JPMTR 124 | 1902
DOI 10.14622/JPMTR-1902
UDC 159.937:004.92-023.8

Research paper
Received: 2019-02-03
Accepted: 2019-07-09

# The evaluation of rendered color shading using edge detection method

*Dejana Javoršek, Nika Bratuž and Helena Gabrijelčič Tomc*

Department of Textiles, Graphic Arts and Design,
Faculty of Natural Sciences and Engineering, University of Ljubljana,
Snežniška 5, 1000 Ljubljana, Slovenia

helena.gabrijelcic@ntf.uni-lj.si

## Abstract

Color reproduction in 3D computer-generated scenes is affected by scene setting, object and camera properties, illumination and rendering technique. Regardless of controllable conditions in a computer-generated environment, the achievement of colorimetric accuracy is complex, it depends the most on hue, saturation, and lightness of the virtual object's color and has to be considered from different perspectives. Visual perception of object color and shading is also influenced by object form, usually defined by contours. The aim of this research was to introduce edge detection method for evaluation of rendered color shading and to define how participants perceptually match color shaded renderings with predefined and proposed edge detection representations. The testing objects were color renderings of shaded objects whose colors were defined with tiff textures in the test chamber consisting of four lights, camera, and an object with shadows. In the experimental part, different edge detection methods and color channel models were used on selected colors of color renderings. It was confirmed that color, light intensity and background influence the frequency of detected edge pixels. Moreover, it was established that visual evaluation is also affected by color, background and light intensity. When the results of the frequency of detected edges and visual evaluation were compared, it was demonstrated that edge detection can be a suitable supportive method for evaluation of rendered color shading.

**Keywords:** color rendering, scene settings, photometric lights, standard lights, visual perception

## 1. Introduction

The validation of static and dynamic meshes is usually performed with quality metrics techniques. Bulbul, et al. (2011), Corsini, et al. (2013), and Yildiz and Capin (2017) presented the development and the review of these methodologies. Yildiz and Capin (2017) further classified techniques in geometry-distance-based metrics and perceptually based metrics, i.e. model-based and image-based perceptual metrics. In perceptually based metrics computational models the algorithms are developed with consideration of human visual system. Many of these researches include contrast sensitivity function, moreover, studies also present visible difference predictor, visual equivalence detector, visual masking etc., where human perceptual phenomena of color are not separately studied. On the contrary, color is studied as a distinctive image characteristic by Bolin and Meyer (1998), where vision-based rendering model was introduced including cone fundamentals, cortex filtering, calculation of local contrast and chromatic aberration, by defining of opponent's contrast space and by using contrast sensitivity function filtering, masking transducer, spatial pooling and distance summation and resulting in just noticeable difference map. Albin, et al. (2002) used LLAB color space for perceptual metrics performed on color images. LLAB color space is a derivation of CIELAB 1976 color space, that in this particular study enabled chromatic adaptation, computation in CIELAB color parameters as lightness, hue and chroma, and prediction of color difference between two selected images rendered under different conditions. In the study of Albin, et al. (2002) perception of color was evaluated with three types of metrics: first, pixel by pixel and local distance map, second, by the comparison of two images with a global value and third, by recursive subdivision and the creation of adaptive distance map. Color as a perception of visual characteristic has been mainly analyzed in terms of color perception, color constancy, color management and color appearance. Studies of color appearance, variations of color parameters analysis (chromatic changes, background) and the introduction of CIECAM02 color appearance model in computer graphic were presented only occasionally (Fedorovskaya, de Ridder and Blommaert, 1997; Trèmeau, Nicolas and Dinet, 2008; Bratuž, et al., 2014).

After the human brain receives visual data from the retina, visual properties of objects such as depth, shape, color and motion are probably extracted in the first stages as color-shape information and depth-motion information. After that, properties are separated and processed as a single parameter (Čadík, 2004; Livingstone and Hubel, 1988). Color and form are basic information for an object's identification. In the striate cortex, edge detection (that renders possible identification of the form) is performed by three types of cortical cells that have different receptive field characteristic. First, simple cells are called edge detector, due to their responding to the luminance edge in the proper orientation; second, complex cells are responding to movement; and third, hyper complex cells are also called end-stopped cells (Hubel and Wiesel, 1959). Biederman (1987) reviewed the basis of perceptual recognition, studied contour-based perceptual categories more closely and presented the 'recognition-by-components'. In Biederman's research edges are detectable due to their five properties: curvature, co-linearity, symmetry, parallelism, and co-termination. Processing stages that allow object recognition are edge extraction followed by two steps detection of non-accidental properties and parsing at region of concavity. After determining components and matching them to object representation, object identification occurs. Objects are segmented into primitive components, i.e. symmetrical volumes as blocks, cylinders, spheres, etc. In his research, Biederman also discussed that surface characteristics such as color and texture are processed simultaneously as volume characteristic (edges, depth), however, color and texture, in his opinion, do not have primary role for mental representation of the objects.

On the contrary, contours are supposed to be crucial for objects identification, even when they are degraded in the collinear and smooth parts. Edges and their reinterpretation have also played an important role in computer graphic visualization, where contours-based and outline-based non-photorealistic renderings are one of the most frequently used techniques (Gooch, et al., 1999; Geng, 2010). Due to their ability to simplify details of objects by means of color, gradates, topological characteristic, they are more efficient for some purposes to communicate the information in comparison with photo-realistic rendering techniques (Gooch and Willemsen, 2001; Strothotte and Schlectweg, 2002).

Edge detection is a spatial operation that detects presence and location of edges defined by changes in image intensity and is one of the most important operations in image analysis and enhancement (Koschan and Abidi, 2005). In general, edge detection methods are based on two principles; first is searching for local minima and maxima of the first derivative and second by zero crossing of the second derivative of an image (Oskoei and Hu, 2010). These operations are to some extent sensitive to noise and spatial filtering methods, and are normally incorporated into algorithm (Marr and Hildreth, 1980; Deriche, 1990; Shen and Castan, 1992; Leclerc and Zucker, 1987; Miché and Debrie, 1995). Methods presented by Sobel (2014), Prewitt (Oskoei and Hu, 2010) and Roberts (Davis, 1975) are based on the first principle and are classified as fast and efficient methods being variously sensitive to noise. Laplacian of Gauss (LoG) is an efficient method combining Gaussian filtering with second derivatives, however, mostly applied is Canny method that combines both aforementioned principles and is also time-efficient, but sensitive to false edges that present themselves in shading or blurring (Koschan and Abidi, 2005; Basu, 2002; Canny, 1986). There is no general edge detection method for all applications and evaluation is still based on subjective judgment. A general rule to define quality edge detection is thin detected edges and no or little noise (Koschan and Abidi, 2005; Oskoei and Hu, 2010). Studies of contour formation in visualizations and analysis of color reproductions after image processing in computer graphics have not yet been firmly implemented (Fu, et al., 2018; Zhou, et al., 2017; Fiorucci, et al., 2017). Besides, a review of the literature revealed that edge detection has not yet been considered as a potential supportive method for evaluation of rendered color shading and also analysis of the results of edge detection by visual perception of rendered color representations is still unknown.

The aim of the research was to introduce an edge detection method as parallel or supportive method for evaluation of rendered color shading. Moreover, the goal was also to analyze the results of edge detection by visual perception of rendered color representations. To study shading in renderings, different edge detection algorithms were implemented and results, i.e. frequency of edge pixels, was analyzed in dependence of color channel models, the object and background color, and light settings. Consequently, a study of the definition of a probable perceptional connection was performed between: first, visual perception of object color and detected contours, and second, results of frequency of edge pixel occurrence of different colored objects and scene settings. The results of the research should open new possible methodologies in the field of evaluation of color renderings and 3D object's shading.

## 2. Methods

### 2.1 Scene setting

Scene was set up in 3ds Max software in a test chamber and consisted of two types of light (four standard lights simulating lamps with intensity 0.4, 0.6, 0.8 and 1, or four photometric lights physically determined with standard illuminant D65 and intensity 3000 lm and 4000 lm), two cameras, and one shaded object (Figure 1).

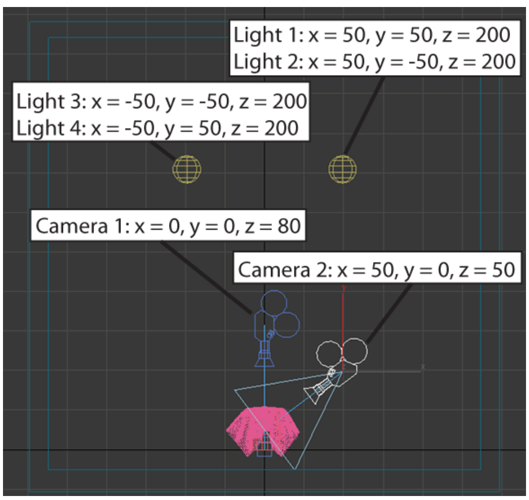


*Figure 1: The positions of the lights and cameras (in cm), shown in x,z plane of virtual x,y,z space*

Scene settings are presented in Table 1. Shadows were defined as raytraced. The scenes were rendered in mental ray render engine, i.e. production-quality rendering application based on ray-tracing technique. Other light and rendering settings were the default.

*Table 1: The optimal conditions chosen for testing with mental ray render engine (Bratuž, Javoršek and Gabrijelčič Tomc, 2015)*

| Type of light | Light intensity | Gamma | Background |
|---|---|---|---|
| **Standard** | 0.4 | 1.8, 2.0, 2.2 | *R*, *G*, *B* = 255 |
| (std.) | 0.6 | 1.8, 2.0, 2.2 | *R*, *G*, *B* = 200 |
| | 0.8 | 1.8, 2.0 | *R*, *G*, *B* = 100 |
| | 0.8 | 2.2 | *R*, *G*, *B* = 50 |
| | 1.0 | 1.8, 2.0, 2.2 | *R*, *G*, *B* = 0 |
| **Photometric** | 3000 lm | 1.8, 2.0, 2.2 | *R*, *G*, *B* = 150 |
| (pho.) | 4000 lm | 1.8, 2.0, 2.2 | *R*, *G*, *B* = 50 |

Preliminary research was carried out to determine optimal conditions for color reproduction that are presented in Table 1. In that research, colorimetric accuracy of color reproduction of test chart colors illuminated with two types of virtual (digital) light sources and rendered with two rendering engines was analyzed (Bratuž, Javoršek and Gabrijelčič Tomc, 2015). Optimal conditions were established in terms of gamma values, illumination and background and those optimal conditions were further used in the presented research for analysis of edge detection and its potentials for the evaluation rendered color shading.

Position of lights and camera was defined with *x*, *y* and *z* coordinates (in cm) of central point. In total, four lights and two cameras (top view and side view) were positioned on the scene (Figure 1). The focus of cameras was defined by their targets, i.e. the position of the camera's focus point.

### 2.2 Selection of colors

In our research 15 colors were chosen from color test chart and tiff texture format was mapped on 3D object. Reference *R*, *G*, *B* values of chosen colors are presented in Table 2.

*Table 2: Reference R, G, B values*

| Patch | *R* | *G* | *B* | Color |
|---|---|---|---|---|
| F17 | 31 | 17 | 38 | dark violet |
| A39 | 5 | 26 | 15 | dark green |
| A1 | 124 | 141 | 158 | grey |
| H6 | 235 | 239 | 251 | white |
| I18 | 205 | 2 | 3 | saturated red |
| I10 | 97 | 191 | 15 | saturated green |
| A2 | 8 | 16 | 139 | saturated blue |
| M39 | 157 | 129 | 206 | violet |
| C1 | 205 | 5 | 247 | magenta |
| N4 | 231 | 228 | 2 | saturated yellow |
| A14 | 155 | 43 | 151 | saturated violet |
| H23 | 217 | 165 | 224 | light pink |
| F30 | 130 | 229 | 206 | light cyan |
| B23 | 233 | 112 | 205 | pink |
| B27 | 130 | 32 | 33 | dark red / brown |

### 2.3 Edge detection

In order to evaluate rendered color shading, edge detection was performed in Matlab with four methods and consequently, the frequency of occurrence of edge pixels in relationship to non-edge pixels on an image was evaluated by calculating the percentage of detected edge pixels in comparison with all pixels on the image.

In general, edge function takes an intensity image $M = f(x,y)$, where $x$ and $y$ represents spatial coordinates and function $f$ denotes intensity function, as its input and returns a binary image $BW = g(x,y)$, where $x$ and $y$ represent spatial coordinates and function $g = [0,1]$. The *BW* is of the same size as *M*, with ones where the function finds edges and zeroes elsewhere. Several methods are provided in Matlab: Sobel, Prewit and Roberts methods find edges using the appropriate (Sobel, Prewit or Roberts, accordingly) approximation to the derivative and they return edges at those points where the gradient of *M* is maximum. The Laplacian of Gaussian (LoG) method finds edges by looking for zero crossings after filtering *M* with a Laplacian of Gaussian filter. Canny method finds edges by looking for local maxima of the gradient of *M*. Gradient is calculated using the derivative of a Gaussian filter. This method uses two thresholds to detect strong and weak edges and includes weak edges in the output only if they are connected to strong edges. This method, therefore, less likely than others provide false information due to noise, and more likely detects true weak edges.

In our research, four methods for edge detection were used: Canny, Sobel, LoG, and Prewitt, abbreviated as C, S, L, and P in figures, respectively. Default Matlab parameters were used for all methods, thus threshold was set automatically for all methods. Direction for Sobel and Prewitt is set to 'both', default sigma parameter is set to '2' for LoG and 'sqrt(2)' for Canny (MathWorks, 2018). As edge detection is performed on intensity channel of an image *M*, three methods were used to calculate *M* (channel color model RGB, I, Y).

First, RGB, where color image was separated into three intensity images R, G, and B, edge detection was performed on each channel separately and results for each channel were merged into binary image *BW* (marked as rgb later on); Y channel of YIQ color space used by NTSC color TV system (marked as y), and I channel of 'hue, saturation, intensity' (HSI) color space (marked as i). Second, frequency of occurrence of edge pixels in relationship to non-edge pixels on an image was calculated as percentage of edge pixels in relation to all pixels in an image. Third, average frequency per color was also calculated. Frequency of edge pixels was discussed in dependence of object and background color, light settings, and color channel model (RGB, I, Y). Detailed numerical and visual inspections were additionally carried out for Canny and LoG algorithms, for dark green A39 and white H6 colors and on one representative topological part (detail) of the object.

### 2.4 Visual perception analysis

Next, a visual evaluation was carried out to determine whether results of edge detection match the visual perception of the observers when they are shown and asked to compare representations of contours of objects in the images (results of edge detection) and color images of objects that represent rendered color shading (Figure 2). The participants were asked to perceptually match color shaded renderings that consists of objects form, contours, color and tone transitions (color gradients) with predefined and proposed edge detection representations. 59 young observers (that voluntarily participated in the research), 45 women and 14 men, aged from 19 to 20 years, with normal sight and no color blindness reported. The participants' task was to correlate proper edge representations with rendered draped object colors: dark green (A39), light cyan (F30), white H6, red I18, and yellow N4 at standard light intensities 0.6 and 1. In test A, observers had to choose the optimal edge detection of a color rendering, to determine preferred edge detection algorithm between Canny (Crgb and Ci) and LoG (Lrgb and Ly). This task was performed merely to determine observers' preference.

In the second part of the analysis of visual perception (test B), observers had to match reference rendering with: first, matching edge detection image; second, edge detection image of the same color rendered in the scene with different background and light intensity, and third, edge detection image of different color, which had a high percentage of area covered with edges. With this test, the analysis of the actual visual correlation between rendered color and edge detection image and an investigation whether the observers tend to select the representation with the highest percentage of edge

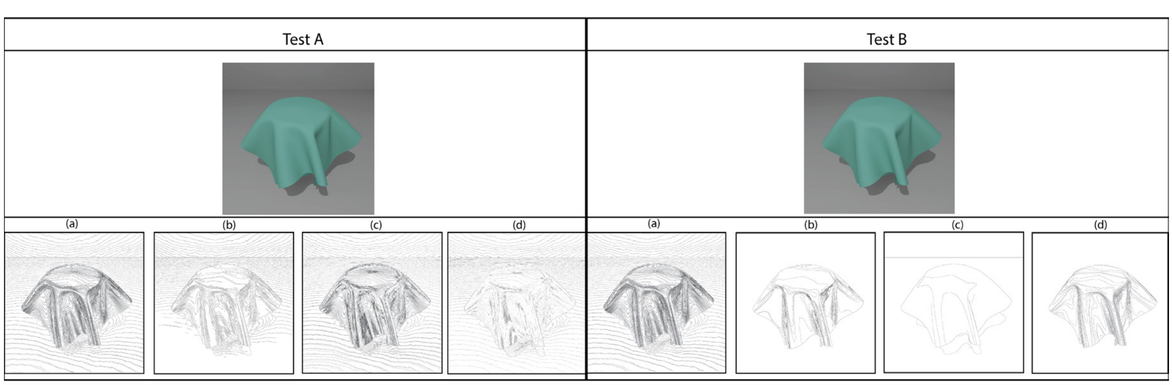


*Figure 2: Test A and Test B with the images shown to the observers during visual perception analysis*

detection regardless the color, background, and light intensity were carried out. For both tests, the visual analysis was performed in test room under controlled viewing conditions, where the images were presented on a 24'' monitor with the resolution of 1920 × 1200 pixels. Distance from the participants to the monitor was 60 cm. The participants were asked to observe the test images (presented in Figure 2, Test A and Test B) for 10 s (without interacting with them) and to choose the edge detection representation (naming the letter a, b, c, and d.). The question that was asked to the participant was: "Which of the four images showing edge detection (a, b, c, d) most accurately represents the rendering of the colored object?" The participants had time to rest between consecutive presentations of tests and when they were prepared, the next presentation was displayed.

## 3. Results and discussion

### 3.1 Results of edge detection

In Figures 3 to 8 results of edge detection frequency are shown for selected colors. In order, darker colors black F17, and dark green A39, brighter colors white H6, and light cyan F30, saturated red I18, and saturated yellow N4, were analyzed and frequency of edge pixels is shown (in %). To clarify, the names of applied edge detection functions and channel used are combined into name of method on the x axis.

For example, Crgb represents Canny edge detection algorithm applied on RGB channels. Abbreviations S, P, and L were used accordingly for other methods, meanwhile, rgb, i, and y stand for investigated channels.

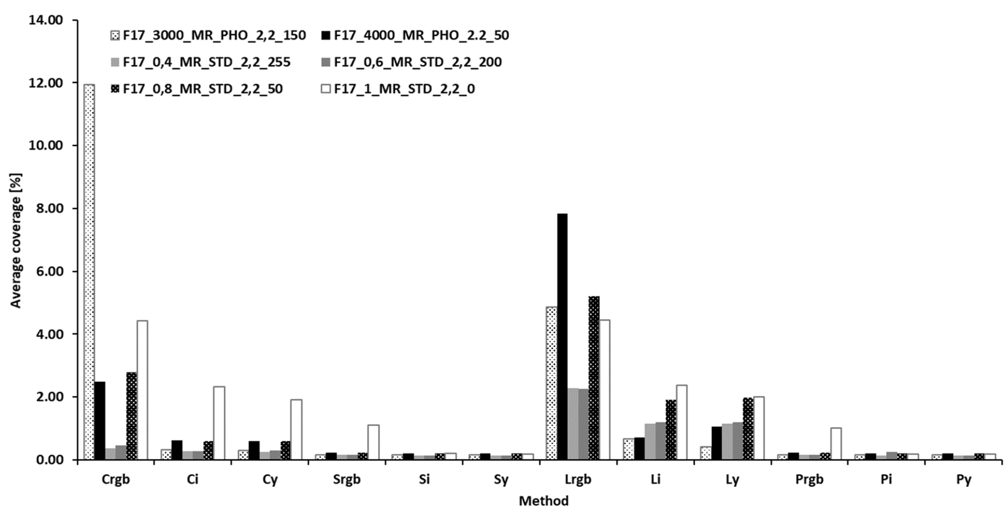


*Figure 3: Edge detection frequency for rendered color black F17*

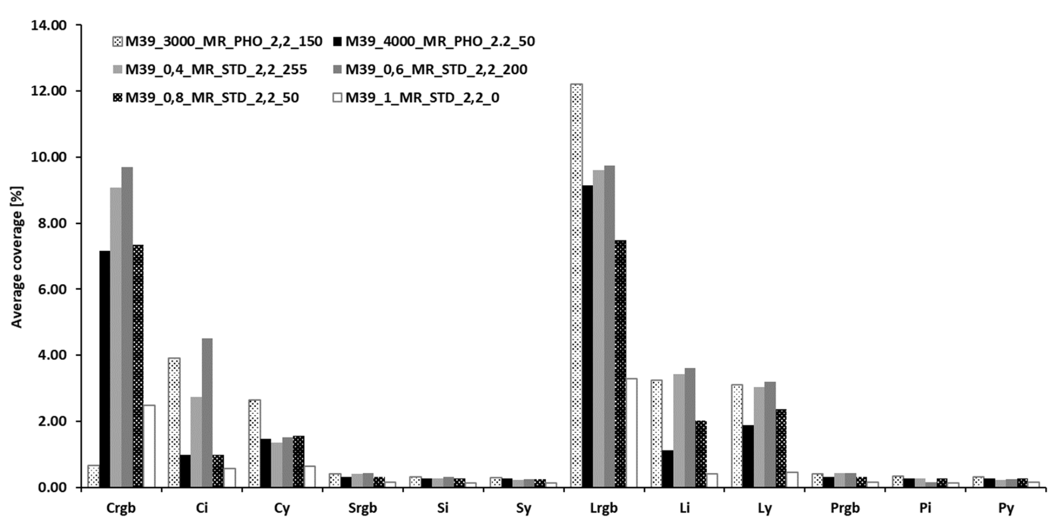


*Figure 4: Edge detection frequency for rendered color dark green A39*

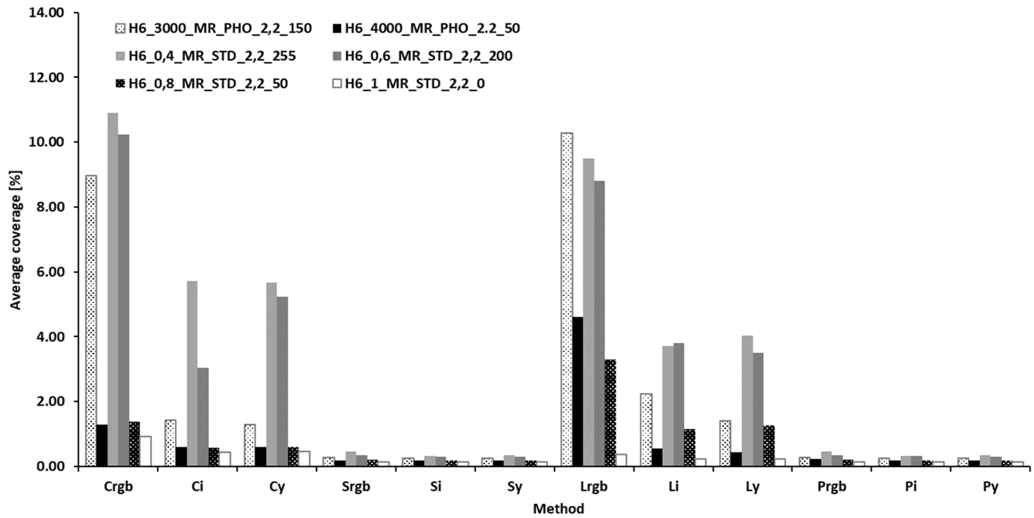


*Figure 5: Edge detection frequency for rendered color white H6*

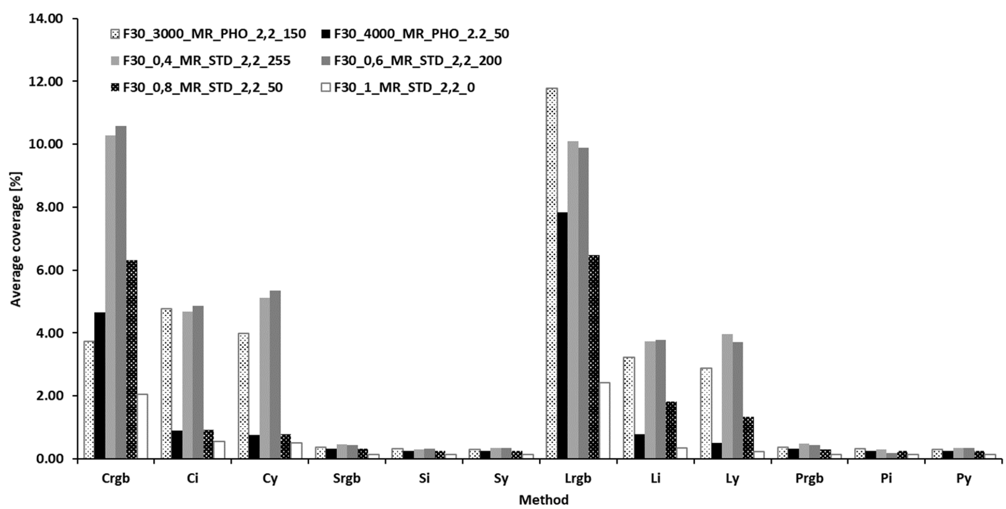


*Figure 6: Edge detection frequency for rendered color light cyan F30*

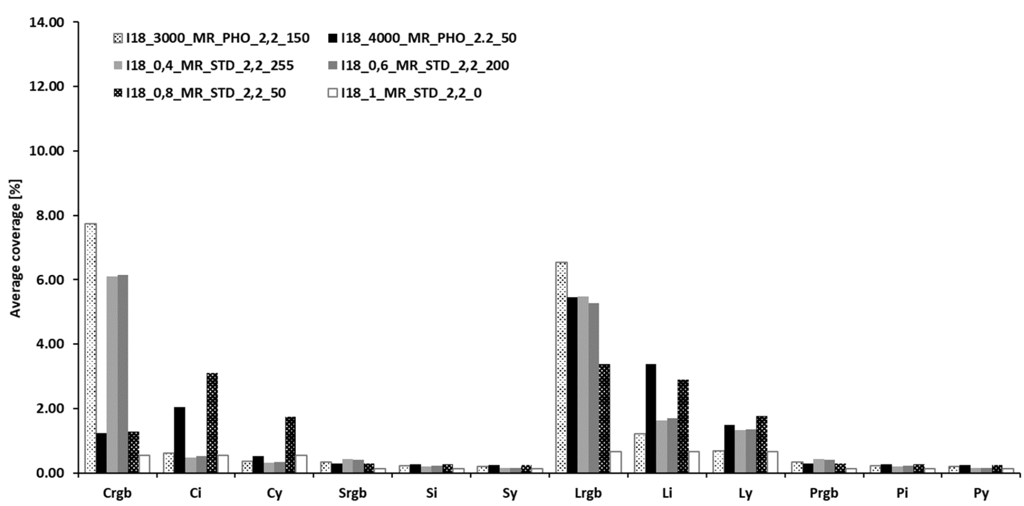


*Figure 7: Edge detection frequency for rendered color red I18*

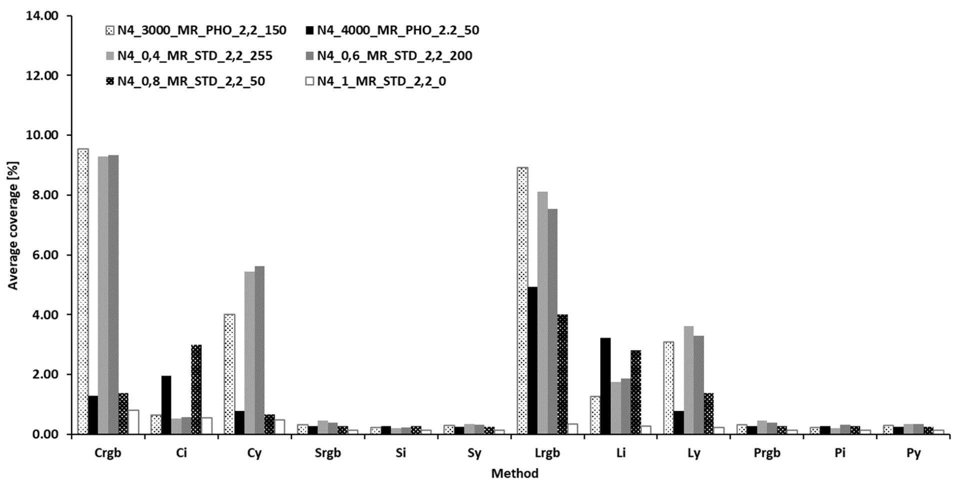


*Figure 8: Edge detection frequency for rendered color yellow N4*

3.1.1 The analysis of edge detection in dependence of color and illumination/background setting

The analysis of edge detection shows that frequency of edge pixels depends on hue, saturation, and lightness of colors. In general, numerical values of edge detection frequency are the highest for brighter colors (i.e. white H6, grey A1, light cyan F30, light pink H23) and the lowest for dark colors (i.e. black F17, dark green A39), as presented in Figures 3 to 8.

When considering the relation between color saturation and frequency of detected edges, it can be deduced that mainly less saturated and brighter colors (white H6, light cyan F30, pink B23, light pink H23) and colors as saturated green I10, violet M39, saturated red I18, magenta C1 and yellow N4 result in higher frequency of detected edges when the intensity of std. lights is 0.4 and 0.6 (background colors *R, G, B* = 255, and *R, G, B* = 200, accordingly) and the intensity of pho. light is 3000 lm (background color *R, G, B* = 150). On the contrary, dark saturated colors (grey A1, dark violet F17, violet A39, dark red/brown B27, saturated blue A2) had higher values of frequency for intensities 0.8 (background colors *R, G, B* = 50 and *R, G, B* = 100) for std. light and intensity 4000 lm for pho. light (background color *R, G, B* = 50). Those trends are present in all types of channel analysis (RGB, I and Y) and in all analyzed algorithms.

The deviations between results of lower (std. lights 0.4 and 0.6; pho. lights 3000 lm) and higher (std. lights 0.8, 1.0; pho. light 4000 lm) light intensities can be observed also when results of edge detection are evaluated on renderings with saturated colors and different hues. At lower light intensities saturated colors in the red and yellow region of RGB color space tend to have a higher frequency of detected edges than colors in blue, cyan and green region. Saturated colors magenta and especially violet have an ambiguous behavior, as, in dependence of lights intensity, they lean to results of both color groups: red–yellow and blue–green.

The results from Oskoei and Hu (2010) show that by use of Sobel and Prewitt algorithms, edge detectors that compute the first derivative captured only the most crucial contours to represent the renderings, regardless of the background/object color and light intensity. In our investigation, edge detection represents mainly the object's silhouette, i.e. the outer contours between the object and background and the shading could not be studied. Consequently, the further application of these algorithms was not useful, and the analysis continued with the Canny and LoG algorithms.

### 3.2 Visual and numerical analysis of edge detection of color samples dark green A39 and white H6

Detailed edge detection analysis was performed on two color samples: dark green A39 and white H6. As there was a distinct difference in-between the results of light intensities 0.4, 0.6 vs. 0.8, 1.0 for std. lights and 3000 lm vs. 4000 lm for pho. lights; both groups of samples and their results are discussed separately.

3.2.1 Light intensity of standard light 0.4 and 0.6, and photometric light 3000 lm

At light intensities, 0.4, 0.6 and 3000 lm, frequency of occurrence of edge pixels were lower for color dark green A39 (Figure 4) and higher for bright color white H6 (Figure 5). Differences in edge detection between two analyzed colors are lower in case of images generated with pho. lights and when edge detection is performed according to RGB channels (especially for color white H6). Frequency of detected edges is evidently higher in case it was performed on RGB channels in

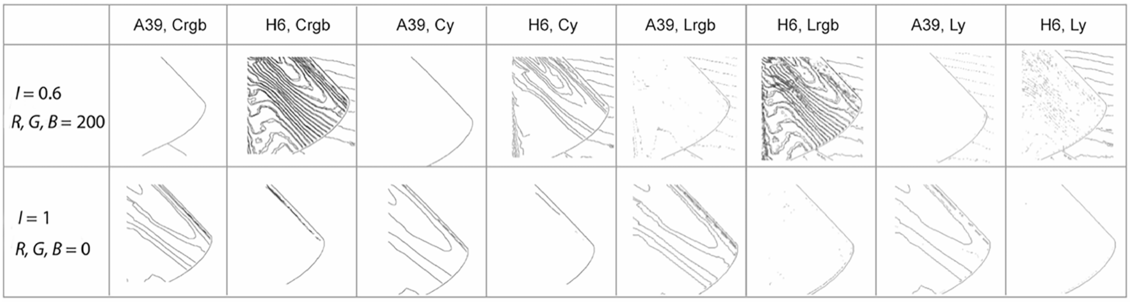


*Figure 9: Detected edges for colors violet A39 and white H6 by use of Canny and LoG algorithms on RGB and Y channels, on the scenes with the background R, G, B = 200 and standard light at intensity I = 0.6 and on the scenes with the background R, G, B = 0 and standard light at intensity I = 1.0*

comparison with the detection on channels I and Y. The comparison between Canny and LoG algorithm showed that the first has minor edge detection in case of color dark green A39, and stronger detection in case of bright color white H6, while the results of the LoG algorithm are just the opposite.

When the visual analysis was performed on processed images, it was found out that when performed on RGB, I and Y channels, the trend of edge detection is similar to both Canny and LoG algorithms. In Figure 9 a detail of an object is presented as edge detection with Canny and LoG algorithms for colors dark green A39 and white H6, according to RGB and Y channels and for the scenes with the background *R, G, B* = 200 and *R, G, B* = 0 illuminated with std. light at intensity 0.6 and 1.0.

#### 3.2.2 Light intensity of standard light 0.8 and 1.0, and photometric light 4000 lm

Analysis of edge detection on renderings at higher light intensities (std. 0.8, 1.0 and pho. 4000 lm) in combination with dark background colors (*R, G, B* = 50 and *R, G, B* = 0) showed opposite results as in the case of scenes with brighter backgrounds and lower light intensities (Figure 9).

Numerical results of edge detection at these light intensities are in general higher for color dark green A39 (Figure 4), and lower for the bright color white H6 (Figure 5). Deviations between the numerical and visual representation of edge detection for both colors are negligible for the analysis of I and Y channels, meanwhile, the analysis of the RGB channels represent the more obvious difference between the results.

Besides, the differences between edge detection of two analyzed colors are lower for the scenes with higher light intensities and darker background *R, G, B* = 50 and *R, G, B* = 0 than it was demonstrated for the scenes with brighter background colors *R, G, B* = 150, *R, G, B* = 200 and *R, G, B* = 255.

Furthermore, the LoG algorithm resulted in a higher frequency of detected edges than the Canny algorithm (especially for the analysis according to RGB channels).

### 3.3 Visual perception analysis

In the first part of the visual analysis, different edge detection algorithms (Canny with RGB and I channels, marked as Crgb, Ci; and LoG with RGB and Y channels, marked as Lrgb, Ly) were tested and a number of observers that have chosen defined edge detection algorithm as the optimal representation of a color rendering were defined. In Figure 10 the numbers of answers that represent the observers' choice of an optimal edge detection algorithm for the images with object colors dark green A39, light cyan F30, white H6, saturated red I18 and saturated yellow N4 at std. light intensities 0.6 and 1.0 are presented.

Preliminary results of edge detection analysis show that when using the Canny algorithm with RGB values, the maximum frequency of detected edge pixels was obtained. It was followed by LoG algorithm performed on RGB channels and Canny and LoG algorithm performed on the Y channel. Visual analysis revealed the difference between results of testing on renderings with light intensity 0.6 (background *R, G, B* = 200) and 1.0 (background *R, G, B* = 0). In case of light intensity 1.0, Canny algorithm prevailed as a preferred solution for colors light cyan F30, white H6, saturated red I18 and saturated yellow N4, in the range from 23 % to 68 % (observers often chose Canny algorithm performed on RGB channels for colors cyan F30 and white H6, and Canny algorithm performed on I channel for colors red I18 and yellow N4). The exception is color dark green A39, where the difference between the results of Canny and LoG algorithms is smaller. The Canny algorithm performed on I channel was chosen by 38 % of observers, followed by LoG algorithm performed on RGB by 30 % of observers. When the light intensity was 0.6, the Canny algorithm performed on RGB and I channels was mainly chosen for bright colors light cyan F30 and

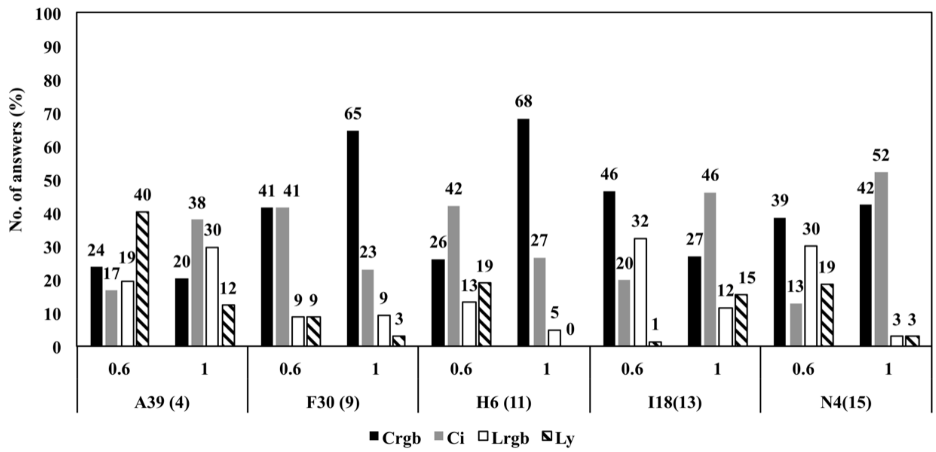


*Figure 10: Results of perceptions in dependence of algorithm*

white H6 (from 26 % to 41 %), meanwhile observers chose more equally between Canny (mainly Crgb) and LoG (Lrgb) algorithm for color saturated red I18 (Crgb = 46 %, Lrgb = 32 %) and color saturated yellow N4 (Crgb = 39 %, Lrgb = 30 %).

Again, the exception here was color dark green A39, were for the 40 % of observations the most appropriate was LoG algorithm performed on Y channel.

In addition, in the second part of analysis of visual perception users had to choose among images that represent proper edge detection image; first, edge detection image of the same color rendered in the scene with different background and light intensity, and second, edge detection image of different color which had a high portion of area covered with edges. The answers on both questions were considered as right, because we were interested in the perception of the colored object, regardless of the illumination and background.

In Figure 11 it can be observed that results again differ in dependence on tested colors. The percentage of wrong answers is high, from 29 % for the color light cyan F30 (light intensity 0.6) up to 73 % for the color saturated red I18 (light intensity 1.0).

Finding that observers have matched wrong edge detection image to color rendering is confirmed especially for the scenes with light intensity 1.0 (dark background $R$, $G$, $B$ = 0) and saturated colors, i.e. saturated red I18 and saturated yellow N4 where the percentage of wrong answers was 59 % and 73 %, respectively. It can be presumed that observers gave wrong answers here due to their choice of edge detection images with a high frequency of detected edges (strong detection of

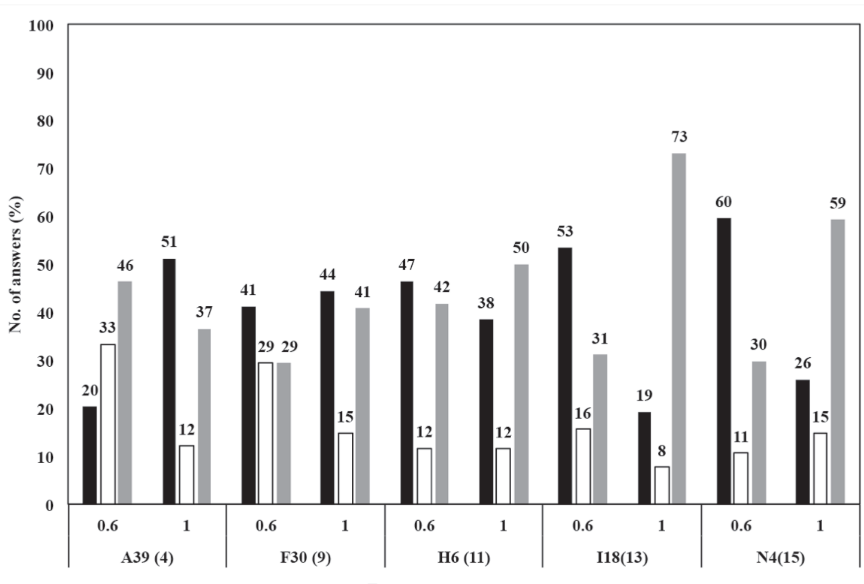


*Figure 11: Results of perception in dependence of background and light intensity*

shading), meanwhile, the algorithms usually detected a lower percentage of edges within these renderings. When all the right answers are summed up, results show that at light intensity 0.6, observers answered correctly for all colors with percentages from 53 % for the darkest color dark green A39 up to 70 % for light cyan F30 and 71 % for saturated yellow N4. By all means, these results do not confirm with full certainty there is a match between visual perception of contours (calculated with edge detection algorithms) and color objects, and that observations are strongly influenced by background and light setting. It can be deduced that when visual perception is considered, the use of edge detection processing could be plausible as an alternative method for evaluation of rendered color shading only for the rendered images at a combination of lower light intensities and bright background colors. Meanwhile, for the scenes rendered at higher light intensities and dark background, the application was not successful enough, due to observers' tendency for perceptual interpretation of dark shaded areas of rendered colors with a high frequency of edge pixel occurrence (regardless of the object's and background color and light intensity).

## 4. Conclusions

In the research, edge detection method and contour formation analysis was introduced on rendered color shading with an aim to define the relevance of the implementation of edge detection method in the color evaluation and its correlation with visual perception.

The analysis of the frequency of detected edge pixels on renderings shows that it depends on hue, saturation, and lightness of colors. The highest frequency was detected for brighter colors, whereas it was high for less saturated brighter colors and less saturated darker colors for lower lights' intensities and vice versa. Results of edge detection also depend on the hue and can be divided into two groups: red–yellow and blue–green.

Edge frequency obtained by Canny and LoG algorithm was adequate for further analysis and all analyzed color channels gave similar results. The highest frequency was achieved with the Canny algorithm performed on RGB channels and it was studied more closely. The visual assessment showed that there was no edge detection method favored by observers and that results depend on color and light intensity. Furthermore, the background also affects observers' judgment, which was proven in the second part of visual assessment.

Methods used in this study are a useful tool to assess color renderings and rendered color shading in computer reproductions of colors and provide new observations and a fresh point of view on the field of evaluation of contour visualizations and implementation of standardized image analysis in the workflow.

### References

Albin, S., Rougeron, G., Péroche, B. and Trémeau, A., 2002. Quality image metrics for synthetic images based on perceptual color differences. *IEEE Transaction on Image Processing*, 11(9), pp. 961–71. https://doi.org/10.1109/TIP.2002.802544.

Basu, M., 2002. Gaussian-based edge-detection methods – a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 32(3), pp. 252–260. https://doi.org/10.1109/TSMCC.2002.804448.

Biederman, I., 1987. Recognition-by-components: a theory of human image understanding. *Psychological Rev*iew, 94(2), pp. 115–147. http://dx.doi.org/10.1037/0033-295X.94.2.115.

Bolin, M.R. and Meyer, G.W., 1998. A perceptually based adaptive sampling algorithm. In: S. Cunningham, W. Bransford, M.F. Cohen, eds. *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. Orlando, FL, USA, 19–24 July 1998. New York: ACM, pp. 299–309.

Bratuž, N., Jerman, T., Gabrijelčič Tomc, H. and Javoršek, D., 2014. Influence of rendering engines on color reproduction. In: D. Novaković, ed. *Proceedings of 7th International Symposium in Graphic Engineering and Design*. Novi Sad, Serbia, 13–14 November 2014. Novi Sad, Serbia: Faculty of Technical Sciences, pp. 395–400.

Bratuž, N., Javoršek, D. and Gabrijelčič Tomc, H., 2015. Defining optimal conditions of colors in 3D space in dependence on gamma values, illumination, and background color. *Journal of Imaging Science and Technology*, 59(4), pp. 040503-1 – 040503-11. https://doi.org/10.2352/J.ImagingSci.Technol.2015.59.4.040503.

Bulbul, A., Capin, T., Lavoué, G. and Preda, M., 2011. Assessing visual quality of 3D polygonal models. *IEEE Signal Process Magazine,* 28(6), pp. 80–90. https://doi.org/10.1109/MSP.2011.942466.

Canny, J., 1986. A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 8(6), pp. 679–698. https://doi.org/10.1109/TPAMI.1986.4767851.

Corsini, M., Larabi, M.C., Lavoué, G., Petřík, O., Váša, L. and Wang, K., 2013. Perceptual metrics for static and dynamic triangle meshes. *Computer Graphic Forum,* 32(1), pp. 101–125. https://doi.org/10.1111/cgf.12001.

Čadík, M., 2004. *Human perception and computer graphic*. Prague: Czech Technical University. [online] Available at: <http://cadik.posvete.cz/papers/psr/cadik04psr.pdf> [Accessed 12 July 2017].

Davis, L.S., 1975. A survey of edge detection techniques. *Computer Graphics and Image Processing*, 4(3), pp. 248–270. http://dx.doi.org/10.1016/0146-664X(75)90012-X.

Deriche, R., 1990. Fast algorithms for low-level vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1), pp. 78–87. https://doi.org/10.1109/34.41386.

Fedorovskaya, A.E., de Ridder, H. and Blommaert, F.J.J., 1997. Chroma variations and perceived quality of color images of natural scenes. *Color Research and Application*, 22(2), pp. 96–110. https://doi.org/10.1002/(SICI)1520-6378(199704)22:2<96::AID-COL5>3.0.CO;2-Z.

Fiorucci, M., Fratton, M., Dulecha, T.G., Pelillo, M., Pravato, A. and Roncato, A., 2017. Computer vision system for the automatic inventory of a cooler. In: S. Battiato, G. Gallo, R. Schettini and F. Stanco, eds. *Image Analysis and Processing – ICIAP 2017: 19th International Conference: Proceedings, Part I*. Catania, Italy, 11-15 September 2017,. Catania: Springer, pp. 575–585. https://doi.org/10.1007/978-3-319-68560-1_51.

Fu, X., Chen, C., Li, J., Wang, C.-C. and, Kuo, J., 2018. Image segmentation using contour, surface, and depth cues (poster). In: *ICIP 2017*. Beijing, China, 17–20 September. IEEE SigPort [online] Available at: <http://sigport.org/2175> [Accessed 10 January 2018].

Gooch, B., Sloan, P.-P.J., Gooch, A., Shirely, P. and Riesenfeld, R., 1999. Interactive technical illustration. In: J. Rossignac, J. Hodgins and J.D. Foley, eds. *I3D '99 Proceedings of the 1999 symposium on Interactive 3D graphics*. Atlanta, GA, USA, 26–29 April 1999. New York: ACM, pp. 31–38. https://doi.org/10.1145/300523.300526.

Geng, W., 2010. *The algorithms and principles of non-photorealistic graphics: artistic rendering and cartoon animation*. Heidelberg, Dordrecht, London, New York: Springer.

Gooch, A.A. and Willemsen, P., 2001. Evaluating space perception in NPR immersive environments. In: *NPAR '02 Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering*. Annecy, France, 3–5 June 2002. New York: ACM Press, pp. 105–110. https://doi.org/10.1145/508530.508549.

Hubel, D.H. and Wiesel, T.N., 1959. Receptive fields of single neurons in the cat's striate cortex. *The Journal of Physiology*, 148(3), pp. 574–591.

Koschan, A. and Abidi, M., 2005. Detection and classification of edges in color images. *IEEE Signal Processing Magazine*, 22(1), pp. 64–73. https://doi.org/10.1109/MSP.2005.1407716.

Leclerc, Y.G. and Zucker, S.W., 1987. The local structure of image discontinuities in one dimension. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(3), pp. 341–355. https://doi.org/10.1109/TPAMI.1987.4767918.

Livingstone, M.S. and Hubel, D., 1988. Segregation of form, color, movement, and depth: anatomy, physiology, and perception. *Science*, 240(4853), pp. 740–749. https://doi.org/10.1126/science.3283936.

Marr, D. and Hildreth, E., 1980. Theory of edge detection. *Proceedings of the Royal Society B.*, 207(1167), pp. 187–217. https://doi.org/10.1098/rspb.1980.0020.

MathWorks, 2018. *Edge: find edges in intensity image*. [online] Available at: <https://www.mathworks.com/help/images/ref/edge.html> [Accessed 10 January 2018].

Miché, P. and Debrie, R., 1995. Fast and self-adaptive image segmentation using extended declivity. *Annales Des Télécommunications*, 50: 401. https://doi.org/10.1007/BF02999740.

Oskoei, M.A. and Hu, H. 2010. *A survey on edge detection methods*. Colchester: University of Essex. [online] Available at: <https://www.researchgate.net/publication/261403578_A_Survey_on_Edge_Detection_Methods> [Accessed 19 December 2017].

Shen, J. and Castan, S., 1992. An optimal linear operator for step edge detection. *CVGIP: Graphical Models and Image Processing*, 54(2), pp. 112–133. https://doi.org/10.1016/1049-9652(92)90060-B.

Sobel, I., 2014. *History and definition of the so-called "Sobel operator", more appropriately named the Sobel-Feldman operator*. [online] Available at: <https://www.researchgate.net/publication/239398674_An_Isotropic_3_3_Image_Gradient_Operator>. [Accessed 13 January 2018].

Strothotte, T. and Schlectweg, S., 2002. *Non-photorealistic computer graphic: modeling, rendering and animation*. San Francisco, USA: Morgan Kaufman.

Trèmeau, A., Nicolas, R. and Dinet, E., 2008. Influence of background on the colour appearance of images. *Color: Design & Creativity*, 3. [online] Available at: <http://www.aic-color.org/journal/v3/jaic_v3_05.pdf> [Accessed 12 June 2017].

Zhou, P., Zhu, J., Su, X., Jing, H. and Zhang, X., 2017. Three-dimensional shape measurement using color random binary encoding pattern projection. *Optical Engineering*, 56(10): 104102. https://doi.org/10.1117/1.OE.56.10.104102.

Yildiz, Z.C. and Capin, T., 2017. A perceptual quality metric for dynamic triangle meshes. *EURASIP Journal on Image and Video Processing*, 12. https://doi.org/10.1186/s13640-016-0157-y.