Columbia University
New York, NY 10027

{ttng,sfchang,yfhsu,xlx}@ee.columbia.edu

Mao-Pei Tsui
Department of Mathematics
University of Toledo
Toledo, OH 43606
mao-pei.tsui@utoledo.edu

ABSTRACT

The increasing photorealism for computer graphics has made computer graphics a convincing form of image forgery. Therefore, classifying photographic images and photorealistic computer graphics has become an important problem for image forgery detection. In this paper, we propose a new geometry-based image model, motivated by the physical image generation process, to tackle the above-mentioned problem. The proposed model reveals certain physical differences between the two image categories, such as the gamma correction in photographic images and the sharp structures in computer graphics. For the problem of image forgery detection, we propose two levels of image authenticity definition, i.e., imaging-process authenticity and scene authenticity, and analyze our technique against these definitions. Such definition is important for making the concept of image authenticity computable. Apart from offering physical insights, our technique with a classification accuracy of 83.5% outperforms those in the prior work, i.e., wavelet features at 80.3% and cartoon features at 71.0%. We also consider a recapturing attack scenario and propose a counter-attack measure. In addition, we constructed a publicly available benchmark dataset with images of diverse content and computer graphics of high photorealism.

Categories and Subject Descriptors: I.4.10 [Image Processing and Computer Vision]: Image Representation – *Statistical*; K.4.4 [Computers and Society]: Electronic Commerce – *Security*

General Terms: Experimentation, Security.

Keywords: Natural image statistics, computer graphics, fractal, differential geometry, image forensics, steganalysis, image authentication.

1. INTRODUCTION

Today, the level of photorealism achievable by the state-of-the-art computer graphics is so convincing that we feel like

we are watching real-life events even for a movie produced by computer graphics effects. Hence, detecting photorealistic computer graphics and separating them from photographic images (e.g. images captured by cameras) has become an important issue in several applications such as criminal investigation, journalistic reporting, and intelligence analysis.

Despite the fact that classification of photographic images and computer graphics has been applied for improving the image and video retrieval performance [5, 13], classification of photographic images (PIM) and photorealistic computer graphics (PRCG) is a new problem. The work in [7] takes advantage of the wavelet-based natural image statistics, and extract the first four order statistics of the in-subband coefficients and those of the cross-subband coefficient prediction errors as features for classifying PIM and PRCG. Promising results, with a PIM detection rate of 67% at a 1% false alarm rate, have been achieved. *However, due to the lack of a physical model for PIM and PRCG, the results have not led to an insight into the question: How PIM are actually different from PRCG?*

In this paper, we propose a new geometry-based image model which is inspired by the physical generation process of PIM and PRCG. Mandelbrot [8] introduced the idea of fractal as a geometric description of a mathematical object with a fractional dimension to generalize the classical geometry which is limited to integer dimensions. He also pointed out that, unlike the ideal fractal which is a mathematical concept, the geometry of real-world objects are often best characterized by having different dimensions over different range of scales. This insight inspires our image description scheme: at the finest scale, we describe the intensity function of an image as a fractal, while at an intermediate scale as a 2D (dimensional) topological surface, which is best described in the language of differential geometry. Additionally, we also model the local geometry of the image intensity function in a “non-parametric” manner by the local image patch. We will show that the geometry-based approach captures the characteristics of the physical image generation process of PIM, such as the gamma correction used in cameras, and those of PRCG, such as the simplified 3D object model. We are thereby able to at least partially explain the actual differences between PIM and PRCG.

In the process of developing this work, we collected 800 personal PIM (*personal*), 800 PIM from Google Image Search (*Google*), 800 PRCG from various 3D artist websites (*CG*), and 800 photographed screen display of the PRCG (*recaptured CG*), for our experiments. We only focus on highly

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MM'05, November 6–12, 2005, Singapore.

Copyright 2005 ACM 1-59593-044-2/05/0011 ...\$5.00.



Figure 1: Examples from our image sets. Note the photorealism of all images.

photorealistic computer graphics and exclude those that are trivially detectable. Figure 1 shows some examples of our image sets. Note that, there could be artificial objects in the PIM, and conversely, nature scenes in the PRCG. For the benefit of the research community, we will release the images as an open dataset, which will eventually contain the personal PIM acquired by the authors and the URL of other images, due to the copyright issue.

Our main contribution in this paper is the proposed geometry-based image description scheme that uncovers the physical differences between PIM and PRCG. Other contributions include the proposal of a two-level definition of image authenticity making image authenticity computable, the creation of an open dataset, and our effective classification scheme for PIM and PRCG. In Section 2, we give the two-level definition of image authenticity. In Section 3, we explore the differences between the physical image generation process of PIM and PRCG. Then, we describe in detail the local patch statistics, the linear Gaussian scale-space, the fractal geometry, the differential geometry, the dataset, and the feature extraction process by rigid-body moments in the subsequent sections. Finally, we present our classification results and a discussion in Section 10, followed by the conclusions.

2. DEFINITION OF AUTHENTICITY

It may not be obvious that image authenticity is a non-absolute idea. For example, computer graphics, being synthetic images, are generally accepted to be inauthentic, but what about photographs of the computer graphics? In this case, the line of authenticity becomes vague; while the photographed computer graphics is actually a photograph, its content may still violate the physics of a real scene, such as having a misplaced shadow or inconsistent lighting. Therefore, we herein propose two levels of definition for image authenticity: the *imaging-process authenticity* and the *scene authenticity*. Images acquired by image acquisition devices such as cameras and scanners are considered to be imaging-process-authentic. Of course, we can further consider an image to be camera-authentic or scanner-authentic. The quality of imaging-process authenticity will be shown in this paper to be the main difference between PIM and PRCG. Secondly, an image of a real world scene that is a result of the physical light transport (i.e., a snapshot of the physical light field) is considered to be scene-authentic. With these definitions, an image can be camera-authentic but may not be scene-authentic, just as the case of the recaptured computer graphics, and vice-versa. Therefore, the final definition of image authenticity would be application-specific and can be obtained by combining or subdividing the two-level authenticity definitions. In this paper, we will evaluate our results with respect to these two definitions of image authenticity.

3. IMAGE GENERATION PROCESS

In general, the image intensity function $I : (x, y) \subset \mathbb{R}^2 \mapsto \mathbb{R}$ arises from a complex interaction of the object geometry, the surface reflectance properties, the illumination and the camera view point. In addition, as photographic or scanned images are captured by an image acquisition device such as a camera or a scanner, they also bear the characteristics of the device. For example, a digital photographic image in general has undergone the optical lens transformation, the gamma correction, the white-balancing and the color-processing while being tinted with the quantization noise and the sensor fixed pattern noise [16].

However, PRCG is produced by a graphics rendering pipeline [1], a different process than the that of the PIM. In general, a graphics rendering pipeline can be divided into three conceptual stages: application, geometry and rasterizer. At the application stage, mainly implemented in software, the developer designs/composes the objects/scene to be rendered. The objects are represented by the rendering primitives such as points, lines and triangles. The geometry stage, mainly implemented in hardware, consists of rendering operations on the rendering primitives. The rasterizer stage is responsible for converting the rendered primitives into pixels which can be displayed on a screen. During this conversion, the camera effect, such as the depth-of-field (DoF) effect or the gamma correction, may or may not be simulated. The main differences between the PIM and PRCG image generation processes are below:

1. **Object Model Difference:** The surface of real-world objects, except for man-made objects, are rarely smooth or of simple geometry. Mandelbrot [8] has showed the abundance of fractals in nature and also related the formation of fractal surfaces to basic physical processes such as erosion, aggregation and fluid turbulence. However, the computer graphics 3D objects are often represented by the polygonal models. Although the polygonal models can be arbitrarily fine-grained, it comes with a higher cost of memory and computational load. Furthermore, such a polygonal model is not a natural representation for fractal surfaces [11]. A coarse-grained polygonal model may be used at the perceptually insignificant area for saving computational resources.
2. **Light Transport Difference** [1]: The physical light field captured by a camera is a result of the physical light transport from the illumination source, reflected to the image acquisition device by an object. The precise modelling of this physical light transport involves an 8D function of the object’s reflectance property, hence its simulation requires substantial computational resources. Therefore, a simplified model based on the assumption of isotropy, spectral independence and parametric representation is often used.
3. **Acquisition Difference:** PIM carry the characteristics of the imaging process, while PRCG may undergo different types of post-processing after the rasterizer stage. There is no standard set of post-processing techniques, but a few possible ones are the simulation of the camera effect, such as the depth of field, gamma correction, addition of noise, and retouching.

To exploit these differences between PIM and PRCG, we propose the two-scale image description framework, inspired

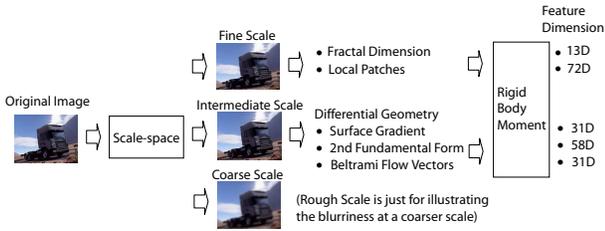


Figure 2: The geometry-based image description framework.

by Mandelbrot [8], in Figure 2. At the finest scale, the image intensity function is related to the fine-grained details of a 3D object’s surface properties. The finest-scale geometry can be characterized by the local fractal dimension (Section 6) and also by the “non-parametric” local patches (Section 4). At an intermediate scale, when the fine-grained details give way to a smoother and differentiable structure, the geometry can be best described in the language of differential geometry, where we compute the surface gradient (Subsection 7.1), the second fundamental form (Subsection 7.2) and the Beltrami flow vectors (Subsection 7.3).

The transition of an image to an intermediate scale is done in the linear Gaussian scale-space (Section 5), which is infinitely differentiable (in order for the “differential” geometry to be well defined).

4. LOCAL PATCH STATISTICS

Natural image statistics [15] represents the statistical regularities inherent in natural images (defined as images commonly encountered by human). Natural image statistics can be applied as an image prior for applications such as image compression, image recognition and image restoration. The important image statistics are the power law of the natural-image power spectrum, the wavelet high-kurtotic marginal distribution, and the higher-order cross-subband correlation of the wavelets coefficients. The wavelet features proposed in [7] are derived from these wavelet-based natural image statistics.

In addition to the transform-domain image statistics, an image-space natural image statistic is proposed [6]: The authors studied the high-dimensional distribution of 3×3 high-contrast local patches which mainly correspond to the edge structures. They found that the distribution is highly structured and concentrates on a 2D manifold in an 8D Euclidean space. By using this method, they managed to uncover the statistical difference between the optical (camera) images and the range (laser scan) images. Just like the PIM and the PRCG, these two groups of images correspond to two distinct physical image generation processes. There is further evidence that local patches can actually capture image styles. Painting, line drawing, computer graphics, photographs and even images of different resolutions can be considered as having different styles. The local patch model has been successfully applied to demonstrate image style translation [12], super-resolution [3], and other applications. This motivates us to employ local patch statistics.

Now, we describe the procedure for extracting the local patch statistic features. We extract (see Figure 3(a) & (b)) the grayscale patch and the joint-spatial-color patch independently at the edge points in two types of image regions: the high contrast region, and the low but non-zero contrast region. The two regions are obtained by thresholding the

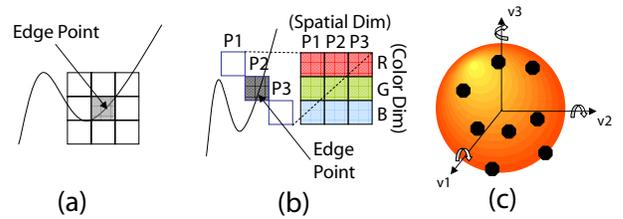


Figure 3: The grayscale patch (a) and the joint-spatial-color patch (b) are sampled at the edge points in an image. (c) Point masses on a sphere.

D -norm defined below. Note that the joint-spatial-color patch is approximately oriented to the image gradient direction which is the direction of maximum intensity variation. Each sampled patch, represented as a vector $\mathbf{x} = [x_1, x_2, \dots, x_9]$, is mean-subtracted and contrast-normalized as in Equation (1):

$$\mathbf{y} = \frac{\mathbf{x} - \bar{\mathbf{x}}}{\|\mathbf{x} - \bar{\mathbf{x}}\|_D} \quad (1)$$

where $\bar{\mathbf{x}} = \frac{1}{9} \sum_{i=1}^9 x_i$ and $\|\cdot\|_D$ is called D -norm. D -norm is defined as $\|\mathbf{x}\|_D = \sqrt{\sum_{i \sim j} (x_i - x_j)^2}$ with x_i and x_j representing the patch elements and $i \sim j$ denoting the 4-connected neighbors relationship of two pixels in a patch. The D -norm can also be expressed as the square root of a quadratic form $\|\mathbf{x}\|_D = \sqrt{\mathbf{x}^T D \mathbf{x}}$ where D is symmetric semi-positive definite matrix [6].

As the patch \mathbf{x} is contrast-normalized by the D -norm, the normalized patch is constrained by a quadratic relationship, $\mathbf{y}^T D \mathbf{y} = 1$. At this juncture, the data points are living on the surface of an ellipsoid in 9D Euclidean space. To facilitate the handling of the data points in a high-dimensional space, we change the elliptic constraint into a spherical constraint by a linear transformation $\mathbf{v} = M \mathbf{y}$. M is a 8×9 matrix and \mathbf{v} is constrained by $\mathbf{v}^T \mathbf{v} = \|\mathbf{v}\|^2 = 1$; hence, \mathbf{v} is located on 7-sphere in a 8D Euclidean space, as illustrated in Figure 3(c) in a 3D example. In this process, \mathbf{v} is reduced from 9D to 8D by taking advantage of the fact that \mathbf{y} is zero-mean. For the each of the four patch types (grayscale/low-contrast, grayscale/high-contrast, joint-spatial-color/low-contrast, and joint-spatial-color/high-contrast), we extract 3000 patches and separately construct a distribution on a 7-sphere in the 8D Euclidean space.

5. LINEAR GAUSSIAN SCALE-SPACE

In this section, we will give a brief introduction of the linear Gaussian scale-space in which we compute the local fractal dimension and the differential geometry quantities. The linear Gaussian scale-space $L : \Omega \subseteq \mathbb{R}^2 \times \mathbb{R}_+ \mapsto \mathbb{R}$ of a 2D image $I : \Omega \subseteq \mathbb{R}^2 \mapsto \mathbb{R}$ is given by:

$$L(x, y; t) = \iint_{\Omega} I(\xi, \eta) \phi(x - \xi, y - \eta; t) d\xi d\eta = \phi(x, y; t) * I(x, y)$$

where $L(x, y; 0) = I(x, y)$, t is a non-negative real number called the scale parameter, $*$ is the convolution operator and $\phi : \mathbb{R}^2 \mapsto \mathbb{R}$ is the Gaussian kernel function:

$$\phi(x, y; t) = \frac{1}{2\pi t} e^{-\frac{x^2 + y^2}{2t}} \quad (2)$$

Even though an image, $I(x, y)$ may not be differentiable initially, the corresponding linear scale-space, $L(x, y; t)$, $t >$

0 is infinitely differentiable with respect to (x, y) as long as $I(x, y)$ is bounded. As differential geometry quantities are the composition of derivative terms, the differentiability property ensures that the computed differential geometry quantities are well-defined. The partial derivative of a scale-space can be obtained by convolving the original image, $I(x, y)$, with the partial derivatives of the Gaussian kernel function $\phi(x, y; t)$:

$$\begin{aligned} L_{x^n y^m}(x, y; t) &= \partial_{x^n} \partial_{y^m} (\phi(x, y; t) * I(x, y)) & (3) \\ &= (\partial_{x^n} \partial_{y^m} \phi(x, y; t)) * I(x, y) & (4) \end{aligned}$$

where $\partial_{x^n} \partial_{y^m}$ is a shorthand for $\frac{\partial_{n+m}}{\partial_{x^n} \partial_{y^m}}$.

6. FRACTAL GEOMETRY

The **Object Model Difference** mentioned in Section 3 implies that the 3D computer graphic model's surface properties deviate from the real-world object's surface properties, which are associated with the physical formation process such as erosion. This deviation will directly result in a deviation of the local fractal dimension measured from the image intensity function, under certain assumptions [11]. Based on this, we conjecture that the deviation of the surface property would result in a different distribution for the local fractal dimension of PRCG.

In this section, we briefly introduce fractal geometry and the techniques for computing fractal dimension. A fractal is defined as a set of mathematical objects with a fractal dimension (technically known as the Hausdorff Besicovitch dimension) strictly greater than the topological dimension of the object but not greater than the dimension of the Euclidean space where the object lives. For example, a fractal coastline lives on a 2D surface, and, being a line, has a topological dimension of one, then its fractal dimension would be $1 < D \leq 2$.

For a real world object, to directly estimate the fractal dimension from the mathematical definition of the Hausdorff Besicovitch dimension is difficult. A fractal is self-similar across scales, so fractal dimension is often estimated as a factor of self-similarity. A commonly used random fractal model for images is called fractional Brownian motion (fBm) [8, 11]. With the fBm model, one method for estimating the fractal dimension is by measuring the self-similarity factor of a quadratic differential invariant in scale-space. We select this estimation technique in order to keep our approach consistent in the sense that both the fractal geometry and the differential geometry quantities are computed in the linear scale-space. We herein describe the estimation procedure. We first compute the $L1$ -norm of the second-order quadratic differential invariant:

$$\|I^{(2)}(t)\| = \sum_{\text{all } (x, y)} |I^{(2)}(x, y; t)| \quad \text{where } I^{(2)} = L_{xx}^2 + 2L_{xy}^2 + L_{yy}^2 \quad (5)$$

at multiple scales from $t = 2^2$ to $t = 2^8$ with an exponential increment. Then, we perform a least square linear regression on the $\log \|I^{(2)}(t)\| - \log t$ plot and measure the slope of the line. With the estimated slope, the fractal dimension is obtained by $D = \frac{1}{2} - \text{slope}$. Figure 4 shows two examples of fractal dimension estimation using the $\log \|I^{(2)}(t)\| - \log t$ plot. Note that a higher fractal dimension for the tree image block indicates a perceptually rougher image function. For feature extraction, we compute the fractal dimension for each of the non-overlapping 64×64 -pixel local blocks, inde-

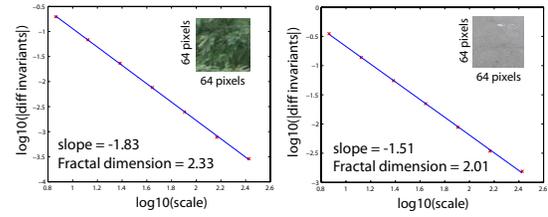


Figure 4: Log-log plot for estimating the fractal dimension of a 64×64 -pixel block from the tree (Left) and road (Right) region

pendently for the R, G and B color channels of an image. As a result, each local block produces a 3D fractal dimension vector across the color channels. For each image, we obtain a distribution of data points in the 3D space.

7. DIFFERENTIAL GEOMETRY

This section introduces three differential geometry quantities: the surface gradient, the second fundamental form and the Beltrami flow vector, computed at the scale $t = 1$ (empirically found to produce good classification performance). We also show the differences in these features for PIM and PRCG.

When computing the differential geometry quantities, we consider a single channel or a joint-RGB color intensity function as a graph defined below:

$$F_I : (x, y) \subset \mathbb{R}^2 \mapsto (x, y, I(x, y)) \subset \mathbb{R}^3 \quad (6)$$

$$F_{RGB} : (x, y) \subset \mathbb{R}^2 \mapsto (x, y, I_R(x, y), I_G(x, y), I_B(x, y)) \subset \mathbb{R}^5 \quad (7)$$

The graphs are submanifolds embedded in a Euclidean space, which naturally induces a Riemannian metric on each submanifold. A metric tensor, g , contains information for computing the distance between two points joined by a curve on a submanifold. Imagine that the measurement is carried out by an ant living on the surface without knowing the ambient space. We can measure distances on a submanifold, so we can also measure the area/volume and the angle between two tangent vectors at a point. Therefore, a metric tensor is an important element for describing the geometry of a manifold. The elements of the metric tensor for the graph of a single channel intensity function as in Equation (6) are computed as:

$$g = \begin{pmatrix} \langle F_{I_x}, F_{I_x} \rangle & \langle F_{I_x}, F_{I_y} \rangle \\ \langle F_{I_y}, F_{I_x} \rangle & \langle F_{I_y}, F_{I_y} \rangle \end{pmatrix} = \begin{pmatrix} 1 + I_x^2 & I_x I_y \\ I_x I_y & 1 + I_y^2 \end{pmatrix} \quad (8)$$

where $\langle \cdot, \cdot \rangle$ is the inner product and I_x, I_y are the derivatives of I .

From the invariant theory, the differential geometry quantities we compute are invariant to rotations and translations of the image intensity function $I(x, y)$ on the (x, y) plane. In addition, the computation of these quantities does not depend on the choice of coordinate systems.

7.1 Gradient on Surface

The **Acquisition Difference**, as mentioned in Section 3, can detect PRCG that have not undergone gamma correction as PIM generally do. One reason for missing gamma correction is that popular rendering platforms such as Silicon Graphics use hardware for gamma correction to enhance the contrast of the displayed images, therefore gamma correction on the images is not necessary. Additionally, gamma

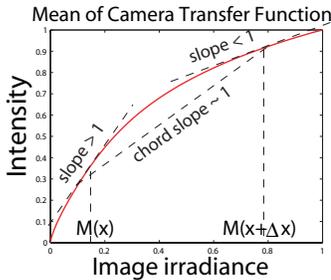


Figure 5: A typical concave camera transfer function. M is the image irradiance function

correction may be performed using the post-processing software such as Adobe Photoshop where the transformation is mainly subjected to the user’s taste. In this section, we will show that the surface gradient of the image intensity function can be used to distinguish PIM and PRCG.

The image intensity function captured by cameras, unless specifically set, has mostly been transformed by a transfer function, for the purpose of display gamma correction as well as for dynamic range compression. The transfer function transforms the image irradiance from the real-world scene to the intensity data $I(x, y)$, which is the output of the sensing process. The typical concave shape of a camera transfer function is shown in Figure 5.

One main characteristic of the camera transfer function in Figure 5 is that the irradiance of low values are stretched and those of high values are compressed. Let the image intensity function be $I(x, y) = f(M(x, y))$ where $f: \mathbb{R} \mapsto \mathbb{R}$ is the camera transfer function and $M: (x, y) \subset \mathbb{R}^2 \mapsto \mathbb{R}$ is the image irradiance function. By the chain rule, we have:

$$I_x = \frac{\partial I}{\partial x} = \frac{df}{dM} \frac{\partial M}{\partial x}, I_y = \frac{\partial I}{\partial y} = \frac{df}{dM} \frac{\partial M}{\partial y} \quad (9)$$

Note that the modulation factor, $\frac{df}{dM}$ is the derivative of the camera transfer function, which is larger (smaller) than 1 when M is small (large). Therefore, the Euclidean gradient $|\nabla I| = \sqrt{I_x^2 + I_y^2}$ of a transformed image is higher (lower) at the low (high) intensity than before the transformation. Namely, the modulation term $\frac{df}{dM}$ in Equation (9) reveals a key difference between PIM and PRCG, when PRCG images are not subjected to such modulation on their gradient values. If the PRCG intensity functions have not undergone such transformation, it can be distinguished from PIM by the gradient distribution.

The analysis above assumes that the image irradiance function M is continuous. There is a non-trivial issue involved in its implementation, when it comes to discrete-sampled images. Consider approximating the gradient at two neighboring pixels at locations x and $x + \Delta x$, Equation (9) becomes:

$$\frac{\Delta I_x}{\Delta x} = \frac{\Delta(f \circ M)_x}{\Delta M_x} \frac{\Delta M_x}{\Delta x} \quad (10)$$

where $\Delta I_x = I(x + \Delta x, y) - I(x, y)$, similarly for $\Delta(f \circ M)_x$ and ΔM_x . Note that, the modulation factor in this case becomes the slope of the chord on the camera transfer function connecting $M(x + \Delta x)$ to $M(x)$. One consequence is that the modulation will only be similar to the continuous case, when $M(x + \Delta x) - M(x)$ is small, because when $M(x + \Delta x) - M(x)$ is large, the slope of the chord is approaching 1 and modulation effect is weak. In other words, due to the discrete repre-

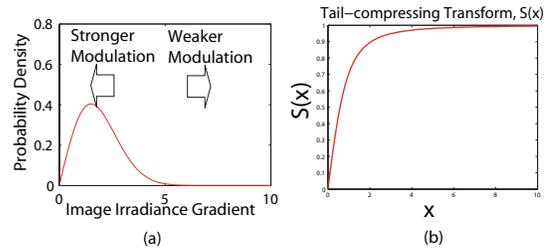


Figure 6: (a) The relative strength of the modulation effect for different magnitude of the image irradiance gradient (b) The tail-compressing function, $S(x; 1)$

sentation, the modulation effect shown in Equation (10) will arise only at points having low gradient values. This idea is illustrated in Figure 6(a). With this property, if we are to compare the distributions of the gradient of PIM with that of PRCG, we should place more weight on the low gradient region than on the high gradient region. Figure 6(a) shows a sample distribution of gradient values of an image, which typically contains long tail. To emphasize the low-gradient region, we employ a tail-compressing transform, S :

$$S(|\nabla I|; \alpha) = \sqrt{\frac{|\nabla I|^2}{\alpha^{-2} + |\nabla I|^2}} = |\text{grad}(\alpha I)| \quad (11)$$

$$\text{where } |\nabla I| = \sqrt{I_x^2 + I_y^2}$$

Figure 6(b) shows that the S transform is almost linear for the small values and it compresses the high values. The width of the linear range can be controlled by the constant, α . Interestingly, Equation (11) is the surface gradient of the scaled image intensity function $|\text{grad}(\alpha I)|$ computed from the Riemannian metric of the graph for the single channel intensity function.

Figure 7 shows the distribution of the mean of surface gradient $|\text{grad}(\alpha I)|$, $\alpha = 0.25$ (selected such that the linear range of the S transform covers the low and the intermediate of the Euclidean gradient), for three intensity ranges, i.e., $[0, \frac{1}{3})$, $[\frac{1}{3}, \frac{2}{3})$ and $[\frac{2}{3}, 1]$, of the blue color channel (the same holds for the red and green channels). These distributions are computed empirically from our actual dataset of PIM and PRCG. Notice that for the low intensity region, the mean of surface gradient for the PIM is higher than that of the PRCG and the opposite is observed for the high intensity region, while the distributions of the two are completely overlapped at the medium intensity range. This perfectly matches our prediction about the effect of the transfer function as described earlier in Equation (9).

The skewness statistic, which is highly overlapping as shown in Figure 7, is not used as classification features. For feature extraction, we compute the surface gradient and obtain $(|\text{grad} \alpha I_R|, |\text{grad} \alpha I_G|, |\text{grad} \alpha I_B|)$, $\alpha = 0.25$ for the three color channels at every pixel of an image. As gamma correction in PIM modulates the gradient differently at different intensity values, we combine the intensity values (I_R, I_G, I_B) with the above surface gradient vector to form a 6D feature vector at every pixel.

7.2 The Second Fundamental Form

Referring to the **Object Model Difference** as mentioned in Section 3, the accuracy of the 3D polygonal model of computer graphics is dependent on the granularity of the

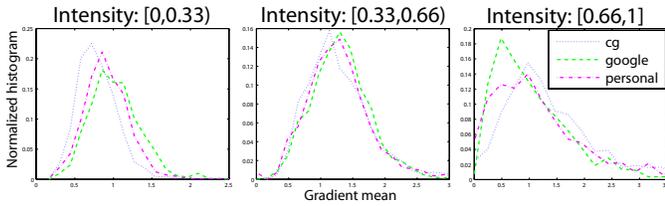


Figure 7: Distribution of the surface gradient for three different intensity ranges of the blue color channel. Each of the distribution is respectively computed from the entire image set

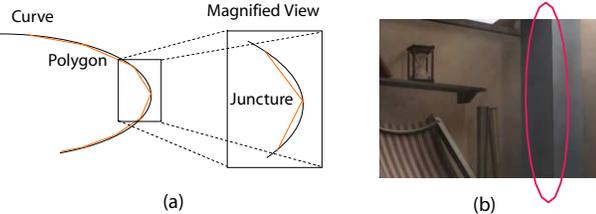


Figure 8: (a) Illustration of the polygon effect. (b) Unusually sharp structure in computer graphics, note the pillar marked by the red ellipse.

polygonal representation. A coarse-grained polygonal model can result in observable sharp structures, such as sharp edges and sharp corners, in the image intensity function of computer graphics. Figure 8(a) gives an intuitive illustration of this point; when a concave line is approximated by a polygonal line, the curvature at the junctures of the polygon segments is always greater than that of the concave line. Figure 8(b) shows an example of the sharp edge structure in the magnified view of a PRCG image. This structural difference between PIM and PRCG can be observed from the local quadratic geometry computed on the image intensity function. Quadratic geometry can be considered as a second-order approximation of the 3D shape at a point. The typical shapes of quadratic geometry are shown in Figure 9(a). In differential geometry, the quadratic geometry at each point (x, y) of an image intensity function (represented as a graph as in Equation (6)) is called the second fundamental form, defined as:

$$\Pi_p(v) = v^T A v \quad (12)$$

where v are the tangent vectors at the location (x, y) , A is a symmetric matrix given by:

$$A = \begin{pmatrix} \langle F_{I_{xx}}, n \rangle & \langle F_{I_{xy}}, n \rangle \\ \langle F_{I_{xy}}, n \rangle & \langle F_{I_{yy}}, n \rangle \end{pmatrix} = \frac{1}{\sqrt{1 + I_x^2 + I_y^2}} \begin{pmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{pmatrix} \quad (13)$$

where n is the unit normal vector at (x, y) . The matrix A determines the local quadratic geometry and can be characterized by its first and the second eigenvalues, γ^1 and γ^2 , where $\gamma^1 > \gamma^2$. In a 2D plot of the first and the second eigenvalues, every point represents a local quadratic geometry shape, as shown in Figure 9(b) (The meaning of the circles, ellipses and so on is given in Figure 9(a)). Note that, large eigenvalues correspond to the ‘sharp’ structures such as sharp ellipses or sharp circles. Given an image, the presence of the large eigenvalues can be measured by the skewness of the distribution of the eigenvalues in each image (Skewness may not be the best measure, but it serves our purpose); the larger the skewness is, the more large val-

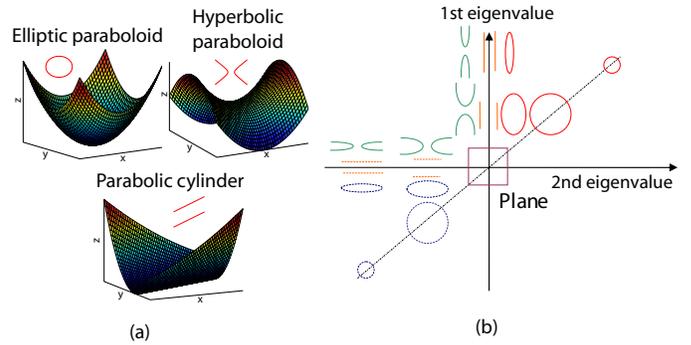


Figure 9: (a) The typical shapes of the quadratic geometry (b) The shapes of the quadratic geometry in a 2D eigenvalue plot. Colors are for visual aid

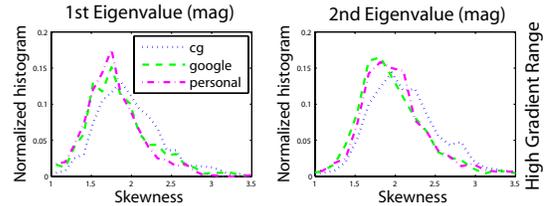


Figure 10: Distribution of the skewness of the 1st and 2nd eigenvalues of the second fundamental form for the blue color channel

ues are there. We compute the skewness of the eigenvalues separately for the images in our dataset and the distribution of the skewness is shown in Figure 10. We can see that the *CG* image set tends to have a larger skewness, while the shape of the distributions for the two photographic sets (*Google* and *Personal*) are quite consistent. This observation indicates that PRCG has more sharp structures than PIM.

The skewness statistic, which is highly overlapping as shown in Figure 10, is not used as classification features. For feature extraction, we compute the two eigenvalues of the quadratic form for the three color channels independently. As the geometry of the edge region and the non-edge regions are different in terms of the generative process, (e.g., low-gradient region is mainly due to smooth surface while high-gradient region is mainly due to texture, occlusion, change of the surface reflectance property or shadow), we therefore try to capture the correlation between gradient and the local quadratic geometry by constructing a 9D joint distribution of $(|\nabla I_R|, |\nabla I_G|, |\nabla I_B|, \gamma_R^1, \gamma_G^1, \gamma_B^1, \gamma_R^2, \gamma_G^2, \gamma_B^2)$, where γ^1 and γ^2 are the first and second eigenvalues and $|\nabla I| = \sqrt{I_x^2 + I_y^2}$ is the Euclidean gradient.

7.3 The Beltrami Flow

In Section 3, we discussed the **Light Transport Difference** in PIM and PRCG. The object reflectance property function in PRCG is often simplified such that its response to different color channels or spectral components are independent. This assumption is not true for the real-world objects in general, therefore this simplification may result in a deviation of the cross-color-channels relationship of PRCG with respect to that of PIM. Such joint-color-channel correlation has been exploited by techniques in image restoration to improve the subjective image quality. Therefore, we consider the Beltrami flow [14], which is an effective joint-

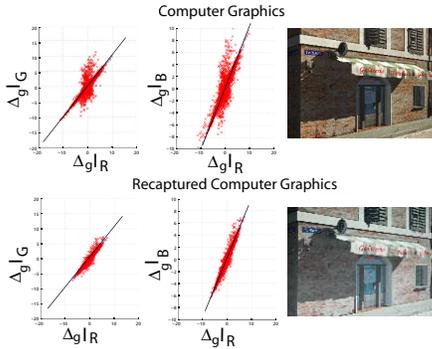


Figure 11: Comparing the joint distribution of the Beltrami flow components of a computer graphics image and that of its recaptured counterpart, the lines correspond to $y = x$

color-channel image restoration technique. Beltrami flow is based on the idea of minimizing the surface area, which has been employed for restoring degraded or noisy images where artifacts or noise are considered singularities on the image intensity function. Minimization of the surface area reduces the magnitude of the singularity.

For a joint-RGB image intensity function (as in Equation (7)), the R, G and B component of the Beltrami flow vector at every point (x, y) are given by,

$$\begin{aligned} \Delta_g I_i &= \frac{1}{|g|} \left(\partial_x \left(\sqrt{|g|} (g^{xx} \partial_x I_i + g^{xy} \partial_y I_i) \right) + \right. \\ &\quad \left. \frac{1}{|g|} \left(\partial_y \left(\sqrt{|g|} (g^{yx} \partial_x I_i + g^{yy} \partial_y I_i) \right) \right) \right) \end{aligned} \quad (14)$$

where $i = R, G, B$, while $g^{xx}, g^{xy}, g^{yx}, g^{yy}$ and $|g|$ are:

$$g^{-1} = \begin{pmatrix} g^{xx} & g^{xy} \\ g^{yx} & g^{yy} \end{pmatrix} = \begin{pmatrix} 1 + \sum_j (\partial_x I_j)^2 & \sum_j \partial_x I_j \partial_y I_j \\ \sum_j \partial_y I_j \partial_x I_j & 1 + \sum_j (\partial_y I_j)^2 \end{pmatrix}^{-1}$$

$$|g| = 1 + \sum_j |\nabla I_j|^2 + \frac{1}{2} \sum_{j,k} |\nabla I_j \times \nabla I_k|^2, j, k = R, G, B \quad (15)$$

Note that, the vector cross-product terms in Equation (15) capture the correlation of the gradients in the R, G and B color channels. We can visualize the 3D joint distribution of the Beltrami flow vectors from the 2D plots of $\Delta_g I_R$ - $\Delta_g I_G$ and $\Delta_g I_R$ - $\Delta_g I_B$. For the 2D plots of $\Delta_g I_R$ - $\Delta_g I_G$ and $\Delta_g I_R$ - $\Delta_g I_B$, we notice that the distribution of the PIM is more aligned to the $y = x$ line of the plots, while the PRCG tends to have misaligned points or outliers. This observation can be seen in Figure 11, showing the 2D plots of a PRCG together with those of its recaptured counterpart. We visually inspected 100 *CG* images and 100 *Google* images, and noticed that about 20% of the *CG* images have this misalignment as compared to less than 5% of the *Google* images. Such misalignment could be due to the spectral independence assumption for the surface reflectance function.

For feature extraction, we follow the same strategy as the second fundamental form and try to capture the correlation between the Euclidean gradient and the Beltrami flow vector. As such, we construct a 6D joint distribution $(|\nabla I_R|, |\nabla I_G|, |\nabla I_B|, \Delta_g I_R, \Delta_g I_G, \Delta_g I_B)$.

7.4 Normalization of Differential Geometry Features

When we compute the differential geometry quantities, we are essentially computing the derivatives of various orders in

scale-space at a fixed scale, $t = T$. We like to make sure that the geometry quantities are as invariant as possible when we resize an image or when the given image has been smoothed beforehand (introducing a scale uncertainty). To understand the effect of these operations on the scale-space computation, consider a simple example. For $f_1(x) = \cos(\omega x)$, the scale-space first derivative (the conclusion can be easily generalized to the higher-order derivatives) is given by:

$$L_{1x}(x; t) = -\omega e^{-\frac{\omega^2 t}{2}} \sin(\omega x) \quad (16)$$

Let's resize $f_1(x)$ by a factor s and obtain $f_s(x) = \cos(s\omega x)$; the corresponding scale-space first derivative would be:

$$L_{sx}(x; t) = -s\omega e^{-\frac{s^2 \omega^2 t}{2}} \sin(s\omega x) \quad (17)$$

As $f_1(x) = f_s(\frac{x}{s})$, we compare $L_{1x}(x; T)$ in Equation (16) with $L_{sx}(\frac{x}{s}; T) = -s\omega e^{-\frac{s^2 \omega^2 T}{2}} \sin(s\omega x)$ in order to understand the effect of image resizing and prior smoothing. The difference between the two terms is from the preceding factor s and the exponential factor decreasing with t . The exponential is content dependent because it is a function of ω . This general observation is applicable to the differential geometry quantities, being a composition of scale-space derivatives. If we compute a differential geometry quantity at every pixel of an image and obtain the distribution of this quantity for each image, the combined effect of the two factors will manifest itself in the standard deviation of the distribution. Therefore, we propose a simple divisive normalization scheme, that divides the differential geometry quantity of an image by its standard deviation. To prove such normalization is effective, we compute the Kullback-Leibler (KL) distance [2] between the distribution of the scale-space first derivative of an image and that of the half-sized version of the same image. Indeed, we find that the KL distance is reduced to about one-third after normalization.

8. DATASET COLLECTION

To ensure that our experimental dataset exemplifies the problem of image forgery detection, our dataset collection effort adheres to the following principles: (1) Images of diverse but natural-scene-only content: we exclude the PRCG of fantasy or abstract category and this restriction ensures a content matching between the PIM and the PRCG image sets, (2) Computer graphics of high photorealism: we subjectively filter the computer graphics from the web to ensure a high photorealism, (3) Images of reliable ground truth: we specifically collect a set of PIM from the personal collections which are known to be authentic. As a comparison, a very different approach in terms of the dataset is adopted in [7], where a very large number of online images, i.e., 32,000 PIM and 4,800 PRCG, are used and there is no mention of the selection criteria. Adhering to the above principles, we collected the below-described four image sets (see Figure 1). A detailed description of the dataset can be found in [10].

1. 800 PRCG (*CG*): These images are categorized by content into architecture, game, nature, object and life, see Figure 12(a). The PRCG are mainly collected from various 3D artists (more than 100) and about 40 3D-graphics websites, such as www.softimage.com, www.3ddart.org, www.3d-ring.com and so on. The rendering software used are such as 3ds MAX, softimage-xsi, Maya, Terragen and so on. The geometry modelling tools used include AutoCAD, Rhinoceros, softimage-3D and so on. High-end rendering techniques used



Figure 12: (a) Subcategories within *CG* and (b) Subcategories within *personal* image set, the number is the image count. (c) Examples of the test image sets in Table 1

- include global illumination with ray tracing or radiosity, simulation of the camera depth-of-field effect, soft-shadow, caustics effect and so on.
- 800 PIM images (*personal*): 400 of them are from the personal collection of Philip Greenspun, they are mainly travel images with content such as indoor, outdoor, people, objects, building and so on. The other 400 are acquired by the authors using the professional single-lens-reflex (SLR) Canon 10D and Nikon D70. It has content diversity in terms of indoor or outdoor scenes, natural or artificial objects, and lighting conditions of day time, dusk or night time. See Figure 12(b).
 - 800 PIM from Google Image Search (*Google*): These images are the search results based on keywords that matches the computer graphics categories. The keywords are such as architecture, people, scenery, indoor, forest, statue and so on.
 - 800 photographed PRCG (*recaptured CG*): These are the photograph of the screen display of the mentioned 800 computer graphics. Computer graphics are displayed on a 17-inch (gamma linearized) LCD monitor screen with a display resolution of 1280×1024 and photographed by a Canon G3 digital camera. The acquisition is conducted in a dark room in order to reduce the reflections from the ambient scene.

The rationale of collecting two different sets of PIM is the following: *Google* has a diverse image content and involves more types of cameras, photographer styles and lighting conditions but the ground truth may not be reliable, whereas *personal* is reliable but it is limited in camera and photographer style factors. On the other hand, based on the two-level definitions of image authenticity introduced in Section 2, we should be able to restore the imaging-process authenticity of the PRCG by recapturing them using a camera. Therefore, we produce the *recaptured CG* image set for evaluating how much the scene authenticity can be captured by the features.

Different image sets have different average resolution. To prevent the classifier from learning the resulting content-scale difference, we resize the *personal* and *recaptured CG* sets, such that the mean of the averaged dimension, $\frac{1}{2}$ (height + width) of the image sets matches that of the *Google* and the *CG* sets, at about 650 pixels.

9. DESCRIPTION OF JOINT DISTRIBUTION BY RIGID BODY MOMENTS

So far, we have described several distributions of features such as local patches (8D) and the differential geometry quantities computed for every pixel in an image. Here we

propose to extract the statistics of these distributions in order to reduce the dimensionality and develop a classification model for distinguishing PIM and PRCG. There are many ways to describe this high-dimensional distribution. One way, as directly follows from [6], is to uniformly quantize the surface of the sphere into 17520 bins (or other suitable bin count) to form a 1D histogram. This method needs a large number of image patches in order to have a stable histogram if the distribution is relatively spread out. In the other extreme, a non-parametric density estimation method, such as the Parzen kernel based method, can be used but the computation cost would be intensive when it comes to computing the distance between two estimated density functions. Besides that, Gaussian mixture model (GMM) clustering can also be used, but the standard GMM algorithm does not take advantage of the (non-Euclidean) spherical data space.

Due to the above considerations and the concern about the computational cost, we develop a framework based on the rigid body moments, which is capable in handling a high-dimensional distribution and is especially suitable for a spherical distribution. Let's first describe the process of computing the rotational rigid-body moments for the local patch distribution in the form of a discrete set of point masses on a 7-sphere [9]. For a distribution of N discrete masses, m_i , $i = 1, \dots, N$, respectively located at $\mathbf{v}_i = [v_{i1} \dots v_{iM}]$, the element of the inertia matrix is given by (18),

$$I_{jk} = \sum_i^N m_i (\|\mathbf{v}_i\|^2 \delta_{jk} - v_{ij} v_{ik}) \quad j, k = 1, \dots, M \quad (18)$$

where the Kronecker delta δ_{ij} is defined as 0 when $i \neq j$, 1 when $i = j$. In a three-dimensional Euclidean space of x - y - z Cartesian coordinates, the inertia matrix would be:

$$I = \sum_i^N m_i \begin{pmatrix} y_i^2 + x_i^2 & -x_i y_i & -x_i z_i \\ -x_i y_i & z_i^2 + x_i^2 & -y_i z_i \\ -x_i z_i & -y_i z_i & x_i^2 + y_i^2 \end{pmatrix} \quad (19)$$

Notice that the inertia matrix is symmetric. The diagonal and the off-diagonal components are respectively called the moments of inertia and the products of inertia. For an n -dimensional space, the inertia matrix has n moments of inertia and $\frac{1}{2}n(n-1)$ unique products of inertia.

For the distribution on the 7-sphere, the mass of the data points is $\frac{1}{N}$. We extract only the moment of inertia, as the number of the unique products of inertia is large. From another perspective, we can consider the elements of the inertia matrix as the second-order statistics of the distribution, and therefore it does not capture the complete information of distribution. Besides the moments of inertia, we also compute the center of mass (a form of the first-order statistics)

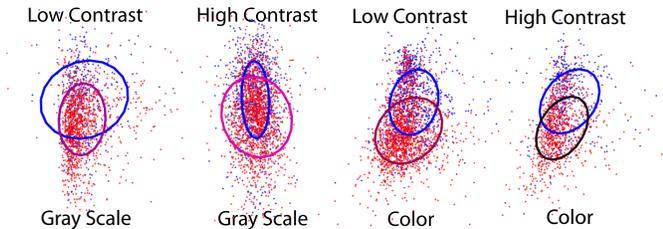


Figure 13: 2D projection of the local patch feature distribution for the (*Google+personal* image sets) (red) and the *CG* image set (blue)

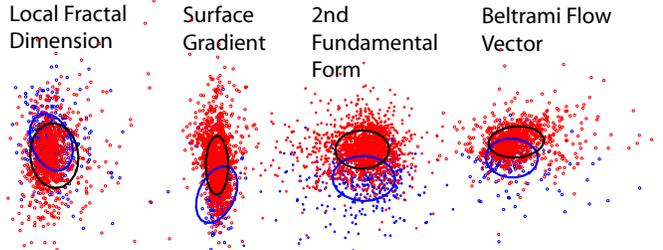


Figure 14: 2D projection of the fractal, the surface gradient, the 2nd fundamental form and the Beltrami flow vector features (from left to right) for the (*Google+personal* image sets) (red) and the *CG* image set (blue)

as well as the mean and the variance of the distance of the data points from the center of mass.

However, for the distribution of the fractal dimension, the surface gradient, the second fundamental form and the Beltrami flow vector, the data points are not restricted to a unit sphere. In this case, the inertia quantities can be affected by two factors: the distribution of points in the spherical and the radial direction. We decouple and model the two effects separately: we model the distribution of the unit-length data vectors using the center of mass as well as the moments and products of inertia, and model the magnitude of the data vectors using the first four moment of the distribution, i.e., mean, variance, skewness and kurtosis.

Figure 13 shows the feature distribution of the four local patch types, after having been linearly projected to a 2D space by Fisher discriminant analysis. The ellipses in the figure depict the mean and the covariance of a single-class feature distribution. We observe that the joint-spatial-color patch provides a better discrimination between the PIM and PRCG. Figure 14 shows the same 2D linear projection of the fractal, the surface gradient, the 2nd fundamental form and the Beltrami flow vector feature distribution. Notice that fractal dimension feature is the weakest discriminant for PIM and PRCG and differential geometry features are strong discriminant features.

10. EXPERIMENTS AND DISCUSSIONS

We evaluate the capability of our geometry-based features (henceforth the geometry feature) by classification experiments on our image sets. We compare the 192D geometry feature against the 216D wavelet feature [7] and the 108D feature obtained from modelling the characteristics of the general (i.e., including both photorealistic and non-photorealistic) computer graphics [5] (henceforth the cartoon feature). For a fair comparison, we compute the wavelet feature on the entire image (for a better perfor-

mance), rather than just on the central 256×256 -pixel region of an image, as described in [7]. The cartoon feature consists of the average color saturation, the ratio of image pixels with brightness greater than a threshold, the Hue-Saturation-Value (HSV) color histogram, the edge orientation and strength histogram, the compression ratio and the pattern spectrum.

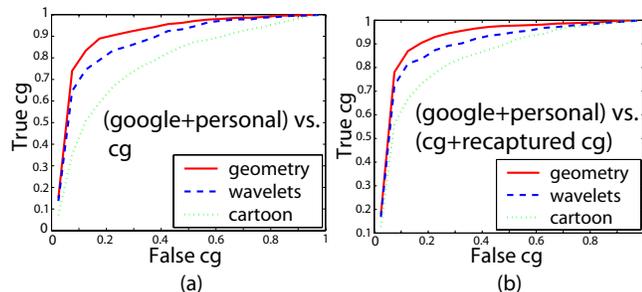
The classification experiment is based on the Support Vector Machine (SVM) classifier of the LIBSVM [4] implementation. We use the Radial Basis Function (RBF) kernel for the SVM and model selection (for the regularization and the kernel parameters) is done by a grid search [4] in the joint parameter space. The classification performance we report hereon is based on a five-fold cross-validation procedure. We train a classifier of the PIM (*Google+personal*) versus the PRCG (*CG*), based on the geometry, wavelet and cartoon features respectively. The receiver operating characteristics (ROC) curve of the classifiers are shown in Figure 15(a). The results show that the geometry features outperform the wavelet features while the conventional cartoon features do the poorest. The overall classification accuracy is 83.5% for the geometry feature, 80.3% for the wavelet feature and 71.0% for the cartoon feature (These numbers are different with 99% statistical significance).

To understand the strengths and the weaknesses of each approach on different images, we further test the classifier with images of the interesting and subjectively confusing categories. Results are shown in Table 1. Example images of the test sets are shown in Figure 12(c). The test accuracy reported is based on the classifier trained with the entire test category held out (i.e., no image of the test category is in the training set). We specifically conduct this experiment in order to study whether a good classifier can be learnt from images of different content categories. Notice that the geometry feature classifier outperforms that of the wavelet feature in three out of the five categories. The poor performance (for both wavelet and geometry features) on test set C indicates that the nature-scene PRCG have unique characteristics which cannot be learnt from the non-nature-scene ones. This could be due to the fact that nature-scene PRCG are mainly generated by the specialized rendering software such as *Terragen* and the nature-scene content often occupy the entire image. In contrast, PRCG with living objects (test set D) have a background which bears the characteristics which can be learnt from other PRCG. The results for the PIM of artificial objects (e.g., wax figures and decorative fruit) of test set B indicates that the artificiality of the real-world objects does not affect the classifiers. For test set E, the camera DoF effect is a global effect and our classifiers are based on local features, therefore simulating the DoF effect on PRCG does not prevent correct classifications.

In Table 1, almost all of the *recaptured CG* (test set A) are classified as PIM, for both sets of feature. This indicates that the features could only capture the imaging-process authenticity but not the scene authenticity. If we consider recapturing computer graphics as a form of attack to our computer graphic detector, it would be very effective. However, we can form a counter-attack measure by incorporating the *recaptured CG* into the computer graphics category during the classifier learning process. By so doing, the resulting classifiers have a ROC curve as shown in Figure 15(b). Note that the classifier is trained by having a pair of the computer graphics and its recaptured counterpart either entirely in

Table 1: Classifier Test Accuracy

Set	Test images (count)	Wavelets	Geometry
A	recaptured CG (800)	97.2%	96.6%
B	photos of artificial objects (142)	94.0%	96.2%
C	CG of nature scenes (181)	57.5%	49.2%
D	CG of living objects (50)	64.0%	74.0%
E	CG with DOF simulation (21)	85.7%	90.5%

**Figure 15: Receiver operating characteristic (ROC) curve for two classification experiments**

the training set or the test set, it is to prevent the classifier from overfitting to the content similarity of the pairs. Results in Figure 15(b) shows that this strategy renders the recapturing attack ineffective, and enables us to go beyond the imaging-process authenticity.

We analyze the computational complexity of the features by performing feature extraction on 100 images in Matlab 7.0. Their per-image feature-extraction time in seconds are 9.3s (wavelet), 5.2s (surface gradient), 8.7s (2nd fundamental form), 6.5s (Beltrami flow vector), 3.0s (local patch), 128.1s (fractal). Except for the fractal feature, the other features are quite computationally efficient.

11. CONCLUSIONS

We have proposed a new approach for PIM and PRCG classification in the context of image forgery detection. The new approach arises from asking a fundamental question of how we should describe images such that PIM and PRCG can be better distinguished. We adopt a geometry-based image description by means of the fractal geometry at the finest scale and the differential geometry at the intermediate scale. Additionally, we sample local patches of the image intensity function to form a patch distribution. The geometry-based approach enables us to uncover distinctive physical characteristics of the PIM and PRCG, such as the gamma correction of PIM and the sharp structures in PRCG, which has not been possible by using any of the previous techniques. We extract the geometry features using the method of rigid body moments, which can capture the characteristics of a high-dimensional joint distribution. The SVM classifier based on the geometry feature outperforms those in prior work. We also analyze the characteristics of recaptured computer graphics and demonstrate that we can make the recapturing attack ineffective. Furthermore, we identify a subset of PRCG with nature scenes, which remains challenging to classify and demands more focused research.

The experiment on recaptured computer graphics indicates the difficulty in capturing the characteristic of scene authenticity. To achieve this, we propose two methods: First, we can capture the characteristics of the forgery tech-

niques, just as we do by incorporating the recaptured computer graphics into the learning of the classifier. The second method, as a future work, is to consider a more fundamental modelling of the 3D scene authenticity using the computer graphics rendering and computer vision techniques. Final future work is to identify the local computer graphic regions within an augmented reality image.

12. ACKNOWLEDGEMENTS

We would like to thank NSF, Singapore A*STAR, Philip Greenspun, Eric Zavesky, Ravi Ramamoorthi, Dhruv Kumar Mahajan, Martin Pepeljugoski, and many others.

13. REFERENCES

- [1] T. Akenine-Moller, T. Moller, and E. Haines. *Real-Time Rendering*. A. K. Peters, Ltd., MA, 2002.
- [2] T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley-Interscience, New York, NY, USA, 1991.
- [3] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. Learning low-level vision. *Int'l J. Computer Vision*, 40(1):25–47, 2000.
- [4] C.-W. Hsu, C.-C. Chang, and C.-J. Lin. A practical guide to support vector classification. July 2003.
- [5] T. Ianeva, A. de Vries, and H. Rohrig. Detecting cartoons: A case study in automatic video-genre classification. In *IEEE Int'l Conf. Multimedia and Expo*, volume 1, pages 449–452, 2003.
- [6] A. B. Lee, K. S. Pedersen, and D. Mumford. The nonlinear statistics of high-contrast patches in natural images. *Int'l J. Computer Vision*, 54(1):83–103, 2003.
- [7] S. Lyu and H. Farid. How realistic is photorealistic? *IEEE Trans. Signal Processing*, 53(2):845–850, February 2005.
- [8] B. B. Mandelbrot. *The fractal geometry of nature*. San Francisco: W.H. Freeman, 1983.
- [9] J. L. Meriam and L. G. Kraige. *Engineering Mechanics Volume 2: Dynamics*. John Wiley and Sons, New York, 1986.
- [10] T.-T. Ng, S.-F. Chang, J. Hsu, and M. Pepeljugoski. Columbia photographic images and photorealistic computer graphics dataset. ADVENT Technical Report 205-2004-5, Columbia University, Feb 2005.
- [11] A. Pentland. On describing complex surface shapes. *Image and Vision Computing*, 3(4):153–162, November 1985.
- [12] R. Rosales, K. Achan, and B. Frey. Unsupervised image translation. In *IEEE Int'l Conf. Computer Vision*, pages 472–478, 2003.
- [13] J. R. Smith and S.-F. Chang. Visually searching the web for content. *IEEE Multimedia*, 4(3):12–20, 1997.
- [14] N. Sochen, R. Kimmel, and R. Malladi. A general framework for low level vision. *IEEE Trans. Image Processing*, 7(3):310–318, 1998.
- [15] A. Srivastava, A. B. Lee, E. P. Simoncelli, and S.-C. Zhu. On advances in statistical modeling of natural images. *J. Math. Imaging and Vision*, 18(1):17–33, 2003.
- [16] Y. Tsin, V. Ramesh, and T. Kanade. Statistical calibration of CCD imaging process. In *IEEE Int'l Conf. Computer Vision*, July 2001.