UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

VITOR AUGUSTO MACHADO JORGE

# Color Wideline Detector and Local Width Estimation

Thesis presented in partial fulfillment
of the requirements for the degree of
Master of Computer Science

Prof. Dr. Edson Prestes
Advisor

Prof. Dr. Luciana P. Nedel
Coadvisor

Porto Alegre, March 2012

**CIP – CATALOGING-IN-PUBLICATION**

Jorge, Vitor Augusto Machado

Color Wideline Detector and Local Width Estimation / Vitor Augusto Machado Jorge. – Porto Alegre: PPGC da UFRGS, 2012.

88 f.: il.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2012. Advisor: Edson Prestes; Coadvisor: Luciana P. Nedel.

1. Wide Line Detector. 2. Color Wide Line Detector. 3. Isotropic filtering. 4. Local density estimate. I. Prestes, Edson. II. Nedel, Luciana P.. III. Title.

*“É preciso amar as pessoas como se não houvesse amanhã...”*
— RENATO RUSSO

# ACKNOWLEDGMENTS

A special thanks goes to my parents, who always believed in me. Without the support of my parents, this work would never be concluded. I love them, and they showed that even when I was in a really bad mood, mumbling all the time. I would like to specially thanks Edson Prestes, who also believed in this work and kept the standards always high, always giving me advice and putting me in the right direction when I was lost and tired. I would like to thank Luciana Nedel, for the scholarship that she provided so that I could start this work. Finally I would like to thank all my family and friends for giving me the means to keep my sanity during this long and tiring process. Finally, I would like to thank all the crew of professors from the Computer Graphics, Interaction and Image Processing group, for all their support when I needed. This whole process was painful, I lost a lot in the process.

In honor of my grandmother, Ziza.

# TABLE OF CONTENTS

# LIST OF FIGURES

# ABSTRACT

Line detection algorithms are used by many application fields, such as computer vision and automation, as a basis for more complex analysis. For instance, line information can be used as input to object detection algorithms or even attitude estimation in flying robots. One way to detect lines is to use an isotropic nonlinear filtering procedure called the Wide Line Detector (WLD). This algorithm is effective to highlight the line pixels in gray scale images, separating dark or bright lines. However, line detection algorithms are not normally concerned with the pixel-wise estimation of thickness. If available, such information could be further explored by computer vision algorithms. Furthermore, color is extensively used in computer vision as an object discriminant, but not by the WLD. In this work, we propose the extension of the WLD to color images. We also develop a method that allows the estimation of the line width locally using only the density information and no border or center line information. Finally, we develop a new monotonically increasing kernel that is more efficient and yet effective to detect lines than the monotonically decreasing kernels used by the WLD. Finally, we devise a way ro obtain the wideline thickness from the density estimate obtained from the similarity between pixels, reverting the process used by the WLD to determine which kernel should be used. We perform several experiments with the proposed method, considering different parameters, and comparing it to the traditional WLD algorithm to assess the effectiveness of the method.

**Keywords:** Wide Line Detector, Color Wide Line Detector, isotropic filtering, local density estimate.

**Um Detector de Linhas Largas para imagens coloridas e Estimativa Local de Largura da linha**

# RESUMO

Algoritmos de detecção de linhas são usados em muitos campos de aplicação, tais como visão computacional e automação como base para análises mais complexas. Por exemplo, a informação de linha pode ser usada como dado de entrada para algoritmos de detecção de objetos ou mesmo para a estimativa da orientação espacial de robôs aéreos. Uma das formas de detectar linhas é através do uso de um processo de filtragem não linear chamado de Wide Line Detector (WLD). Esse algoritmo é eficaz na marcação de pixels de linha em imagens em tons de cinza, separando linhas claras ou linhas escuras. Contudo, os algoritmos de detecção de linha não estão normalmente preocupados com a estimativa de largura local individual associada a um pixel. Se disponível, tal informação poderia ser explorada por algoritmos de visão computacional. Além do mais, a informação de cor também é extensivamente usada em visão computacional como um discriminante de objetos, mas o WLD não a usa. Neste Trabalho, nós propusemos a extensão do WLD para imagens em cores. Nós também desenvolvemos um novo kernel monotonicamente crescente que é mais eficiente e mais robusto para detectar linhas do que que os kernels monotonicamente decrescentes usados pelo WLD. Por fim, desenvolvemos uma maneira de obter uma estimativa de largura de linha partindo da densidade local associada a similaridade entre pixels, revertendo o processo usado pelo WLD para estimar qual kernel deve ser usado. Diversos experimentos foram realizados com o método proposto considerando diferentes parâmetros, além da comparação com o WLD tradicional, para analizar a eficácia do método.

**Palavras-chave:** Estimativa de Densidade Local, Detector de Linhas Largas, Kernels Isotrópicos, Kernels Monotonicamente Decrescentes, Kernels Monotonicamente Crescentes, Differenças de Cores Perceptuais.

# 1 INRODUÇÃO

A detecção de objetos relevantes, tais como faces e ruas de uma dada imagem, é uma tarefa difícil para a visão computacional e reconhecimento de padrões. Uma forma de abordar este problema é começar com a detecção de estruturas básicas, de tal modo que estruturas mais complexas possam ser obtidas posteriormente. Por exemplo, linhas e estruturas curvilineares tem um papel importante em diferentes áreas, incluindo inteligência artificial e reconhecimento de padrões (LI; LIU; QIAN, 2009; SUN; LIU, 2009; WANG; WU; HU, 2009), na medicina (ZHANG; LIU, 2007; LIU; ZHANG, 2008; JIANGUO et al., 2009; PAKTER et al., 2010) e imagens aéreas (FISCHLER; TENENBAUM; WOLF, 1981; LIU et al., 2008; CHIANG et al., 2009). Estas estruturas curvilineares normalmente variam ao longo de seu cumprimento, por exemplo, a direção de ruas, a largura dos galhos de uma árvore, etc. O processo de detecção de tais estruturas envolve a demarcação do agrupamento completo de pixels que o compõem uma imagem digital, para extrair estruturas importantes como direção comprimento e largura.

Em robótica, estruturas curvilineares fornecem informações valiosas sobre o ambiente e sobre o robô em diferentes tarefas, incluindo localização de objetos de interesse, além de tarefas de controle, estabilização e orientação. A pesquisa corrente sobre Veículos Aéreos não Tripulados (VANTs) mostra que estas estruturas podem ser usadas inclusive para obtenção de informação sobre a localização do robô. Métodos tradicionais baseados no horizonte e geometria epipolar possuem diversas limitações incluindo a falta de estruturas, necessidade de extrair os parâmetros a partir de decomposição de matrizes (BAZIN et al., 2008). Além disso, a informação de GPS pode ser imprecisa quando a aeronave (MONDRAGON et al., 2010) voa baixo. Mesmo os odômetros visuais de drones podem falhar em baixas altitudes (GUIZILINI; RAMOS, 2011). Portanto, técnicas inovadoras de reconhecimento de estruturas básicas podem abrir espaço para novas estratégias para melhorar o controle dinâmico de drones. Por exemplo, BAZIN et al. (2008) usam câmeras catadióptricas ao invés de câmeras tradicionais, para extrair um grande número de linhas, estimando o os pontos de fuga da imagem e os rastreando durante longas sequências de vídeo de forma precisa e robusta. Eles exploram a forte regularidade exibida em estruturas conhecidas pelo homem presentes em áreas urbanas. HWANGBO; KANADE (2011) observa que esta regularidade exibe segmentos de linha que são paralelos ou ortogonais ao vetor da força da gravidade. Usando esta informação o VANT pode estimar sua própria pose, incluindo posição altitude e orientação, além de corrigir erros de odometria acumulados. Os segmentos de linha são classificados usando um conjunto de premissas sobre a quantidade de pontos de fuga existentes[1] na cena. Mondragoń et

[1]Quando uma cena tridimensional é projetada em um plano usando uma transformação projetiva, o ponto de fuga é um ponto no plano de projeção onde duas ou mais linhas paralelas (mas não paralelas ao

al. (MONDRAGON et al., 2010) propõe um algoritmo online de estimativa 3D da pose do drone usando estruturas métricas e projetivas de marcadores terrestres ou do próprio heliporto com respeito ao sistema de coordenadas calibrado do VANT. O algoritmo estima com sucesso a posição 3D do helicóptero em baixas altitudes em tempo real. Com esta estimativa é possível realizar tarefas de controle complexas como pouso autônomo e tarefas de carga e descarga. Todos estes trabalhos se aproveitam de estruturas visuais elementares presentes no ambiente.

A informação de cor é também bastante relevante para a robótica em muitas áreas. Ela pode ser usada para classificar vegetação automaticamente em ambientes naturais usando imagens aéreas de alta resolução obtida em vôos de baixa altitude (REID; RAMOS; SUKKARIEH, 2011); para navegação usando câmeras monoculares em Veículos Terrestres não Tripulados (UGVs) usados em ambientes perigosos tais como áreas de contaminação química, radioativa ou biológica, evitando a necessidade de descontaminação de pessoal e sensores (MIKSIK et al., 2011); para checar se a sequência de cor de fios e cabos está correta, enquanto estes estão sendo presos a conectores por um robô de inspeção de cabos (GHIDONI; FINOTTO; EMANUELEMENEGATTI, 2011); ou usada juntamente com informação ode textura por algoritmos de segmentação para classificar rios, céu e margens (ACHAR et al., 2011). Note que cabos e rios são ambos estruturas curvilineares largas com cores específicas.

Recentemente, LIU; ZHANG (2005); LIU; ZHANG; YOU (2007) introduziu um algoritmo de detecção de estruturas curvilineares chamado de Detector de Linhas Largas (WLD) baseado no algoritmo SUSAN de processamento de imagens (SMITH; BRADY, 1997). O WLD funciona em imagens em escala de cinza em cada pixel para cada pixel. A estimativa é uma soma ponderada de medidas de similaridade obtidas ao comparar o pixel central com os outros pixels dentro do kernel. Depois disso, o WLD determina se o pixel central pertence a uma linha ou à imagem de fundo usando um conjunto de equações (ver Capítulo 4). LIU; ZHANG; YOU (2007) determinam o tamanho do kernel *a priori*, de acordo com a largura máxima da linha a ser detectada. O algoritmo requer a seleção de linhas claras ou escuras *a priori*. Além disso, é sugerido que a largura da linha possa ser obtida pelo seguinte processo: 1) a linha completa é marcada usando o WLD, seguida da aplicação de um algoritmo de geração de esqueletonização; 2) o resultado deste processo é usado com as bordas para subsequentemente medir a largura (espessura) da linha.

## 1.1 Contribuições

Este trabalho propõe melhorias ao método WLD. Primeiro, estendemos o método para trabalhar com imagens coloridas. O algoritmo tradicional desconsidera a informação de cor. Segundo, enquanto o WLD originalmente usa a relação entre estimativa de densidade e largura para selecionar o tamanho de kernel que detecta todas as linhas até um valor máximo de espessura (fornecido *a priori*), nós desenvolvemos uma função de transferência que converte a estimativa de densidade em uma largura correspondente para todos os pixels que podem ser detectados pelo algoritmo. Esta função gera um mapa de largura associado a todas os pixels de todas as estruturas curvilineares presentes na imagem. Em seguida, propusemos um novo kernel que não é monotonicamente decrescente. O uso de tal kernel reduz o custo computacional da detecção de linhas em relação ao algoritmo tradicional, uma vez que ele requer um raio menor que o dos usados tradicionalmente pelo algoritmo. Isso resulta em menos pixels no processo, além de manter ou melhorar os

plano de projeção) convergem.

resultados do algoritmo.

## 1.2 Estrutura desta Dissertação de Mestrado

Esse trabalho é organizado da seguinte forma. O Capítulo 3 apresenta trabalhos atuais sobre detecção de estruturas curvilineares em imagens. O Capítulo 4 revê os conceitos principais do WLD, com foco em como o WLD estima o raio do kernel. Nós apresentamos o método proposto, chamado de Color Wide Line Detector (CWLD), no Capítulo 5, juntamente com experimentos e imagens artificiais. Primeiro, é explicado como adaptar diferenças de cor para o método WLD para gerar o CWLD. Então, executamos experimentos que mostram que kernels monotonicamente crescentes são mais eficientes que os monotonicamente decrescentes. A função que converte a estimativa de densidade local em largura de linha, para kernels monotonicamente decrescentes e crescentes conclui o capítulo. O Capítulo 6 apresenta experimentos e resultados em imagens naturais, incluindo a comparação de efetividade entre CWLD e WLD para detectar linhas; uma análise de desempenho de cada kernel para detectar pixels pertencentes a linhas; e análise estatística da estimativa de largura obtida em cada imagem natural, com diferentes combinações de kernel e diferença de cor. Tabelas com os resultados completos são extraídas e reapresentadas parcialmente ao longo do Capítulo para facilitar o entendimento dos resultados. O Capítulo 7 apresenta a conclusão e trabalhos futuros, fornecendo comentários sobre os melhores resultados obtidos considerando diferenças de cor e kernels. Incluímos ainda um apêndice com contendo algumas equações de conversão de cores e de diferença de cor relevantes para este trabalho.

# 2 INTRODUCTION

The detection of relevant objects, such as faces and roads from a given image, is a difficult task for computer vision and pattern recognition. One way to approach this problem is to start with the detection of basic structures, so that more complex information can be obtained thereafter. For instance, lines and curvilinear structures play a vital role in different sub-areas, including machine intelligence and pattern recognition (LI; LIU; QIAN, 2009; SUN; LIU, 2009; WANG; WU; HU, 2009), medical (ZHANG; LIU, 2007; LIU; ZHANG, 2008; JIAN-GUO et al., 2009; PAKTER et al., 2010) and aerial imaging (FISCHLER; TENENBAUM; WOLF, 1981; LIU et al., 2008; CHIANG et al., 2009). These curvilinear features commonly vary across their length, e.g., the direction of roads, the width of tree branches, etc. The detection process of such basic structures involves the highlighting of the whole group of image pixels that comprises them, to extract characteristics such as direction and width.

In robotics, curvilinear features provide valuable information about the environment and the robot for different tasks, including target localization, control and orientation tasks. Current research on Unmanned Aerial Vehicles (UAVs) shows that these features are quintessential for the localization problem. Traditional methods based on horizon and epipolar geometry have several limitations that include lack of features, need to retrieve motion parameters from matrix decomposition, and the rotation-translation ambiguity (BAZIN et al., 2008). Also, the GPS information is often inaccurate if the aircraft is flying at low altitudes (MONDRAGON et al., 2010). Even the traditional UAV visual odometers fail at low altitudes (GUIZILINI; RAMOS, 2011). Therefore, they open space to new strategies to improve UAV dynamic control. For instance,BAZIN et al. (2008) use catadioptric instead of traditional perspective cameras to extract a large number of lines, estimate the associated vanishing points, and track them even during long video sequences accurately and robustly. They explore the strong regularity of available man-made structures present in urban areas. HWANGBO; KANADE (2011) observe that this regularity exhibits line segments that are either parallel or orthogonal to the gravity force vector. Using this information, the UAV can estimate its own attitude and correct accumulated errors. They classify line segments using a set of assumptions about the number of vanishing points[1] in the scene (horizontal and vertical). The resulting algorithm uses the line segments directly instead of the vanishing points to obtain another robust attitude estimation algorithm. MONDRAGON et al. (2010) proposes a real time UAV 3D pose estimation to obtain metric and projective components of a landmark or helipad with respect to the UAV calibrated camera coordinate system. The algorithm successfully estimates

[1]When a three dimensional scene is projected into a plane using a perspective transformation, the vanishing point is a point in the projection plane where two parallel lines (not parallel to the view plane) converge.

the helicopter's 3D position at low altitudes. With this estimate it is possible to perform complex control tasks such as accurate positioning at low altitudes, autonomous landing and payload dropping.

Color information is also relevant for robotics in a broad range of areas. It has been used to automatically classify vegetation in natural environments based on high resolution aerial imagery acquired by a low flying UAV (REID; RAMOS; SUKKARIEH, 2011); for navigation with a monocular camera by Unmanned Ground Vehicles (UGVs), used in hazardous environments, such as chemical, nuclear and biological contamination areas, avoiding complications with sensor decontamination (MIKSIK et al., 2011); to check if the color sequence of wires are correct, while the cables are being crimped to connectors by a cable inspection robot (GHIDONI; FINOTTO; EMANUELEMENEGATTI, 2011); or used along with texture information by a segmentation algorithm to classify river, sky and shore (ACHAR et al., 2011). In the cable sequence analysis and classification of rivers, note that both the cables and rivers are wide lines with specific colors.

LIU; ZHANG (2005); LIU; ZHANG; YOU (2007) introduced a robust curvilinear feature detection method called Wide Line Detector (WLD) based on the SUSAN image processing framework (SMITH; BRADY, 1997). The WLD works on gray scale images at every pixel, in combination with an isotropic kernel to obtain the local density estimate. This estimate is a weighted sum of the similarity measures obtained by comparing the central pixel with every other pixel. After, the WLD determines if the central pixel belongs to a line or background region using a set of equations (see Chapter 4). Liu et al. determine the kernel size *a priori*, according to the maximum line width that should be detected. The algorithm also requires the selection of bright or dark lines *a priori*. Furthermore, they suggest (LIU; ZHANG; YOU, 2007) that the line width could be obtained through the following process: 1) the whole line is marked, using the WLD, and thinned, using any thinning algorithm. 2) the result of the thinning algorithm is used, together with the line border, to subsequently measure the line width.

## 2.1 Contributions

In this work we propose some improvements to the WLD method. First, we extend it to work with color images. The original method disregards color information, working only with gray scale images. Second, while the WLD originally uses the relation between density estimate and width to select the minimum kernel size that can detect all the line features up to the maximum line tickness (provided *a priori*), we developed a transfer function that converts the density estimate into a corresponding line width for all the line pixels that can be detected by the kernel. This function generates a map of widths associated to the curvilinear features detected in the entire image. The line width can be used to classify different aspects of a real environment, for instance, a mobile robot could extract the lines of a wall or floor to determine its position accurately. Furthermore, it could be used in combination with a tracking algorithm (COMANICIU; RAMESH; MEER, 2000) to monitor user motion for recognizing gestures. Then, we propose a new kernel that is not monotonically decreasing. The use of such kernel reduces the computational cost of line detection, since it requires smaller kernel radius than the others. This results in few pixels to process, while maintaining or improving the quality of the results.

## 2.2 Structure of this Dissertation

This Work is organized as follows. Chapter 3 presents the current work on curvilinear line detection. Chapter 4 reviews the main concepts of the WLD, with a focus on how the WLD estimates the size of the kernel radius. We demonstrate our method, called Color Wide Line Detector (CWLD), in Chapter 5, together with some experiments with artificial images. First, we explain how to adapt the color-differences to the WLD to generate the CWLD. Then, we perform several experiments that show that the monotonic increasing kernels are more efficient than monotonic decreasing kernels. The transfer-functions that convert the local density estimates into width measurements, for both the monotonic increasing and decreasing kernels, conclude this chapter. Chapter 6 presents the experiments and results considering natural images, including the comparison of effectiveness between CWLD and WLD to detect lines; the performance analysis of each kernel to detect line pixels; and the statistical analysis of the width estimations obtained for each natural images, with different combinations of kernels and color-difference equations, with detailed tables, which contain complete statistical information from this work. Throughout the Chapter, we present relevant parts from these tables, making use of smaller tables showing relevant information to improve text readability. Chapter 7 presents the conclusion and future works, providing comments regarding the best results with color-difference and kernel, as well as the possible ramifications of this work. We also include an Appendix section, containing some color transformations and color-difference equations relevant to this work.

# 3 RELATED WORK

In this Chapter, we present relevant works related to the scope of this Work. One of the first techniques developed in the image processing field to detect structures (like lines, circles, etc.) was the Hough Transform (DUDA; HART, 1972). It detects structures like lines, circles or other desired features only when the corresponding equation is available. To detect a line, for instance, the algorithm takes advantage of the point line duality to construct a voting histogram. Basically, it uses the attributes of the line to count and store the number of matching pixels in the histogram. The most probable lines are those with the highest number of votes. Similar approaches apply to any feature, other than lines, that needs to be detected. The Hough transform has a couple of problems. For instance, it does not filter adequately all false positive features detected (GIOI et al., 2010) and the voting map can use a considerable amount of memory or be affected by aliasing (FERNANDES; OLIVEIRA, 2008).

The method proposed by GEMAN; JEDYNAK (1996) detects 1-D linear structures from a starting point and specific direction in the line. It is used to track highways over long distances in real time without manual intervention. Basically, it considers that a main road is constructed from an ordered and equally spaced sequence of small line segments with width and length equal to 2 and 12 pixels, respectively. Two subsequent segments have an angular constraint, i.e., the angle between them must be in the interval of $[+5, -5]$ degrees. The algorithm analyzes local image data, around the current position, to determine the line continuation that has a relative brightness difference similar to that of the line being currently built. On the other hand, FISCHLER; TENENBAUM; WOLF (1981) algorithm detects and delineates curvilinear structures, such as roads, in low-resolution aerial images. It combines information from different sources, including the output of several line and edge detectors, and the road maps available. A road is detected using graph search and dynamic programming techniques.

KWEON; KANADE (1994) propose a method for building high-level terrain descriptions from an elevation map. The contour lines obtained from this map are analyzed using directional derivatives to extract terrain features such as peaks, pits, ridges, and ravines in gray scale.

KOLLER et al. (1995) use a nonlinear combination of linear filters to search for lines across scale-space. Their method does not deal with complex structures, like crossings and junctions, and has difficulty to interpret the different meanings of structures across scales. This happens because structures from a given scale might merge or disappear in larger scales. LINDEBERG (1996) improves the scale-space analysis performing automatic scale selection. This selection uses maxima over scales and aims to better interpret structures like edges, ridges and corners.

STEGER (1998) uses differential geometry and Gaussian masks to estimate the first

and second derivatives of an image, to obtain wide line information. This method selects the center lines as the maxima over scale-space. After, it uses a modification of Canny edge detector (CANNY, 1986) to identify the edges and estimate the line width with sub-pixel precision.

The method proposed by ZHANG; LIU (2007) marks the pixels and estimates the center line of vessels in retinal images. First, it defines, *a priori*, the length, $l$, and the width, $w$, of the minimum vessel to be detected. Then, the method generates several line segments $s_0$ with length $l$, starting from every pixel,$\mathbf{p}$, in the image in different orientations, $\theta$. From each pixel that belongs to a given segment, the method draws two segments, $s_1$ and $s_2$, that are perpendicular to the former segment with length slightly higher than the $w$. For all pairs of segments $s_1$ and $s_2$ generated from $s_0$, the method determines the segment mean intensity, that is, the average of the intensities of the pixels belonging to the segment. Next, they calculate the contrast ratio, $c$, between the generating pixel and the mean intensity of the pixels belonging to the segment. The method considers $s_0$ as a vessel pixel if the number of perpendicular segments, whose $c$ is less or equal a given perceptual contrast threshold, is sufficiently large. This method is relevant because it introduces the concept of perceptual contrast (ZHANG; LIU, 2007).

QIN-LI et al. (2008) proposes a dark (or bright) line detection method based on the first order derivative of the Gaussian function. This method uses a set of oriented Gaussian derivative filters, where the zero crossings are considered as the center lines. The line width is simultaneously estimated using the difference between the local maximum and minimum. This method does not perform well in noisy images and the tests are provided for few images only.

LI; CHANG; ZHU (2009) propose an approach to detect lines based on density estimation. This density is calculated using a set of rules to speed-up the calculation, for instance, pixels whose intensity value is zero are disregarded, what is particularly convenient for binary images. They apply thresholding rules over the obtained density estimate to perform background pixels removal. However, the resulting image is not entirely free of undesirable pixels, so a noise removal technique is employed. After, they perform a set of procedures including non-maximum suppression to detect the center line pixels. Finally, they connect them to generate the vector representation of the detected lines. This method is very efficient, and claims to be the first to use the density estimate, but it was devised after the WLD (LIU; ZHANG, 2005; LIU; ZHANG; YOU, 2007), and they differ only in how the density estimate is computed and how the segmentation is performed.

GIOI et al. (2010) proposed a line segment detector (LSD) that runs in linear time *O(n)* and does not require parameter tuning. It detects and controls false positive features generating accurate results. The algorithm uses a combination of region growing and geometric structure matching to detect segments. It does not enhance all pixels of curvilinear features and often generates aliased segments that do not always match exactly the feature edges.

The WLD (LIU; ZHANG, 2005; LIU; ZHANG; YOU, 2007) detects lines by means of a non-linear isotropic filtering algorithm. The method estimates the line direction finding the longest axis of symmetry. It detects bright or dark lines using a *step function* that selects bright or dark responses. This method is particularly useful to segment curvilinear features, because it eliminates the need to detect edges. Edge detection can be difficult, since it can easily be influenced by noise. In addition, determining specific edge positions can be subjective (ZHANG; LIU, 2007). Edge detection in color images is still an active research topic (VERMA, 2010). We observed that the density estimation could be

improved in the WLD using perceptual color-differences instead of intensity differences. The addition of color allows the detection of line pixels in natural color images, as you will see in Chapter 6. Moreover, the WLD also uses an interesting relationship between density estimate and the line width to determine the smallest kernel needed to detect a line of width $w$ – thickness. We take advantage of this relationship to develop a new kernel that is more efficient than those proposed by the WLD. It also allows the extraction of the line width locally, through a transfer-function that receives the local density estimate as input and returns the line width estimation, without using the the center line or the borders. This modifications give rise to our method called the CWLD. The WLD and the CWLD algorithms are discussed in the next Chapters. The WLD was taken as the basis for this work because we are particularly interested in extracting higher level information from kernel density estimates – e.g. the line thickness. The idea of extracting complex information from a singled function is of particular interest to the authors of the present work.

# 4 REVISITING THE WIDE LINE DETECTOR

The WLD applies a nonlinear filter to a gray scale image that combines a chosen circular mask, a set of functions and rules to determine the line pixels. Initially, the WLD computes the ***W***eighed ***M***ask having ***S***imilar ***B***rightness (***WMSB***)

$$m(\mathbf{p}_0) = \sum_{\mathbf{p}} s(\mathbf{p}_0, \mathbf{p}) K(||\mathbf{p} - \mathbf{p}_0||) \tag{4.1}$$

for every pixel in the image, where $\mathbf{p_0}$ is the center pixel in the kernel at position $(x_0, y_0)$, $\mathbf{p}$ is a pixel in the kernel and $K(.)$ is the kernel profile and

$$s(\mathbf{p}_0, \mathbf{p}) = \text{ sech} \left[ \left( \frac{I(\mathbf{p}_0) - I(\mathbf{p})}{t} \right)^5 \right] \tag{4.2}$$

where $t$ is the *brightness contrast threshold*, which is the standard deviation of the intensities in the image, $\text{sech}(.)$ is the hyperbolic secant function and $I(.)$ is the pixel intensity. Equation 4.2 is a smoothing function used to avoid simple binary decision cut when $I(\mathbf{p}_0) - I(\mathbf{p}) > t$.

The WLD uses the uniform, $K_u(.)$, and Gaussian, $K_g(.)$, kernel profiles as $K(.)$. Their responses are represented by

$$K_u(d) = \begin{cases} a & \text{, if } d \leq r \\ 0 & \text{, otherwise} \end{cases} \tag{4.3}$$

$$K_g(d) = \begin{cases} \exp\left(\frac{-d^2}{2r^2}\right) & \text{, if } d \leq r \\ 0 & \text{, otherwise} \end{cases} \tag{4.4}$$

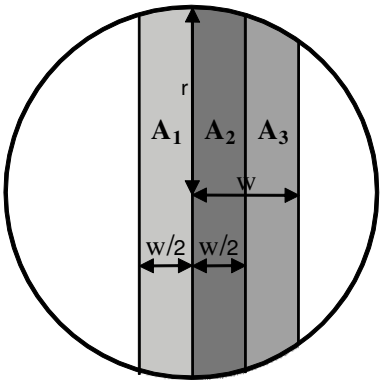
where $a$ is the height of the uniform kernel, typically $a = 1$, and $r$ is the radius of the kernel. Even though only such kernels have been used, the theory behind the method is valid for monotonically decreasing kernels (LIU; ZHANG; YOU, 2007).

After, the line response $L(\mathbf{p})$ is computed

$$L(\mathbf{p}) = \begin{cases} m_{max} - m(\mathbf{p}) & \text{, if } m(\mathbf{p}) < m_{max} \\ 0 & \text{, otherwise} \end{cases} \tag{4.5}$$

where $m_{max}$ is a geometric threshold used to accept a pixel as belonging to the line. $L(\mathbf{p})$ will be computed considering that $\mathbf{p}$ is the kernel center. When $L(\mathbf{p}) = 0$, the pixel $\mathbf{p}$ is considered as background. Otherwise, $\mathbf{p}$ is a line pixel, with probability proportional to $L(\mathbf{p})$.

Figure 4.1: A1+A2 corresponds to the volume occupied by $\lambda$ when it is centered in the kernel, while A2+A3 is the volume of the same line when one of its edges crosses the center of the kernel.




To understand that, consider a line $\lambda$ that is parallel to the vertical Cartesian axis, with width $w$, crossing an isotropic kernel $K$ with radius $r$ and volume $V_K$, as in Figure 4.1.

The volume occupied by $\lambda$ is defined by

$$V_\lambda = \int\int_\lambda K(||(x,y)||)dxdy \tag{4.6}$$

considering that the kernel center is at $\mathbf{p}_0 = (0,0)$. It corresponds to the continuous version of Equation 4.1 in ideal conditions. That is, $s(\mathbf{p}_0, \mathbf{p}) \approx 1$ insofar $I(\mathbf{p}_0) - I(\mathbf{p}) \rightarrow 0$. For values of $(I(\mathbf{p}_0) - I(\mathbf{p}))$ larger than $t$, $s(\mathbf{p}_0, \mathbf{p}) = 0$. We can see that $V_\lambda \in V_K$.

The portion $V_\lambda$ occupied by $\lambda$ is associated to the volume $V_K$ by two equations. The first one is given when the edge of $\lambda$ touches the center of $K$.

$$V_E(w,r) = \int_0^w \int_{-\sqrt{r^2-x^2}}^{\sqrt{r^2-x^2}} K(||(x,y)||)dxdy \tag{4.7}$$

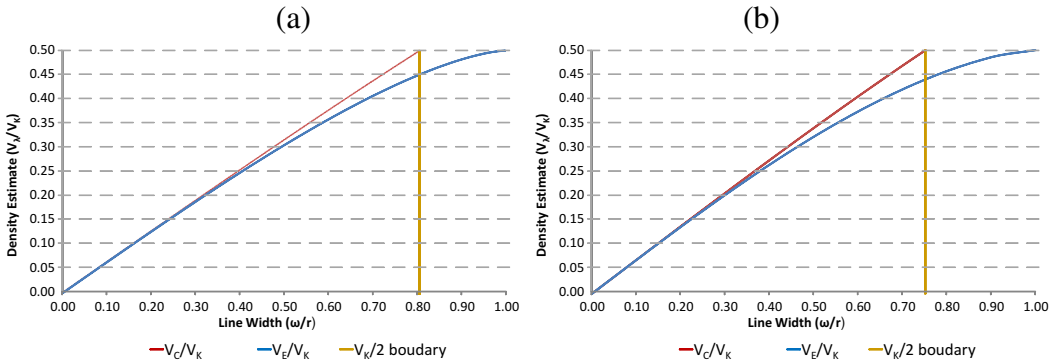
When the kernel is at the center of the line, we reach a point where each half of the line occupies equal volume. In this case, the following equation is valid

$$V_C(w,r) = \int_{-\frac{w}{2}}^{\frac{w}{2}} \int_{-\sqrt{r^2-x^2}}^{\sqrt{r^2-x^2}} K(||(x,y)||)dxdy \tag{4.8}$$

According to Liu et al. (LIU; ZHANG; YOU, 2007), if the kernel is uniform or monotonically decreasing, when the kernel center is inside a line region, the inequality $V_E \leq V_\lambda \leq V_C$ is true. This can be observed on Figure 4.1. From that, Liu defines $m_{max} = \alpha V_K$, where $\alpha \approx 1/2$ and depends on the $K(.)$ used. When a pixel $\mathbf{p}$ belongs to a line $V_E \leq m(\mathbf{p}) \leq V_C$, that is, a line pixel has been detected if $m(\mathbf{p}) < \alpha V_K$ (see Equation 4.5).

Liu determines the minimum kernel radius, $r$, required to detect all the lines with width less than or equal to $w$, analyzing the relationship between $w$ and the kernel volume occupied by the line according to its position inside the kernel. Figures 4.2 (a) and (b) illustrate this relationship for $K_u$ and $K_g$ considering Equations 4.7 and 4.8, respectively.

Figure 4.2: The relationship of $w$ (as the ratio of $r$, i.e., $w/r$) and $V_\lambda$ (as the ratio of $V_K$, i.e., $V_\lambda/V_K$) using (a) $K_u$ and (b) $K_g$ considering Equations 4.7 and 4.8.




They show that by increasing the line width, Equations 4.7 and 4.8 reach $V_K/2$ at different widths. That is, if Equation 4.7 is used, both kernels return the density estimate of $V_K/2$ when $w = r$, while if we use Equation 4.8 we reach the same distribution at considerably smaller widths: $w \approx 0.8r$ for $K_u$ and $w \approx 0.75r$ for $K_g$. This implies that any line having $w > 0.8r$ for $K_u$ and $w > 0.75r$ for $K_g$ will no longer be completely detected and hence will be considered as background, since depending on its position, $V_C > V_K/2$, i.e., $m(\mathbf{p}) \geq m_{max}$ in Equation 4.5.

Finally, Equation 4.2 is modified with the addition of a step function to detect bright or dark lines:

$$s(\mathbf{p}_0, \mathbf{p}) = \begin{cases} \text{sech}\left[\left(\frac{\Delta(\mathbf{p},\mathbf{p}_0)}{t}\right)^5\right] & \text{, if } \Delta(\mathbf{p}_0, \mathbf{p}) > 0 \\ 1 & \text{, otherwise} \end{cases} \tag{4.9}$$

and

$$\Delta(\mathbf{p}_0, \mathbf{p}) = \begin{cases} I(\mathbf{p}_0) - I(\mathbf{p}) & \text{for bright lines detection} \\ I(\mathbf{p}) - I(\mathbf{p}_0) & \text{otherwise} \end{cases} \tag{4.10}$$

# 5 COLOR WIDE LINE DETECTOR

In this Chapter, we present the core of this work. It corresponds to an extension of the WLD, called Color Wide Line Detector (CWLD). The CWLD presents three important differences over the WLD. First, it considers color information to detect wide lines. Second, it uses a new efficient kernel that detects wider lines than those used by the original WLD considering that they have the same kernel radius. Lastly, it uses a transfer function that computes the line width locally from the relationship of width and density distribution associated with the line response. This process does not require center line and border information to compute the line width. In the following Section we describe formally the CWLD, presenting the equations to elicit lines and their width.

## 5.1 Incorporating color information

As mentioned in Chapter 4, the WLD identifies lines only in gray scale images using the *WMSB*. In this section, we generalize the WLD to use color-difference, $\Delta E$, instead of intensity difference. We call this generalization the ***W***eighed ***M***ask having ***S***imilar ***C***olors (**WMSC**). Basically, the WMSC is equal to the r.h.s. of Equation 4.9 considering

$$s(\mathbf{p}_0, \mathbf{p}) = \begin{cases} \text{sech}\left[\left(\frac{\Delta E(\mathbf{p}_0,\mathbf{p})}{t}\right)^5\right] & \text{, if } \Delta_C(\mathbf{p}_0, \mathbf{p}) > 0 \\ 1 & \text{, otherwise} \end{cases} \tag{5.1}$$

and

$$\Delta_C(\mathbf{p}_0, \mathbf{p}) = \begin{cases} Q(\mathbf{p}_0) - Q(\mathbf{p}) & \text{, for bright lines detection} \\ Q(\mathbf{p}) - Q(\mathbf{p}_0) & \text{, otherwise} \end{cases} \tag{5.2}$$

where $\Delta E(\mathbf{p}, \mathbf{p}_0)$ is the color-difference between $\mathbf{p}$ and $\mathbf{p_0}$, and $Q(.)$ is the appropriate intensity indicator defined by the selected color space.

Choosing $\Delta E(\mathbf{p}, \mathbf{p}_0)$ is not an easy task since the color models available may not be suitable for complex scenarios involving color comparison. Color appearance is highly associated with each individual, i.e., the color experience, such as looking at a reddish evening sky, is subjective for every one of us (qualia problem (CRICK, 1995)). Industries commonly use CIE (Commission Internationale de l´Eclairage) $L^*a^*b^*$ and $L^*C^*h$ as reference (FAIRCHILD, 2005; BERNS, 2000), because these spaces were designed to be perceptually uniform (FAIRCHILD, 2005). Theoretically, similar colors should have difference smaller than 1 unit in this space. However, in practice CIE $L^*a^*b^*$ and $L^*C^*h$ produce differences larger than 1 unit even for some perceptually similar colors. As a result, several color-difference equations have been proposed (a review is presented in (LUO, 2002)).

For color images, the information at each pixel is now a color vector, so we need to convert the line response equations from Chapter 4 accordingly. The first step is to use the norm of the color-difference vector, instead of the pixel intensity difference. For the RGB color space, $\Delta E(\mathbf{p}, \mathbf{p}_0)$ is the Euclidean distance between the color of pixels $\mathbf{p}$ and $\mathbf{p}_0$. $Q(\mathbf{p})$, in this case, is the intensity, computed by

$$Q(\mathbf{p}) = I(\mathbf{p}) = \alpha R(\mathbf{p}) + \beta G(\mathbf{p}) + \gamma B(\mathbf{p})$$

where $R(.), G(.)$ and $B(.)$ are the color components red, green and blue, respectively, of pixel $\mathbf{p}$ and $\alpha, \beta, \gamma \in [0, 1]$ are the weighing factors given according to the Illuminant. For CIE $L^*$ difference we use Equations 4.9 and 4.10 , where $I(\mathbf{p})$ is replaced by CIE luminance $L^*$ of pixel $\mathbf{p}$. For other CIE color spaces, we associate $\Delta E$ as the chosen available color-difference. In this case, as $\Delta E$ alone does not provide enough information to classify bright or dark lines in an image, we set $Q(\mathbf{p})$ as the luminance $L^*$ of pixel $\mathbf{p}$.

In this work, we have tested several color-difference equations (CIELAB 1976 $L^*a^*b^*$, CIE 1994, CIE 2000 and CMC(*l, c*) (FAIRCHILD, 2005)), the gray scale (original WLD), CIE $L^*$ differences and the simple RGB Euclidean distance. In our experiments (see Section 6) we have observed that CIE color-differences produce consistent results with the threshold of 5 units for some images, which is also in accordance to previous observation that similar colors are distant from each other of at most 5 units (FAIRCHILD, 2005). However, we also observed in our experiments that this threshold was sometimes too susceptible to noise. As the WLD uses the standard deviation of the gray scale intensities of the image (LIU; ZHANG; YOU, 2007), we decided to use the same strategy, calculating the standard deviation of the color-differences instead of using a fixed threshold of 5 units.

The separation of bright and dark lines using Equations 4.10 or 5.2 artificially augments the density estimate to remove undesired lines (bright or dark), classifying them as background. It does that through Equations 4.5, 4.9 and 4.10. That is, the dark (bright) pixel responses are excluded by setting the density estimate to values wider than $m_{max}$, i.e. set to background. When a pixel shows a response for dark (bright) lines, probably this pixel is mostly darker (brighter) than its surrounding pixels, that is, $I(\mathbf{p}_0) - I(\mathbf{p}) < 0$ for most of the pixels (or $I(\mathbf{p}) - I(\mathbf{p}_0) < 0$ to exclude bright lines). In this way, when Equation 4.10 sets every pixel comparison with negative values to 0, the resulting value of Equation 4.9 becomes one. As a consequence, the density estimate of such pixels reach values greater than $m_{max}$, and therefore they are excluded through Equation 4.5. Besides, the wide line borders may have very low density estimates when compared to those associated with the other line pixels. As the WLD augments the density of border pixels, their line responses can become visually more similar to those of the other line pixels.

This separation strategy, employed by the WLD and CWLD, may lead to problems for two main reasons. First, it is not trivial to know when a line is perceptually dark or bright from color information alone (NAYATANI, 1997; NAYATANI; SAKAI, 2008). Second, a line may not be completely darker or brighter than the surroundings. For instance, consider the scenario of Figure 5.1, which corresponds to a gray line surrounded by a mixed black and white background.

Figures 5.2 (a) and (b) present respectively the **binary** line response ($L(\mathbf{p})$, see Equation 4.5) and the associated local density estimates ($m(\mathbf{p})$) for bright lines, while Figures 5.2 (c) and (d) present the same information for dark lines. In Figure 5.2 (b) and (d) the brighter the pixel, the higher the density estimate. In Figures 5.2 (a) and (c), the white pixels are those considered as line while the black pixels are those that were classified as background. The results were obtained using the WLD.

Figure 5.1: An ideal line with half-bright and half-dark surroundings.

Figure 5.2: The WLD bright binary response (a) and its local density estimates (b) for $K_u$, followed by the dark binary response (c) and the corresponding local density estimates (d) using the same kernel. For the density estimates, the brighter the pixel the higher the density estimate

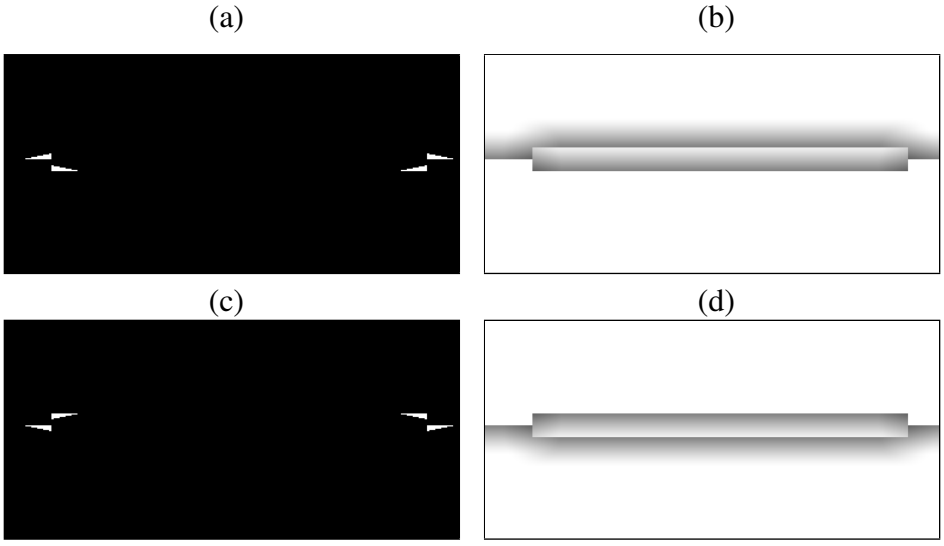
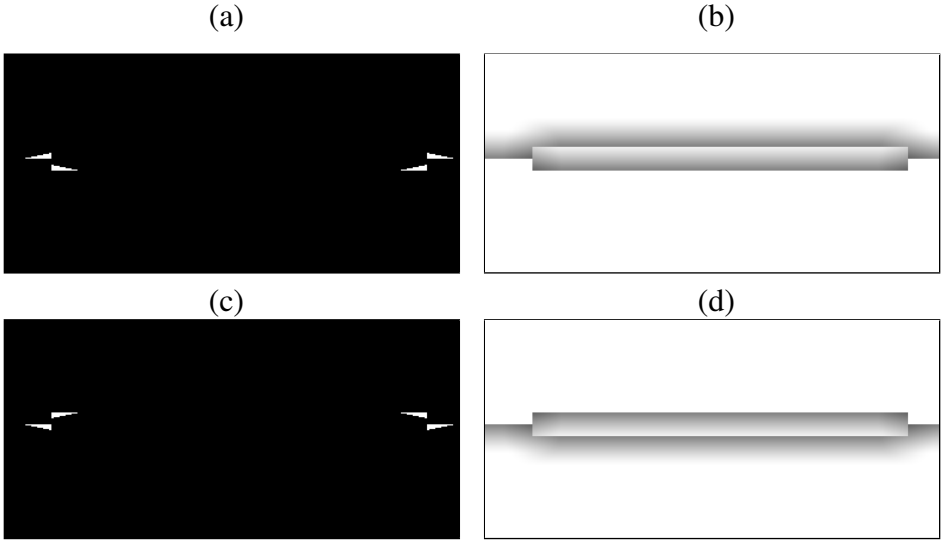


Figure 5.3: The (a) binary response for bright lines using the CWLD, along with (b) the corresponding local density estimates, followed by (c) the binary response for dark line detection and (d) its local density estimates. For the density estimates, the brighter the pixel the higher the density estimate




The WLD fails to detect most of the wide line pixels. This happens because the local density estimates augment excessively, leading the pixel associated to be wrongly considered as background and, consequently, to detection failure. In Figures 5.2 (b) and (d), there are white pixels in region associated to the line that have high density estimates and, therefore, have been classified as background. Figure 5.1 shows the same experiment using CWLD with $K_u$ and CIE 1976 color-difference. Observe that the results present the same WLD problem. Moreover, a side effect of this approach is that the density estimate becomes too unreliable to map the density estimate to any other measurement.

To deal with these problems we change the Equation 5.1 to

$$s(\mathbf{p}_0, \mathbf{p}) = \operatorname{sech}\left[\left(\frac{\Delta E(\mathbf{p}_0, \mathbf{p})}{t}\right)^5\right] \tag{5.3}$$

disregarding the dark and bright separation strategies represented by the conditions in the Equation. The results obtained for the CWLD are shown in Figures 5.4 (a) and (b), which illustrate the binary line response and the local density estimates respectively. Disregarding also this condition in Equation 4.9 we obtain the same results for the WLD in gray scale (see Figures 5.4 (c) and (d)). This strategy successfully detects all line pixels without corrupting density estimate information. However, some false positives can be seen in Figures 5.1 (a) and (c), next to the line endings, because this line is not only surrounded by brighter or darker pixels.

False positives are always generated in regions where the surrounding pixels have small density distribution, including the background pixels next to the line endings in the example above or curve surroundings. The false positives caused by high curvature regions can be removed enlarging the kernel only if the line is completely surrounded by background pixels of the same color. This enlargement augments the contribution of background pixels in the density computation eliminating this type of false positive, as we can see in Figure 5.5

Figure 5.4: The binary response of the CWLD without bright or dark filters (a) along with its local density estimate (b). The same information is provided for the WLD in (c) and (d). For the density estimates, the brighter the pixel the higher the density estimate

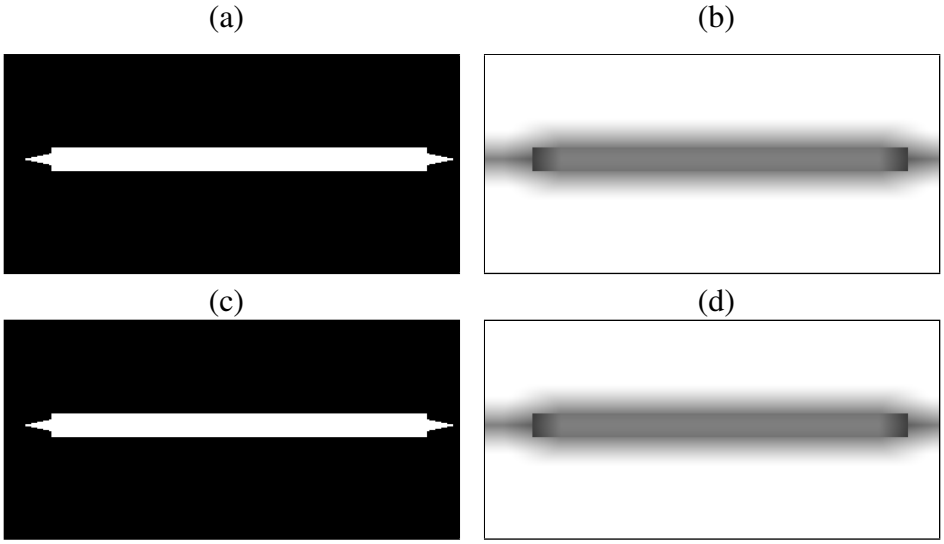



Figure 5.5: A curve with homogeneous background (a), followed by the CWLD responses using kernels with radius of $75$ (b), $78$ (c) and $80$ pixels (d).

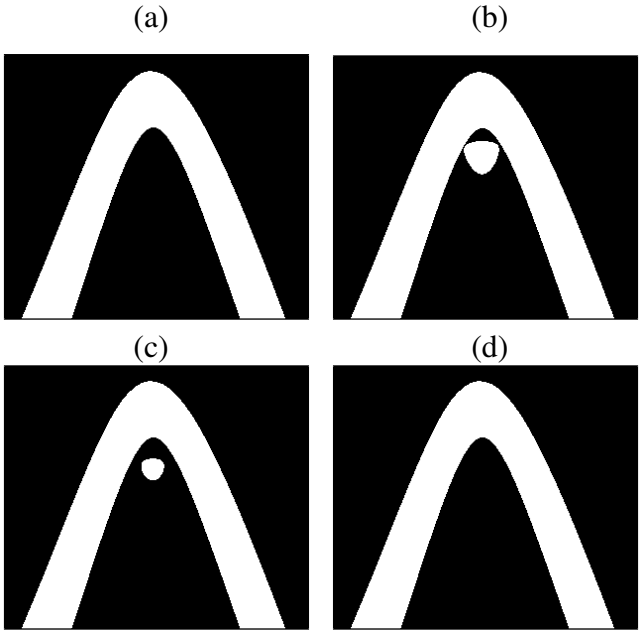


Figure 5.6: The binary line response for the line filtering specific undesired colors.

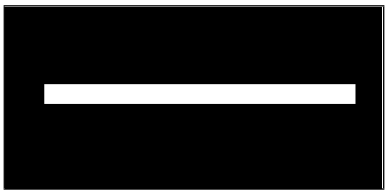


Figure 5.5 (a) shows a curve inside a region with homogeneous background in a binary image. Figures 5.5 (b) and (c) show the result of the CWLD using a uniform kernel of radius of 75 and 78 pixels, respectively. In these figures we observe the presence of false positives that diminish as the kernel radius augments. In Figure 5.5 (d), the kernel radius is equal to 80 pixels and large enough to remove all the false positive pixels. When the line is not surrounded by background pixels of the same color, we can use color information to cut out undesired line responses according to application requirements. For instance, in Figure 5.1 simply removing black and white pixels responses solve the problem, as we can see in Figure 5.6.

## 5.2 Monotonically increasing kernels

Section 4 showed that the kernel radius limits the lines that can be detected. The maximum line width $w$ and kernel radius $r$ are related by the following inequations: $w < 0.8r$ for $K_u$ and $w < 0.75r$ for $K_g$. By definition, values of $w$ and $r$ that violate these inequations imply $V_\lambda > V_K/2$, i.e., the line pixel may be erroneously classified as background. This also means that to detect wide lines we need kernels with big radius.

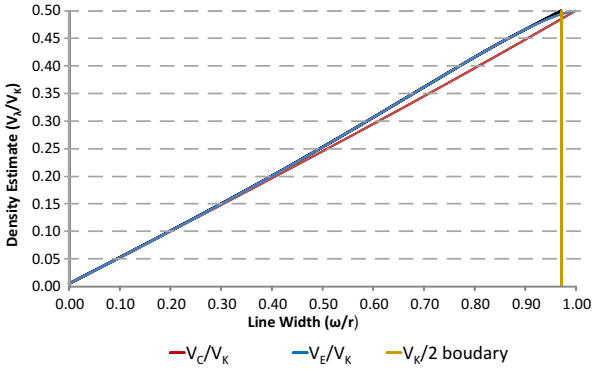
A large kernel may offer some disadvantages that must be taken into consideration. First, it implies in high computational cost, since the kernel considers more pixels in the calculations. Second, a large kernel receives more influence from outlier pixels, which can affect the filter response. For instance, the filter might fail to detect two lines that are too close to each other, because the distance between them is smaller than the kernel radius.

To minimize these problems, we propose a different kernel $\mathcal{K}(.)$ that is a combination of an uniform kernel with a monotonically increasing kernel based on the complement of the oblate Ellipsoid. It detects thicker lines than previous kernels with the same size. We observed that for $K_u$, the height of each kernel sample is always the same, therefore the only factor that influences the line response is its area inside the kernel (see Section 4). In monotonically decreasing kernels, e.g. $K_g$, the height of the kernel sample varies along the kernel, increasing the differences between the responses associated to $V_C$ and $V_E$. This can be seen comparing Figures 4.2 (a) and (b). Our new kernel, $\mathcal{K}(d)$, uses its monotonic increasing characteristic to compensate the decay of density as the kernel moves from the center to the edge of the line. It is defined by

$$\mathcal{K}(d) = \begin{cases} r\left(a + b\left(1 - \sqrt{1 - \frac{d^2}{r^2}}\right)\right) & \text{if } d \leq r \\ 0 & \text{otherwise} \end{cases} \tag{5.4}$$

where $a$ and $b$ are respectively the height of the uniform cylinder and the length of the semi-axis of the oblate ellipsoid. These values are chosen to obtain the same result for $V_C$

Figure 5.7: The relationship of $w$ (as the rate of $r$) and $V_\lambda$ (as the rate of $V_K$) using $\mathcal{K}$ and Equations 4.7 and 4.8. Note that for $\mathcal{K}$, $V_C$ and $V_E$ are plotted together with the maximum $V_\lambda(w, r, \Delta w)$ at each width value.




and $V_E$ for a $w$ (Equations 4.7 and 4.8, respectively), when $V_\lambda$ is maximum ($V_\lambda = V_K/2$). This will force the two equations to get closer to each other again, the opposite behavior of $K_u$ and $K_g$. This is equivalent to

$$\int_0^w \int_{-\sqrt{r^2-x^2}}^{\sqrt{r^2-x^2}} \mathcal{K}(||(x,y)||)dxdy = \int_{-\frac{w}{2}}^{\frac{w}{2}} \int_{-\sqrt{r^2-x^2}}^{\sqrt{r^2-x^2}} \mathcal{K}(||(x,y)||)dxdy$$

Solving this Equation, we see that $\frac{a}{b} \approx 0.14681$, resulting in Figure 5.2, which present $V_C = V_E = V_K/2$ when $w = r$. It means that the same kernel can detect thicker lines. Still, as the kernel is no longer monotonically decreasing, we cannot state that $V_E \leq V_\lambda \leq V_C$. Thus, we must obtain the maximum and minimum $V_\lambda$ at each width, moving the line position from the kernel center to the kernel border. The maximum, $V_{\lambda_{max}}$, and minimum, $V_{\lambda_{min}}$, values for $V_\lambda$ are obtained using Equation

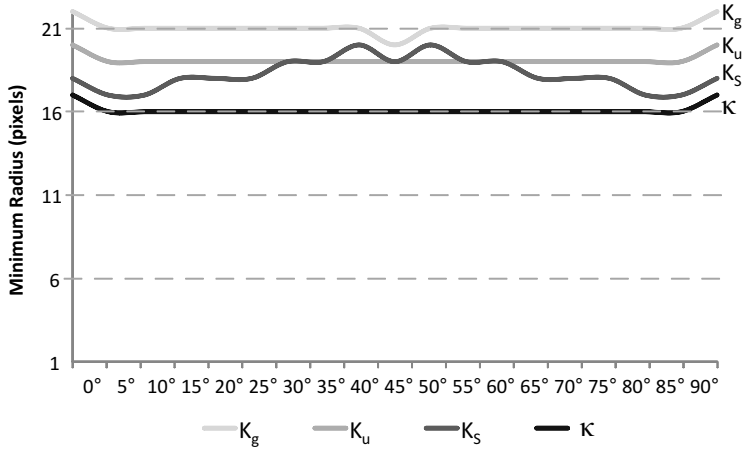
$$V_\lambda(w, r, \Delta w) = \int_{-\frac{w}{2}+\Delta w}^{\frac{w}{2}+\Delta w} \int_{-\sqrt{r^2-x^2}}^{\sqrt{r^2-x^2}} \mathcal{K}(||(x,y)||)dxdy \tag{5.5}$$

and varying the displacement between the centers of the line and the kernel $\Delta w$ considering that $0 \leq w \leq r$ and $|\Delta w| \leq w/2$. Figure 5.2 illustrates that when $w \approx r$, i.e., $w/r \approx 0.97$ – theoretically.

Table 5.1 shows the minimum radius required to detect a horizontal bar shaped line with width of 16 pixels using $K_u$, $K_g$, both defined in Chapter 4, $\mathcal{K}$ and a Gaussian kernel approximation with square mask, $K_s$, of size $2r$+1 used by Liu (LIU; ZHANG; YOU, 2007).

$$K_s(d) = \exp\left(\frac{-d^2}{2r^2}\right)$$

Figure 5.8: The minimum radius required to detect a bar shaped line rotated by several degrees.




and computed for all pixels inside the mask considering the distance, $d$, from the pixel in the center of the mask, $\mathbf{p_0}$, to every pixel inside it.

Table 5.1: Minimum Kernel Radius

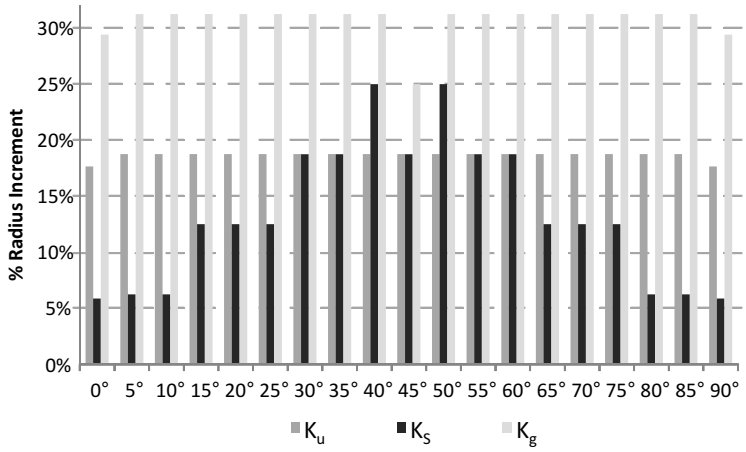
| **Kernel** | **Minimum Radius** |
|---|---|
| $\mathcal{K}$ | 17 pixels |
| $K_s$ | 18 pixels |
| $K_u$ | 20 pixels |
| $K_g$ | 22 pixels |

Initially, comparing only $K_u$ and $K_g$, we can see that $K_u$ detects the line with a smaller radius than $K_g$, which is in accordance to Chapter 4. Whereas $K_s$ needed a smaller kernel than $K_g$ and $K_u$, since it uses a larger area to compute its density distribution (squared vs circular areas). On the other hand, $\mathcal{K}$ has the best results than the others, but similar to $K_s$. That may suggest that there is no significant difference between $\mathcal{K}$ and $K_s$ for detecting wide lines. However, if we consider the same line rotated by several degrees, we note that this conclusion is not consistent in every case. Figure 5.2 shows the minimum kernel radius needed to detect a line under several rotations using all kernels.

From these results, $K_s$ does not outperform $K_u$ when the line is rotated between $30°$ and $60°$. Besides, the kernel radius required to detect the line using $K_s$ oscillated considerably as the line was rotated, making the required radius dependent on the rotating angle. This happened because the density distribution is affected by the shape of the kernel used in the calculations. $\mathcal{K}$ presented more stable results across the rotating angles. Figure 5.9 shows the rate increase in the radius using $K_u$, $K_g$ and $K_s$ compared to $\mathcal{K}$ for the results in Figure 5.2.

Observe that the radius increment can be in the range from $5\%$ to $30\%$ , moreover, an augment in the radius implies in quadratic augment of the amount of pixels analyzed. That makes $\mathcal{K}$ more efficient computationally in this experiment when compared to the other kernels. In Section 6 we perform more experiments to validate our kernel.

Figure 5.9: Rate increase required to make $K_u$, $K_g$ and $K_s$ to detect all line pixels, when compared to $\mathcal{K}$.




## 5.3 Line width estimation

Liu (LIU; ZHANG; YOU, 2007) only determined what was the required radius to detect the thickest line using monotonically decreasing kernels. Here, our goal is to construct an univocal relationship between density estimate and line width using the theory behind the WLD and CWLD. We start presenting this relationship in a slightly different way than Chapter 4 and 5.2 in Figure 5.10 for $K_u$, $K_g$ and $\mathcal{K}$, inverting the graph orientation, that is, width as a function of density estimate. It shows that none of the kernels provides a unique relationship between width and density estimate, since it depends on $\Delta w$, i.e., the displacement between the center line and the center of the kernel. Considering $K_u$ and $K_g$. this relationship results in larger ranges of $w$ for each $V_\lambda$ insofar as $V_C$ and $V_E$ approach $V_K/2$. To diminish this range we can select a unique curve, $V_{\lambda_{fit}}(w, r)$, that maps univocally each $V_\lambda$ to a single $w$ estimate. As Liu (LIU; ZHANG; YOU, 2007) proved that for $K_u$ and $K_g$, $V_E \leq m(\mathbf{p}) \leq V_C$, we have chosen the curve that provides equal amount of errors for the cases of $V_C$ and $V_E$, that is

$$V_{\lambda_{fit}}(w, r) = \frac{V_C(w, r) + V_E(w, r)}{2} \tag{5.6}$$

As $\mathcal{K}$ is a monotonically increasing kernel, $m(\mathbf{p})$ values are not always between $V_C$ and $V_E$. Therefore, Equation 5.6 has to be modified to accommodate the maximum and minimum values of width considering $\mathcal{K}$. Figure 5.10 (c) shows these two additional functions considering Equation 5.5 and the different values of $\Delta w$. Even for these values, we can see that the range of $w$ in $\mathcal{K}$ is still small, resulting in width estimates that are more precise for wide lines than the ones obtained with $K_u$ and $K_g$. This is another advantage of choosing to approximate $V_C$ to $V_E$ when $V_\lambda = V_K/2$. Now we modify Equation 5.6 to

$$V_{\lambda_{fit}}(w, r) = \frac{V_{\lambda_{max}}(w, r) + V_{\lambda_{min}}(w, r)}{2}. \tag{5.7}$$

where

$$V_{\lambda_{max}}(w, r) = \max_{\Delta w}(V_\lambda(w, r, \Delta w)) \tag{5.8}$$

Figure 5.10: The relationship of $w$ (as the ratio of $r$, i.e. $w/r$) and $V_\lambda$ (as the ratio of $V_K$, i.e. $V_\lambda/V_K$), for $K_u$, $K_g$ and $\mathcal{K}$. Note that for $\mathcal{K}$, $V_C$ and $V_E$ are plotted together with the minimum and maximum $V_\lambda(w, \Delta w)$ at each width value.

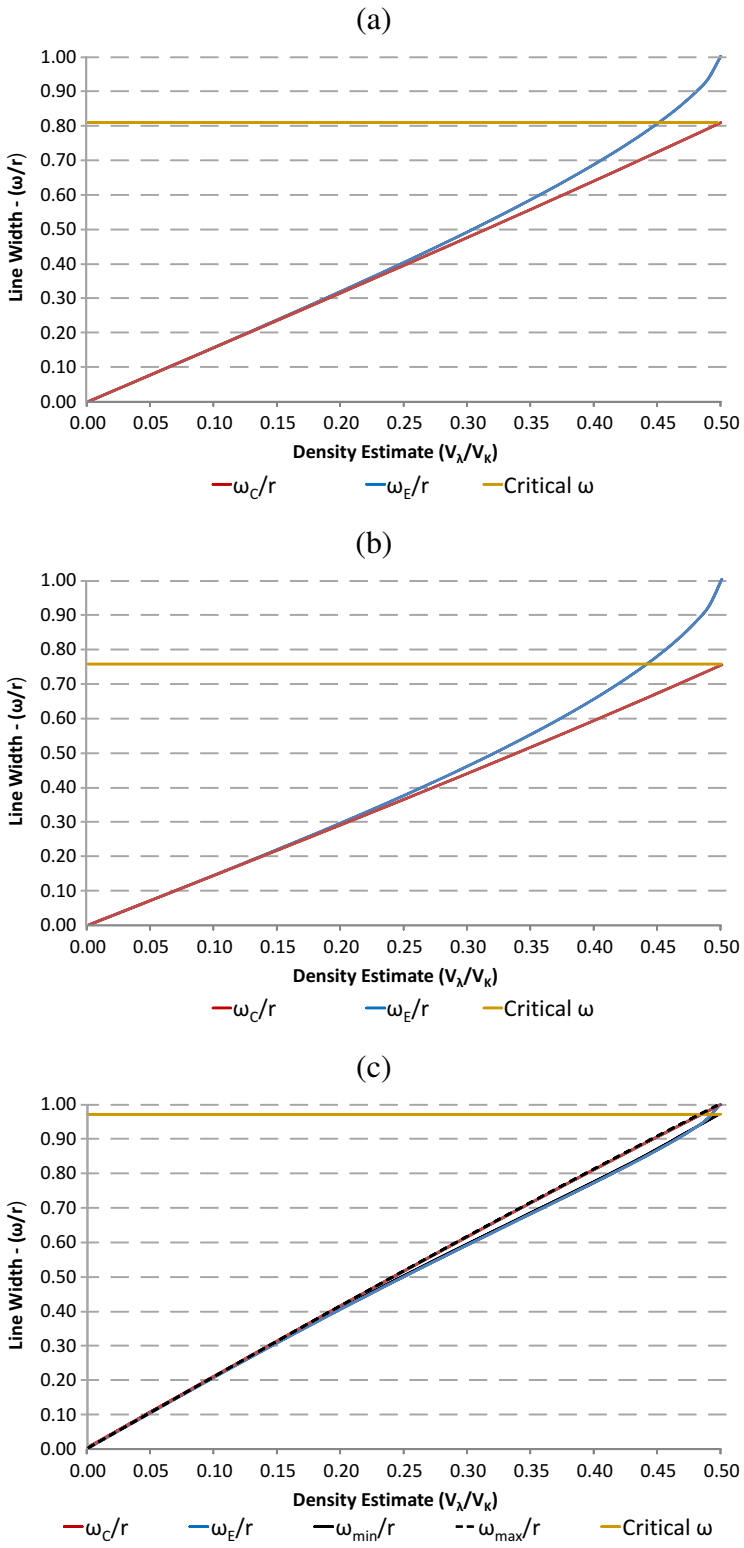


Figure 5.11: An ideal horizontal line (a) and its width estimates represented as intensities using $K_u$ is shown in (b), where the wider the thickness measured, the higher the pixel intensity. Note in (b) the normalized width estimates decay at the extremities of the line.

(a) (b)

Table 5.2: Ideal Line Width Measurement Analysis (actual width = 16 pixels)

| **Kernel** | **Minimum Radius** | **Mean Width Estimate** | **MSE** |
|---|---|---|---|
| $K_s$ | 18 pixels | 13.87 pixels | 0.0242 |
| $K_g$ | 22 pixels | 16.18 pixels | 0.0105 |
| $K_u$ | 20 pixels | 16.09 pixels | 0.0087 |
| $\mathcal{K}$ | 17 pixels | 15.61 pixels | 0.0067 |

and

$$V_{\lambda_{min}}(w, r) = \min_{\Delta w}(V_\lambda(w, r, \Delta w)) \tag{5.9}$$

To obtain the pixel wise line width from the estimated volume occupied by the line, $V_\lambda$, we construct the function

$$w(\mathbf{p}) = f(V_\lambda) \tag{5.10}$$

from data obtained numerically.

For further precision for the width estimates applied to natural images, we decided to calculate the density estimate through a region growing algorithm. We use as seed the central pixel color. If the pixel under analysis has a similar color to the central pixel, i.e., the color-difference is below a threshold, the region is allowed to grow in that direction. This way, we reduce undesired density estimate interference. That is, lines too close to each other, or even unconnected regions with similar colors are no longer considered in the density estimate. This strategy can be expensive, since it performs the region grow for every pixel of the line in a region of specific kernel radius. Thus, it is recommended for natural images with strong interference problems.

Now consider again a bar shaped line with width of 16 pixels as shown in Figure 5.11 (a). Figure 5.11 (b) shows the local width estimates, represented as intensities, computed at each pixel using $K_u$ for the line in 5.11 (a). Table 5.2 shows the mean estimate and the mean square error (MSE) associated with the local width estimate when using each kernel with the smallest radius, computed previously, that can detect all line pixels.

The mean width estimate and MSE using $K_s$ did not show good results when compared to the other kernels. This happened because the density estimate of $K_s$ is computed considering the entire square region of the kernel, instead of a circular region. Therefore, the mean width estimate obtained using $K_s$ underestimates the influence of the additional pixels of the square region of the kernel, taking the mean to a considerably smaller value than the actual line width. In the same test, $K_u$ outperformed $K_g$, what was also expected since for $K_u$ the difference between $V_C$ and $V_E$ is smaller than $K_g$ (compare Figures 5.10 (a) and (b)), resulting in smaller error in the width estimates for $K_u$. Finally, the proposed kernel $\mathcal{K}$ presented the best MSE, but not the closer mean estimate.

Table 5.3: Ideal Line Width Measurement Analysis without Edge Pixels

| **Kernel** | **Minimum Radius** | **Mean Width Estimate** | **MSE** |
|---|---|---|---|
| $K_s$ | 18 pixels | 14.13 pixels | 0.0152 |
| $K_g$ | 22 pixels | 16.52 pixels | 0.0041 |
| $K_u$ | 20 pixels | 16.42 pixels | 0.0023 |
| $\mathcal{K}$ | 17 pixels | 15.88 pixels | 0.0004 |

These results led to further investigation of the width estimate. Observe that there are two anomalies in Figure 5.11 (b), represented by a smooth intensity decay pattern towards each of the extremities of the line. Such pattern is independent of the width of the line and it appears for all the kernels considered so far. This type of anomaly is caused by the behavior of the local density estimates in these regions. Figure 5.12 presents the behavior of the density estimates at a distance $d$ from one of the edges of the line for $K_u$, $K_g$ and $\mathcal{K}$ considering $V_C$ and $V_E$. Each kernel uses the minimum radius required to detect the wide line completely (see Table 5.2). The distance $d$ is in the range, $r \geq d \geq 0$, where $r$ is the kernel radius and position 0 corresponds to the limit between line and background. For the density calculation, the kernel center is displaced along the center line, for $V_C$ and along the edge of the line, for $V_E$.

From Figure 5.12 we see that the density estimate decreases despite the fact that the width of the line remains the same. Insofar as the kernel approaches the end of the line, the presence of background pixels augments, leading to density estimation errors, what ultimately results in width estimation errors. Therefore, such width estimates tend to be less reliable for lines with small length (similar to the diameter of the kernel). Since we now know that the pixels in these regions do not perform well in the width estimation, we can perform the same width measurement experiment for the ideal line neglecting the anomalous pixels at the extremities of the line. Such results are presented in Table 5.3.

Table 5.3 shows that the removal of such pixels with the previously mentioned anomaly from the width measurement calculation changed the performance for all the kernels when compared to Table 5.2. Also, now $\mathcal{K}$ has better performance than all the other kernels, followed by $K_u$, $K_g$ and $K_s$ respectively.

Now we study the same line at several different rotations. Figure 5.13 presents width estimation results for the bar shaped line when rotated by several degrees of rotation. Responses for the angles from $90^\circ$ to $180^\circ$ are supposed to be symmetric responses to the interval we have tested here. As the line is rotated, it has to be smoothed to display accurate edges. The presence of smoothed edge pixels, due to aliasing problems, adds additional error to the measurement. The amount of edge pixels is higher near $45^\circ$, what justifies the errors for all kernels shown in Figure 5.13.

Figures 5.13 shows that $K_s$ performance varies considerably across the rotation angles. This is a consequence of using the entire square mask in the density estimate calculation, that is, the same line presents different density estimates according to the line orientation inside the kernel. As a result, the width measurement becomes dependent on the orientation of the line. For this same test, $K_u$ and $K_g$ provided better, or at least as good as, results than $\mathcal{K}$ because to detect the line completely, their radius had to be increased, leading to smaller error between $V_C$ and $V_E$. This can be understood by looking

Figure 5.12: The density estimate at the edge and center at positions next to the transition between line and background for $K_u$ (a), $K_g$ (b) and $\mathcal{K}$ (c). The transition between line and background occurs at position 1 in the end of the chart.

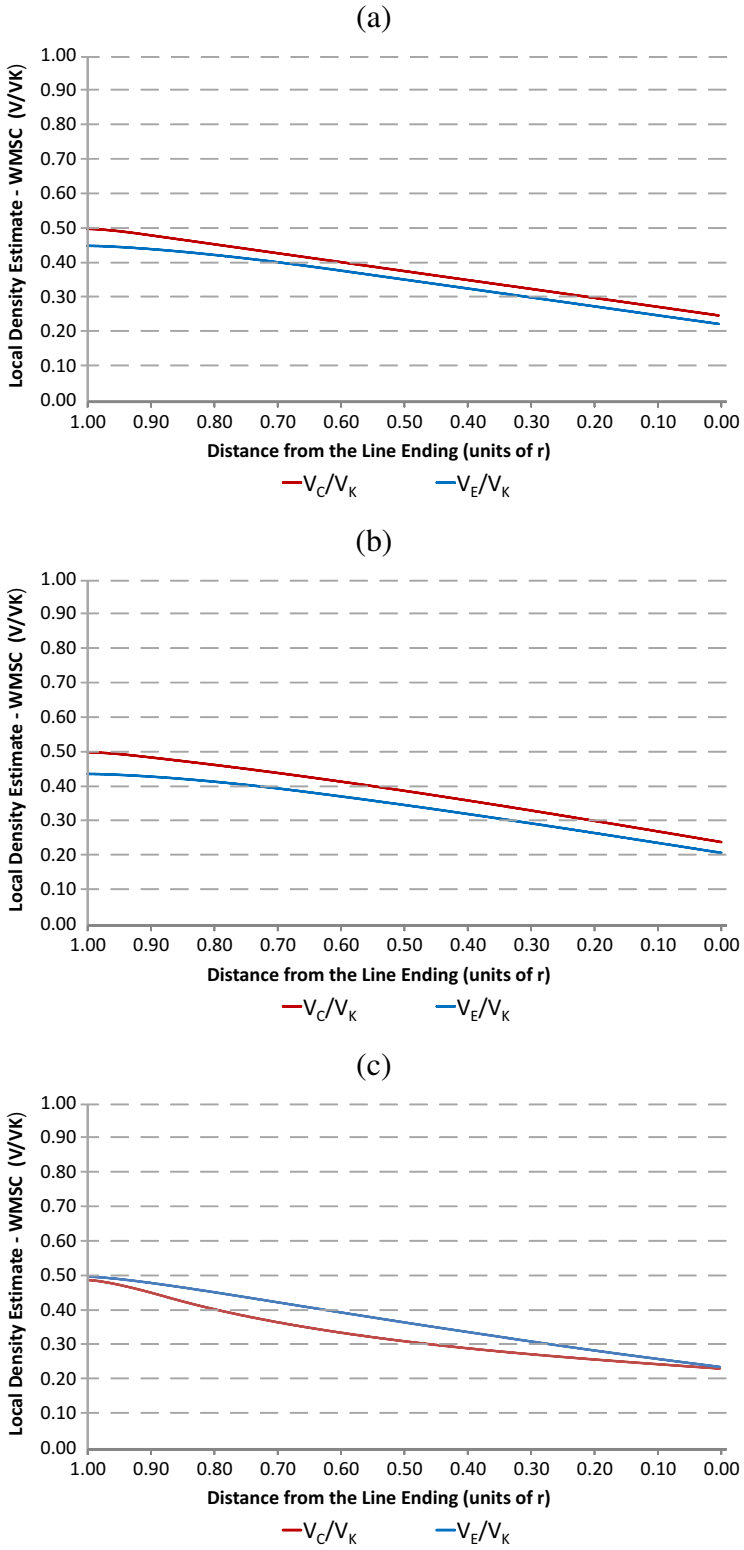
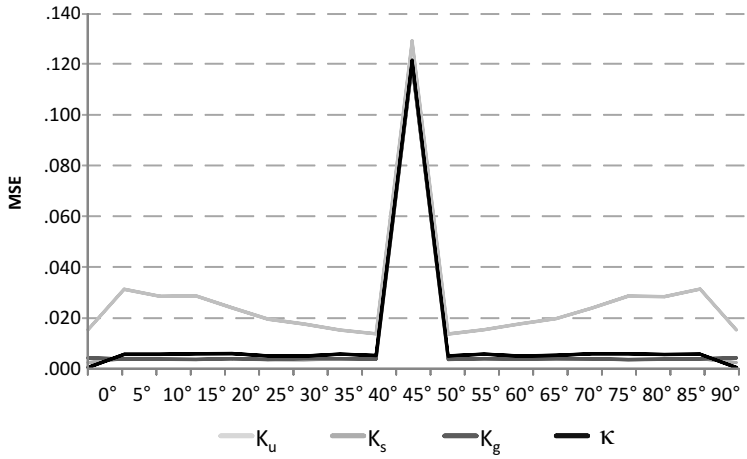


Figure 5.13: Width estimates obtained using all the kernels for an ideal bar shaped line of known width rotated by several degrees.

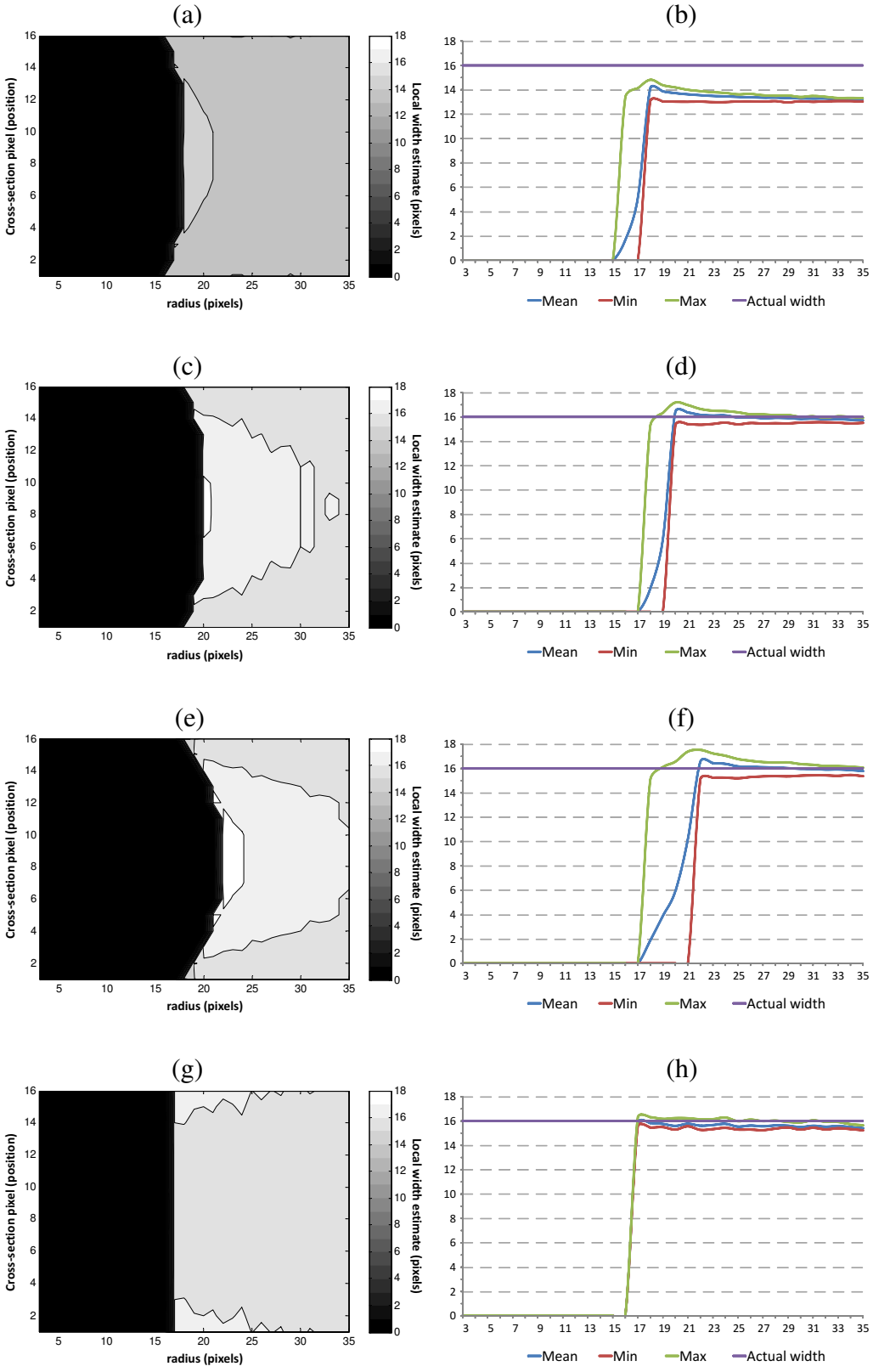



at Figures 5.10 (a) and (b). Insofar the kernel radius gets bigger than the line width, the difference between the available measurements becomes smaller, making the error associated to the width estimate smaller, at the expense of increased computation cost for these kernels. At this time $\mathcal{K}$ did not provide the most accurate measurements when compared to the other two kernels. This can be explained by Figure 5.10 (c), where the difference between maximum and minimum estimates slightly augments, before starting to decrease again.

Further, Figures 5.3 (a), (c), (e) and (g) present the contour map with the width estimates on the same cross-section of pixels using different kernel radii for $K_s$, $K_u$, $K_g$ and $\mathcal{K}$, respectively. For this test we have selected the cross-section of an ideal horizontal line with width of 16 pixels at the central position along the line length, to avoid the previously mentioned density estimate decay at the edge regions. Figures 5.3 (b), (d), (f) and (h) present the mean, maximum and minimum width values computed using $K_s$, $K_u$, $K_g$ and $\mathcal{K}$ at the cross-section.

Figure 5.3 provides a clearer picture of the performance of the width estimate with all the kernels. It shows how the line detection behaves as the kernel radius gets bigger. When the width is 0, it means that the algorithm classified the pixel as a background pixel. We can see that $\mathcal{K}$ detects all pixels from the cross-section as line pixels using a smaller kernel radius, 17 pixels, than all the other kernels. In addition, with smaller radii the width estimates for $K_u$ and $K_g$ are less consistent or void when compared to $\mathcal{K}$. These two observations imply that the computational cost of $\mathcal{K}$ is considerably smaller than those of $K_u$ and $K_g$. Moreover, $K_s$ minima and maxima are bounded, but clearly below the actual ideal line width. One could argue that a straightforward solution would be to scale the estimates of $K_s$ to solve this problem. However, as we have already shown, $K_s$ estimates are not isotropic, that is, such solution would have to consider the line orientation in the computation. In a nutshell, we can see that $K_u$, $K_g$ and $\mathcal{K}$ can extract local width information for the ideal line case. To the best knowledge of the author, this is the first time that width is extracted from the density distribution, computed using a kernel.

Figure 5.14: Contour map obtained when applying the CWLD algorithm to each pixel of the same cross-section of a line with width 16 pixels, using $K_s$ (a), $K_u$ (c), $K_g$ (e) and $\mathcal{K}$ (g), but with varying kernel radius. The analysis of mean, maximum and minimum width values is also provided in $K_s$ (b), $K_u$ (d), $K_g$ (f) and $\mathcal{K}$ (h) in the same conditions.



# 6 EXPERIMENTS

In this Chapter, we present several results using natural images. We start defining the ground truth and the statistical error measures used to evaluate the CWLD. We perform several tests using the CWLD in the images, varying color-differences and kernels, to compare the effectiveness to detect lines, and also to determine the CWLD performance to measure width.

## 6.1 The ground truth

Defining which pixels belong to a given line is difficult in natural images, because this depends on the edge interpretation and the color perception by the human eye, which varies for each subject. As we have chosen to use perceptually uniform color spaces in the CWLD, the ground truth construction starts with the manual selection of the line pixels in the natural images to be tested. This results in the binary images shown in Figures 6.1 (d)-(f), obtained from (a)-(c), respectively. We use it to assess the capacity of the CWLD to detect line pixels. Here, we compute the ratio of pixels that the CWLD marks accurately as line pixels when compared with the amount of line pixels of the ground truth. We also compute this ratio for the WLD and compare it to the one obtained using the CWLD, considering $K_u$, $K_g$, $K_s$ and $\mathcal{K}$ with varying widths. These results are shown in Sections 6.3 and 6.4.

We do not count false detections at this time, since the definition of what is not a curvilinear feature can be somewhat subjective. For instance, the background of natural images sometimes presents curvilinear features that are not of interest. Besides, the WLD and CWLD require robust false detection mechanisms. We discuss that in the conclusions and future work Chapter (see Chapter 7).

Afterwards, we perform the thinning of the binary images (GUO; HALL, 1989) to obtain the approximation of the center line, that for now on we call only **center line**. Figures 6.1 (g)-(i) show the centerline of binary images in (d)-(f), respectively. Then, to each point in the center line, we determine the nearest pixel in each border of the binary line and calculate the Euclidean distance between them. That is, using the pixels of the center line we obtain the normal vectors, preceded by the derivatives, at every pixel of the center line. Then, we use these vectors together with the binary image to obtain the boundary pixels. Next, we calculate the Euclidean distance between the two border pixels of the line (obtained from (d)-(f)). Since we calculate the distance between the center of the two edge pixels, we add one unit to this distance to adjust the half pixel missing at each border, something between $1$ and $\sqrt{2}$ pixels. We correct the thinning errors manually and select a line that is used for comparative purposes, marking some reference points, as shown in (j)-(l). The distance already calculated for the pixels of the selected line is the

ground truth width used in the width estimation experiments. In this work we perform the comparison between width estimation produced by the CWLD and WLD and ground truth width only at the pixels of the center line. This aims to facilitate the visualization and analysis of results.

## 6.2 Error Analysis

The metrics used in this work are based on distribution-based similarity measures and traditional error measures. Given a set of ground truth widths $\mathbf{G} = \{g_i\}_{i=1}^{n}$ with expected mean value $\mu_g$ and standard deviation $\sigma_g$ and, the a set of width estimates $\mathbf{W} = \{w_i\}_{i=1}^{n}$ generated by the WLD/CWLD, with expected mean value $\mu_w$ and $\sigma_w$, the correlation, $\rho(\mathbf{G}, \mathbf{W})$, is defined as

$$\rho(\mathbf{G}, \mathbf{W}) = \frac{E\left[(\mathbf{G} - \mu_g)(\mathbf{W} - \mu_w)\right]}{\sigma_g \sigma_w} \tag{6.1}$$

The absolute mean error, $\xi_{abs}$, is given by:

$$\xi_{abs} = \frac{1}{n} \sum_{i=1}^{n} |w_i - g_i| \tag{6.2}$$

Using the same information, we obtain the Mean Square Error between $\mathbf{G}$ and $\mathbf{W}$

$$MSE(\mathbf{G}, \mathbf{W}) = \frac{1}{n} \sum_{i=1}^{n} (w_i - g_i)^2 \tag{6.3}$$

then, the corresponding Root Mean Square Deviation

$$RMSE(\mathbf{G}, \mathbf{W}) = \sqrt{MSE(\mathbf{G}, \mathbf{W})} \tag{6.4}$$

and the Normalized Root Mean Square Deviation

$$N_{RMSE}(\mathbf{G}, \mathbf{W}) = \frac{RMSE(\mathbf{G}, \mathbf{W})}{w_{max} - w_{min}} \tag{6.5}$$

where $w_{max} - w_{min}$ define the range of observed values.

Regarding the distribution-based similarity measures, we assume that sets $\mathbf{G}$ and $\mathbf{W}$ have Probability Distribution Functions, $P_{\mathbf{G}}$ and $P_{\mathbf{W}}$ respectively. Then, we can calculate the Bhattacharrya Coefficient, BC($\mathbf{G}$, $\mathbf{W}$), through

$$BC(\mathbf{G}, \mathbf{W}) = \sum_{r \in X} \sqrt{P_{\mathbf{G}}(r) P_{\mathbf{W}}(r)} \tag{6.6}$$

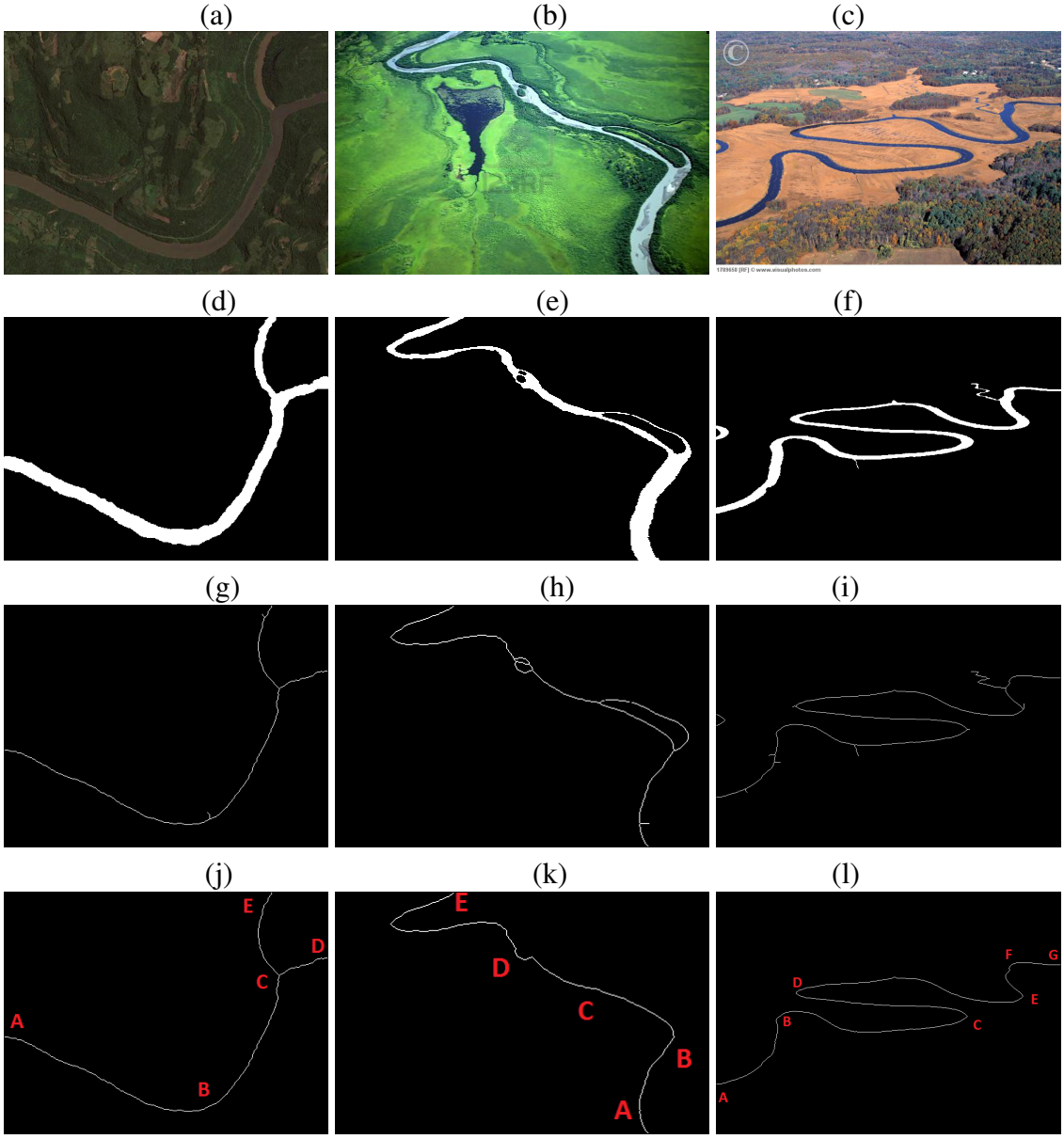
where X is the domain that represents all possible widths. Insofar as $BC(\mathbf{G}, \mathbf{W}) \rightarrow 1$, the more similar the sets $P_{\mathbf{G}}$ and $P_{\mathbf{W}}$ are. $BC(\mathbf{G}, \mathbf{W})$ enables the computation of the Hellinger Distance

$$HD(\mathbf{G}, W) = \sqrt{1 - BC(\mathbf{G}, \mathbf{W})} \tag{6.7}$$

and also, the Bhattacharrya Distance, defined by

$$BD(\mathbf{G}, W) = -\ln BC(\mathbf{G}, \mathbf{W}) \tag{6.8}$$

Figure 6.1: Our ground truth width is obtained from images (a)-(c). First, a binary image containing some of the lines is generated by marking the line pixels manually((d)-(f)). After, the thinning method is applied to (d)-(f) to obtain the center line shown in (g)-(i). Then, the ground truth width is obtained using the border information from (a)-(c) and the center line information from (g)-(i). Finally, we correct thinning errors manually and select a line from the image, marking some reference points for analysis (j)-(l).



where, the closer the measure is from zero, the closer the two distributions are[1].

Furthermore, we provide the median and also the percentile at $75\%$ of the Cumulative Distribution Function of the error in pixels, $\xi$, between $\mathbf{G}$ and $\mathbf{W}$. The median is the error value, $M$, where probability that $\xi \leq M$ and $\xi \geq M$ are both $\approx 0.5$. Similarly, the percentile at 75% is the error value, $\tau$, where probability that $\xi \leq \tau$ is $\approx 0.75$, and the probability that $\xi \geq \tau$ is $\approx 0.25$. These statistics give an overview of the trend of the magnitude of the error distribution from 50% to 75% of the distribution.

We also used the Kolgomorov-Smirnov Hypotheses Test, $KS$. It performs a non-parametric goodness-of-fit analysis using the differences between the cumulative probability distributions of the ground truth, $C_{\mathbf{G}}$, and the estimated widths provided by the CWLD (where the WLD is a subset of the CWLD), $C_{\mathbf{W}}$. Here, we are interested in the hypothesis $H_0$, i.e., the probability that the two distributions are the same. If $H_0$ is rejected, we accept $H_1$. It means that the two distributions are different. $H_0$ is rejected or accepted using the significance of $5\%$, $1\%$ or $0.1\%$. This involves the number of estimates, n (the same for $\mathbf{G}$ and $\mathbf{W}$ in our case), and the same domain $\mathbf{X}$, defined previously. The criteria used to accept $H_0$ is

$$KS = D\sqrt{\frac{n}{2}} \leq K_\alpha \tag{6.9}$$

with

$$D = \sup_{r \in \mathbf{X}}(||C_{\mathbf{W}}(r) - C_{\mathbf{G}}(r)||) \tag{6.10}$$

where $K_\alpha$ depends on the desired significance and is obtained through the tables provided by Smirnov in his work (SMIRNOV, 1948). $H_0$ is accepted if $KS \leq K_\alpha$, otherwise $H_0$ is rejected and $H_1$ is accepted. This test shows if the ground truth and the CWLD width estimates can be considered similar or not. It is a very difficult test, because it rejects the hypothesis if there is a large difference between just one pair of bins. Therefore, if the results show the acceptance of $H_0$, it means that the distributions are similar and that the width estimates can be correct.

Finally, another metric used is the hit count of correct estimates

$$Hits(\mathbf{G}, \mathbf{W}) = \frac{1}{n}\sum_{i=1}^{n} \Psi(g_i, w_i). \tag{6.11}$$

where

$$\Psi(g_i, w_i) = \begin{cases} 1 & \text{, if } \frac{||g_i - w_i||}{g_i} \leq \frac{\phi}{100} \\ 0 & \text{, otherwise} \end{cases} \tag{6.12}$$

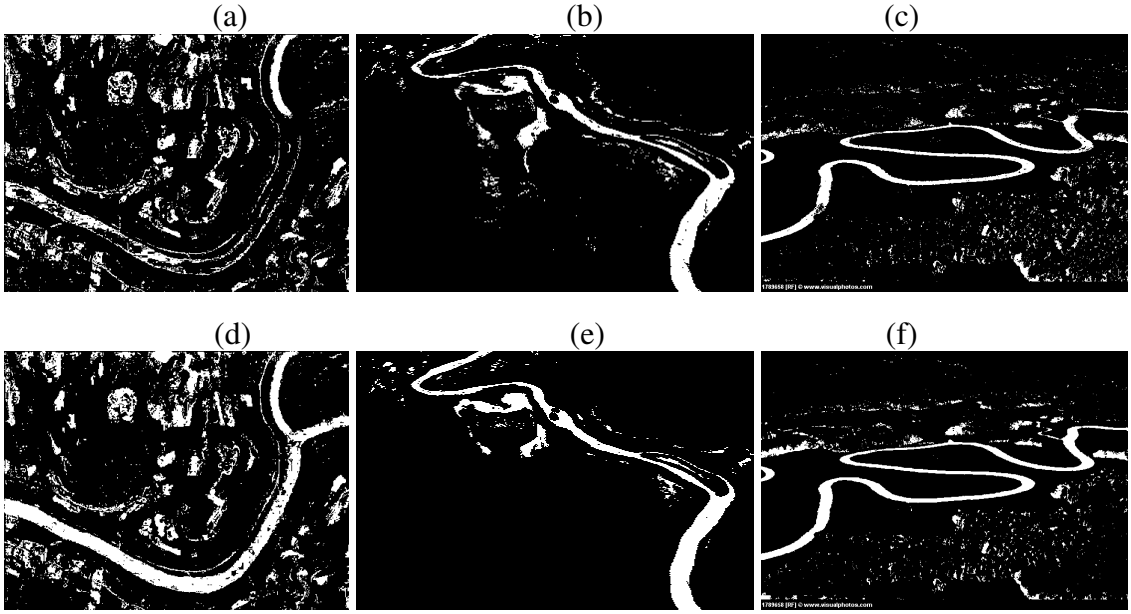
where $\phi$ can be $10\%$, $20\%$ or $30\%$. Where it should be noted that thin lines offer a real challenge to the CWLD, since at 10%, 20% or 30% the error must be smaller than a pixel.

## 6.3 Line Detection

In this section, we compare the CWLD and WLD using three natural images and their gray scale versions. In all experiments, we keep the same attributes for the WLD and

[1]When considering two distributions, care must be taken, since two similar distributions do not necessarily imply measurements in the same order. Still, this is an indication of how similar the two distributions of measurements are and their correlation.

Figure 6.2: Comparison of line detection using the WLD and CWLD. Figures (a)-(c) correspond to the detection using the WLD, while (d)-(f) correspond to the binary line detection using the CWLD, using images shown in Figure 6.1 (a)-(c).



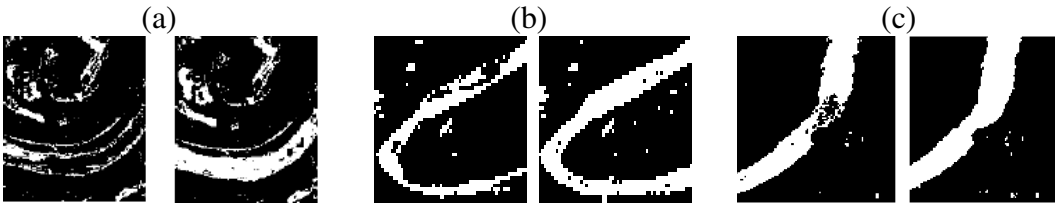

CWLD (kernel size, kernel type, etc.) for both methods, e.g., the kernel used is $K_s$ with a radius of 30 pixels. The main difference is the threshold used to determine if a pixel belongs or not to a line. The CWLD uses the standard deviation of the CIE 2000 color-difference as threshold (see Chapter 5), while the WLD uses the standard deviation of the gray scale intensities of the image (see Liu et. al (LIU; ZHANG; YOU, 2007)). No pre or post-processing is applied to the results, such as smoothing filters, morphologic reconstruction or binarization.

Figures 6.1 (a)-(c) are the tested natural images. Figures 6.2 (a)-(c) are the detection results using the WLD, while Figures 6.2 (d)-(f) correspond to the binary line detection using the CWLD. Comparing Figures 6.1 (d)-(f) (ground truth) and the individual CWLD detection results, we can see that they are remarkably similar.

Specifically, Figure 6.2 (a) shows that the WLD does not detect most of the line when compared to the result produced by the CWLD in Figure 6.2 (d). Careful analysis of this natural image reveals the level of noise, making the whole detection extremely difficult. In the same way, Figure 6.2 (e) shows that the CWLD not only detects more line pixels, but it also presents visibly less false positives than the WLD shown in Figure 6.2 (b). Figure 6.1 (c) is also challenging, because the colors of the linear features are not exactly the same. Still, the WLD presents more detection failures than the CWLD (compare Figures 6.2 (c) with 6.2 (f)). Overall, in this experiment, a qualitative analysis shows that the WLD has more detection failures than the CWLD.

Figure 6.3 shows the detailed views of the tests presented in Figure 6.2 for further inspection. The CWLD is visibly more effective to detect line pixels, i.e., color-difference did improve the effectiveness of line detection.

Figure 6.3: (a)-(c) present the detailed comparative views of the results from the WLD and CWLD, shown in Figure 6.2. For each Figure, the images on the left represent the WLD, while those on the right, the CWLD.




## 6.4 Qualitative Analysis

In this section, we verify if $\mathcal{K}$ can detect line pixels with smaller radius than the other kernels considering the same natural images from the previous experiment. At this time, we verify how the line pixel detection behaves for $K_u$, $K_s$, $K_g$ and $\mathcal{K}$, insofar we augment the kernel radius, using the same color-difference, CIE 2000. Basically, we compare the number of line pixels of the ground truth with those marked by the CWLD.

Figures 6.4 (a)-(c) correspond respectively to the rate of line pixel detection for Figures 6.1 (a)-(c). For each image, we show the results for $K_u$, $K_s$, $K_g$ and $\mathcal{K}$, increasing the kernel radius of 1 unit, successively. That is, we perform 37 passes of the algorithm for each image. Again, we do not apply pre or post-processing to the results, such as smoothing filters, morphologic reconstruction or binarization.

Observe that $\mathcal{K}$ enables the detection of more line pixels with smaller radii than $K_u$, $K_g$ and $K_s$. Section 5.2 showed that, for an ideal line, the line pixel detection effectiveness for each kernel followed the resulting sequence (from best to worst): $\mathcal{K}$, $K_s$, $K_u$ and $K_g$. Looking at Figures 6.4 (a)-(c), the same sequence of kernel effectiveness arises for all images. $\mathcal{K}$ curve rises faster than the monotonic decreasing kernels. Hence, these results are consistent with the theory exposed in Chapters 4 and 5, i.e., the kernel $\mathcal{K}$ needs a smaller radius to detect a thick line than the kernels $K_u$, $K_g$ and $K_s$.

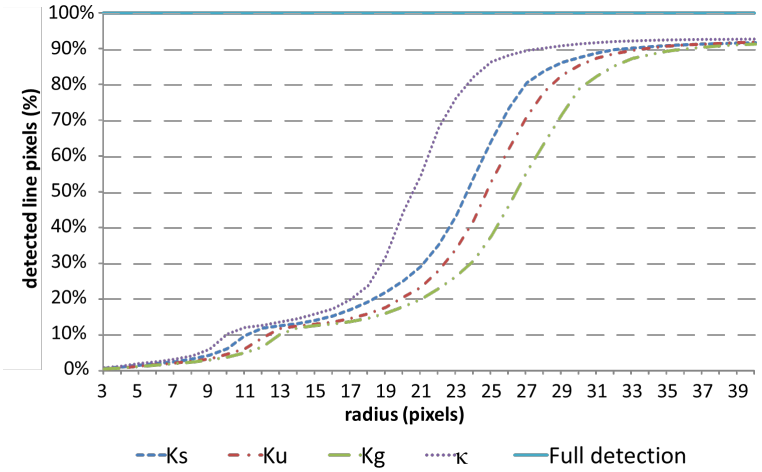
## 6.5 Width Estimates

In this section we analyze the CWLD regarding its local width estimation in the same natural images from the previous experiments. We evaluate the performance of the CWLD using several combinations of color-differences (CIE 2000, CIE 1994, CMC 1984, CIE 1976, RGB Euclidean distance, $L^*$ and gray scale) and kernels ($\mathcal{K}$, $K_s$, $K_u$ and $K_g$). To enable a better visualization of the results we compare the estimation obtained from the CWLD with the measurements obtained only at the pixels of the center line. For each kernel we use the smallest kernel that detects most the line pixels, which can be sampled from Figure 6.4. Table 6.1 shows the radius in pixels used for each kernel.

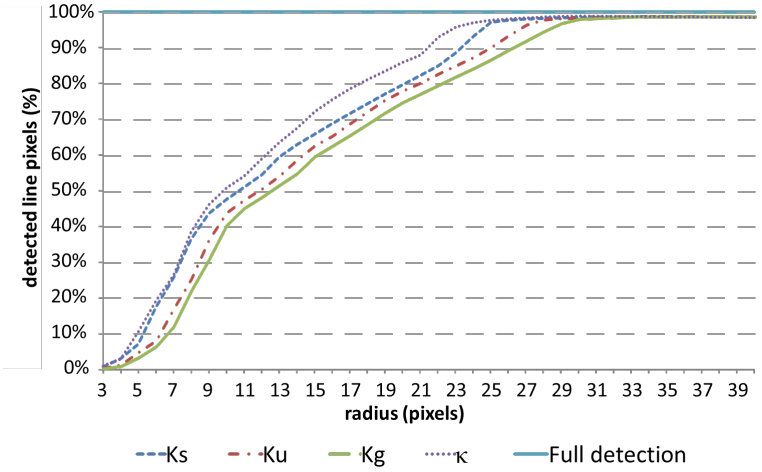
Table 6.2 presents the mean, variance, standard deviation and minimum and maximum width values (in pixels) for Figure 6.1 (a), considering different combinations of color-differences and kernels. Table 6.3 shows the results, using the same combinations of kernels and color-differences presented previously, for Figure 6.1 (b), while Table 6.10 presents the results obtained for Figure 6.1 (c). The results shown in these tables permit to compare the variations of ground truth and the CWLD under different color-differences

Figure 6.4: The detection ratio of the CWLD for all kernels. Figures (a)-(c) are associated to the natural images shown in Figure 6.1 (a)-(c), respectively.

(a)




(b)




(c)

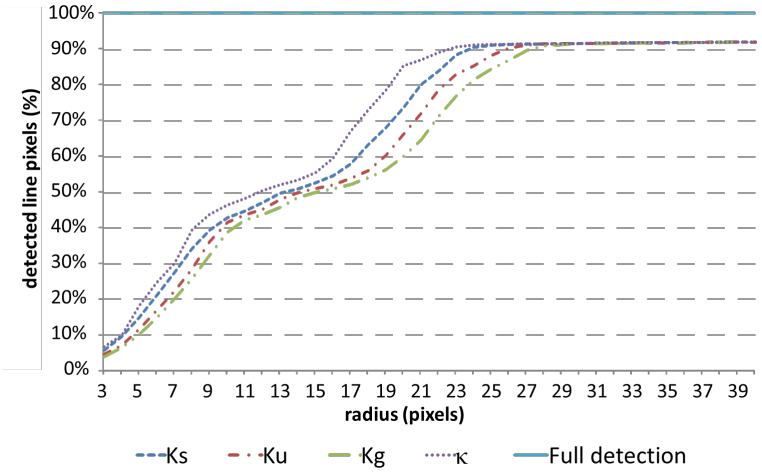


Table 6.1: Radius in pixels used in the width estimation

| **Kernel** | **Figure 6.1 (a)** | **Figure 6.1 (b)** | **Figure 6.1 (c)** |
|---|---|---|---|
| $K_g$ | 32 | 28 | 29 |
| $K_u$ | 30 | 26 | 29 |
| $K_s$ | 28 | 24 | 28 |
| $\mathcal{K}$ | 25 | 22 | 27 |

and kernels. As the WLD originally does not estimate the widths of lines locally, we adapt it to use the same strategy employed by the CWLD to estimate widths. These results are presented as the CWLD using **grayscale** difference in all tables of this section.

Figures 6.5, 6.6 and 6.7 display the Error Distribution in pixels for Figures 6.1 (a)-(c), respectively, using different color-differences and kernels. Each color-difference chart includes the error distributions using $K_u, K_s, \mathcal{K}$ and $K_g$.

Note that the information so far is not sufficient to analyze the results, in the next section we present further detailed statistical tests.

### 6.5.1 Detailed Statistics

We provide two summary tables to make the visualization of statistical results clearer. Table 6.11 shows what color-difference achieved the best scores for each statistic, considering Figures 6.1 (a)-(c) individually. Table 6.12 shows the kernels with the best results for each statistic, considering Figures 6.1 (a)-(c) separately. A more detailed view of Table 6.11 and 6.12 is shown in Tables 6.4 and 6.5, for Figure 6.1 (a); Tables 6.6 and 6.7, for Figure 6.1 (b); and Tables 6.8 and 6.9, for Figure 6.1 (c). The detailed Tables are also presented and were partitioned in two pages due to the large number of data. In each table the bold font is used to point out the best results to facilitate the analysis.

Table 6.2: Ground Truth and Estimation data using different kernels and color-differences for Figure 6.1 (a)

| | Ground Truth | CIELAB L* Ku | CIELAB L* Ks | CIELAB L* k | CIELAB L* Kg | CMC Ku | CMC Ks | CMC k | CMC Kg |
|---|---|---|---|---|---|---|---|---|---|
| **Mean** | 15.58 | 9.95 | 8.53 | 10.36 | 10.12 | 16.01 | 14.64 | 15.18 | 16.36 |
| **Variance** | 13.71 | 101.45 | 83.34 | 89.10 | 106.79 | 32.58 | 30.35 | 25.91 | 34.55 |
| **Std. Dev.** | 3.70 | 10.07 | 9.13 | 9.44 | 10.33 | 5.71 | 5.51 | 5.09 | 5.88 |
| **Max** | 25.50 | 27.07 | 24.32 | 24.57 | 28.00 | 26.62 | 23.91 | 24.45 | 27.26 |
| **Min** | 7.07 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | **Ground Truth** | **CIELAB 1976 Ku** | **CIELAB 1976 Ks** | **CIELAB 1976 k** | **CIELAB 1976 Kg** | **CIELAB 1994 Ku** | **CIELAB 1994 Ks** | **CIELAB 1994 k** | **CIELAB 1994 Kg** |
| **Mean** | 15.58 | 16.25 | 14.87 | 15.24 | 16.56 | 16.66 | 15.13 | 15.56 | 17.01 |
| **Variance** | 13.71 | 26.07 | 24.54 | 21.49 | 27.02 | 25.51 | 25.27 | 21.10 | 27.55 |
| **Std. Dev.** | 3.70 | 5.11 | 4.95 | 4.64 | 5.20 | 5.05 | 5.03 | 4.59 | 5.25 |
| **Max** | 25.50 | 25.24 | 22.96 | 23.30 | 27.92 | 27.08 | 24.53 | 23.49 | 27.10 |
| **Min** | 7.07 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | **Ground Truth** | **CIELAB 2000 Ku** | **CIELAB 2000 Ks** | **CIELAB 2000 k** | **CIELAB 2000 Kg** | **Grayscale Ku** | **Grayscale Ks** | **Grayscale k** | **Grayscale Kg** |
| **Mean** | 15.58 | 15.34 | 14.12 | 14.51 | 15.66 | 11.92 | 10.04 | 11.55 | 11.98 |
| **Variance** | 13.71 | 24.38 | 21.86 | 20.54 | 26.01 | 98.47 | 82.92 | 82.62 | 104.68 |
| **Std. Dev.** | 3.70 | 4.94 | 4.68 | 4.53 | 5.10 | 9.92 | 9.11 | 9.09 | 10.23 |
| **Max** | 25.50 | 21.62 | 21.50 | 24.59 | 22.48 | 26.96 | 24.44 | 24.63 | 27.96 |
| **Min** | 7.07 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | **Ground Truth** | **RGB Distance Ku** | **RGB Distance Ks** | **RGB Distance k** | **RGB Distance Kg** | | | | |
| **Mean** | 15.58 | 16.39 | 14.49 | 15.80 | 16.73 | | | | |
| **Variance** | 13.71 | 55.26 | 53.16 | 41.77 | 59.47 | | | | |
| **Std. Dev.** | 3.70 | 7.43 | 7.29 | 6.46 | 7.71 | | | | |
| **Max** | 25.50 | 27.04 | 24.50 | 24.58 | 28.02 | | | | |
| **Min** | 7.07 | 0.00 | 0.00 | 0.00 | 0.00 | | | | |

Table 6.3: Ground Truth and Estimation data using different kernels and color-differences for Figure 6.1 (b)

| | Ground Truth | CIELAB L* Ku | CIELAB L* Ks | CIELAB L* k | CIELAB L* Kg | CMC Ku | CMC Ks | CMC k | CMC Kg |
|---|---|---|---|---|---|---|---|---|---|
| **Mean** | 7.70 | 6.78 | 6.19 | 6.60 | 6.85 | 7.48 | 6.86 | 7.22 | 7.55 |
| **Variance** | 35.21 | 17.57 | 15.11 | 16.39 | 18.00 | 21.95 | 19.36 | 19.39 | 22.65 |
| **Std. Dev.** | 5.93 | 4.19 | 3.89 | 4.05 | 4.24 | 4.68 | 4.40 | 4.40 | 4.76 |
| **Max** | 20.88 | 19.13 | 17.91 | 19.64 | 19.68 | 20.40 | 19.18 | 19.08 | 20.60 |
| **Min** | 1.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.25 | 0.17 | 0.38 | 0.23 |
| | **Ground Truth** | **CIELAB 1976 Ku** | **CIELAB 1976 Ks** | **CIELAB 1976 k** | **CIELAB 1976 Kg** | **CIELAB 1994 Ku** | **CIELAB 1994 Ks** | **CIELAB 1994 k** | **CIELAB 1994 Kg** |
| **Mean** | 7.70 | 8.14 | 7.48 | 7.84 | 8.23 | 6.75 | 6.20 | 6.49 | 6.81 |
| **Variance** | 35.21 | 25.91 | 23.13 | 22.05 | 26.66 | 18.79 | 16.23 | 16.98 | 19.22 |
| **Std. Dev.** | 5.93 | 5.09 | 4.81 | 4.70 | 5.16 | 4.33 | 4.03 | 4.12 | 4.38 |
| **Max** | 20.88 | 21.69 | 19.84 | 19.27 | 21.04 | 19.72 | 17.96 | 17.90 | 20.13 |
| **Min** | 1.00 | 1.64 | 1.46 | 1.44 | 1.63 | 0.05 | 0.18 | 0.00 | 0.01 |
| | **Ground Truth** | **CIELAB 2000 Ku** | **CIELAB 2000 Ks** | **CIELAB 2000 k** | **CIELAB 2000 Kg** | **Grayscale Ku** | **Grayscale Ks** | **Grayscale k** | **Grayscale Kg** |
| **Mean** | 7.70 | 7.58 | 6.94 | 7.29 | 7.65 | 6.14 | 5.60 | 5.93 | 6.20 |
| **Variance** | 35.21 | 21.03 | 18.45 | 18.45 | 21.97 | 15.08 | 12.70 | 14.10 | 15.46 |
| **Std. Dev**. | 5.93 | 4.59 | 4.30 | 4.29 | 4.69 | 3.88 | 3.56 | 3.76 | 3.93 |
| **Max** | 20.88 | 20.39 | 18.38 | 18.42 | 20.85 | 18.75 | 17.04 | 17.42 | 19.02 |
| **Min** | 1.00 | 0.25 | 0.17 | 0.38 | 0.23 | 1.28 | 1.05 | 0.93 | 1.29 |
| | **Ground Truth** | **RGB Distance Ku** | **RGB Distance Ks** | **RGB Distance k** | **RGB Distance Kg** | | | | |
| **Mean** | 7,70 | 5.75 | 5.28 | 5.55 | 5.83 | | | | |
| **Variance** | 35.21 | 15.94 | 13.40 | 15.16 | 16.20 | | | | |
| **Std. Dev**. | 5.93 | 3.99 | 3.66 | 3.89 | 4.02 | | | | |
| **Max** | 20.88 | 18.24 | 17.42 | 17.32 | 19.04 | | | | |
| **Min** | 1.00 | 0.01 | 0.03 | 0.00 | 0.02 | | | | |

Table 6.4: Statistics for image 1

| | **CIELAB L*** | | | | **CMC** | | | | **CIELAB 1976** | | | | **CIELAB 1994** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Ku** | **Ks** | **k** | **Kg** | **Ku** | **Ks** | **k** | **Kg** | **Ku** | **Ks** | **k** | **Kg** | **Ku** | **Ks** | **k** | **Kg** |
| $\rho$ | 0.13 | 0.11 | 0.23 | 0.13 | 0.71 | 0.69 | 0.73 | 0.72 | **0.79** | 0.75 | **0.79** | 0.79 | 0.78 | 0.72 | 0.76 | 0.77 |
| $MSE$ | 136.74 | 139.37 | 114.12 | 140.42 | 16.28 | 16.95 | 12.37 | 17.67 | 10.36 | 11.27 | **8.35** | 11.35 | 11.34 | 12.34 | 8.90 | 13.35 |
| $RMSE$ | 11.69 | 11.81 | 10.68 | 11.85 | 4.03 | 4.12 | 3.52 | 4.20 | 3.22 | 3.36 | **2.89** | 3.37 | 3.37 | 3.51 | 2.98 | 3.65 |
| $N_{RMSE}$ | 0.43 | 0.49 | 0.43 | 0.42 | 0.15 | 0.17 | 0.14 | 0.15 | 0.13 | 0.15 | **0.12** | **0.12** | **0.12** | **0.14** | 0.13 | 0.13 |
| $\xi_{abs}$ | 9.40 | 9.25 | 8.02 | 9.67 | 2.99 | 2.82 | 2.51 | 3.17 | 2.47 | **2.39** | **2.12** | 2.64 | 2.57 | **2.39** | 2.14 | 2.83 |
| $M$ | 7.03 | 6.52 | 4.49 | 7.86 | 2.40 | 2.12 | 1.85 | 2.68 | 2.07 | 1.86 | **1.56** | 2.31 | 2.15 | **1.82** | 1.57 | 2.40 |
| $\tau$ | 15.62 | 15.73 | 15.23 | 15.62 | 3.91 | 3.44 | 3.34 | 4.20 | 3.28 | 3.07 | **2.70** | 3.53 | 3.42 | **2.95** | 2.76 | 3.81 |
| $KS(5\%, 1.36)$ | 8.73 | 10.13 | 7.97 | 8.73 | 5.17 | 3.61 | 3.61 | 5.48 | 5.09 | 2.79 | 3.07 | 5.48 | 5.42 | 3.31 | 3.75 | 6.11 |
| $KS(1\%, 1.63)$ | 8.73 | 10.13 | 7.97 | 8.73 | 5.17 | 3.61 | 3.61 | 5.48 | 5.09 | 2.79 | 3.07 | 5.48 | 5.42 | 3.31 | 3.75 | 6.11 |
| $KS(.1\%, 1.95)$ | 8.73 | 10.13 | 7.97 | 8.73 | 5.17 | 3.61 | 3.61 | 5.48 | 5.09 | 2.79 | 3.07 | 5.48 | 5.42 | 3.31 | 3.75 | 6.11 |
| $BC$ | 0.51 | 0.52 | 0.60 | 0.50 | 0.87 | 0.84 | 0.90 | 0.85 | **0.89** | 0.88 | 0.91 | 0.88 | **0.89** | **0.89** | **0.92** | 0.87 |
| $BD$ | 0.67 | 0.66 | 0.52 | 0.69 | 0.14 | 0.17 | 0.10 | 0.16 | **0.12** | 0.12 | 0.10 | 0.13 | **0.12** | **0.12** | **0.09** | 0.14 |
| $HD$ | 0.70 | 0.70 | 0.63 | 0.71 | 0.36 | 0.40 | 0.31 | 0.39 | **0.33** | 0.34 | 0.30 | 0.34 | **0.33** | **0.33** | **0.29** | 0.36 |
| $Hits(< 10\%)$ | 11.5% | 13.3% | 22.5% | 9.6% | 31.5% | 36.7% | 43.5% | 29.5% | 36.6% | 43.9% | **49.8%** | 33.1% | 37.6% | **44.1%** | 49.6% | 33.6% |
| $Hits(< 20\%)$ | 24.4% | 31.3% | 39.3% | 22.6% | 61.8% | 64.5% | 70.0% | 58.0% | 69.7% | 67.3% | **75.6%** | 65.1% | 65.8% | **70.6%** | 74.5% | 63.4% |
| $Hits(< 30\%)$ | 39.0% | 43.6% | 49.0% | 35.4% | 81.6% | 82.3% | 82.5% | 78.4% | 84.4% | 83.7% | 85.8% | 83.5% | 84.7% | **84.1%** | **85.9%** | 82.0% |

Table 6.5: Statistics for image 1 - Continuation of Table 6.4

| | **CIELAB 2000** | | | | **Grayscale** | | | | **RGB Distance** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Ku** | **Ks** | **k** | **Kg** | **Ku** | **Ks** | **k** | **Kg** | **Ku** | **Ks** | **k** | **Kg** |
| $\rho$ | **0.79** | **0.77** | 0.78 | **0.80** | 0.19 | 0.16 | 0.19 | 0.17 | 0.54 | 0.47 | 0.54 | 0.53 |
| $MSE$ | **9.13** | **10.90** | 9.07 | **9.59** | 111.66 | 116.32 | 99.78 | 118.23 | 40.04 | 42.53 | 29.65 | 44.05 |
| $RMSE$ | **3.02** | **3.30** | 3.01 | **3.10** | 10.57 | 10.78 | 9.99 | 10.87 | 6.33 | 6.52 | 5.44 | 6.64 |
| $N_{RMSE}$ | 0.14 | 0.15 | 0.12 | 0.14 | 0.39 | 0.44 | 0.41 | 0.39 | 0.23 | 0.27 | 0.22 | 0.24 |
| $\xi_{abs}$ | **2.30** | 2.50 | 2.26 | **2.44** | 8.17 | 8.07 | 7.17 | 8.62 | 4.58 | 4.32 | 3.64 | 4.97 |
| $M$ | **1.87** | 2.00 | 1.76 | **2.06** | 5.39 | 4.03 | 3.59 | 6.22 | 3.23 | 2.42 | 2.24 | 3.60 |
| $\tau$ | **3.11** | 3.45 | 3.02 | **3.27** | 15.00 | 15.23 | 14.42 | 15.23 | 5.68 | 5.06 | 4.63 | 6.23 |
| $KS(5\%, 1.36)$ | 3.42 | 3.12 | 2.30 | 4.19 | 6.93 | 8.60 | 6.65 | 7.12 | 7.42 | 5.15 | 6.49 | 7.91 |
| $KS(1\%, 1.63)$ | 3.42 | 3.12 | 2.30 | 4.19 | 6.93 | 8.60 | 6.65 | 7.12 | 7.42 | 5.15 | 6.49 | 7.91 |
| $KS(.1\%, 1.95)$ | 3.42 | 3.12 | 2.30 | 4.19 | 6.93 | 8.60 | 6.65 | 7.12 | 7.42 | 5.15 | 6.49 | 7.91 |
| $BC$ | **0.89** | 0.88 | 0.89 | **0.89** | 0.58 | 0.58 | 0.66 | 0.55 | 0.74 | 0.77 | 0.81 | 0.71 |
| $BD$ | **0.11** | 0.13 | 0.11 | **0.12** | 0.54 | 0.54 | 0.42 | 0.60 | 0.30 | 0.26 | 0.21 | 0.34 |
| $HD$ | **0.33** | 0.35 | 0.33 | **0.33** | 0.65 | 0.64 | 0.58 | 0.67 | 0.51 | 0.48 | 0.43 | 0.54 |
| $Hits(< 10\%)$ | **41.1%** | 40.8% | 44.7% | **35.1%** | 15.3% | 19.2% | 28.2% | 11.7% | 24.9% | 34.0% | 39.0% | 19.3% |
| $Hits(< 20\%)$ | **70.3%** | 64.8% | 72.3% | **68.7%** | 33.4% | 39.3% | 45.0% | 27.6% | 48.3% | 55.8% | 60.3% | 44.7% |
| $Hits(< 30\%)$ | **85.6%** | 82.2% | 84.4% | **85.2%** | 46.3% | 50.5% | 52.5% | 40.9% | 66.9% | 69.4% | 72.0% | 63.1% |

Table 6.6: Statistics for image 2

| | **CIELAB L*** | | | | **CMC** | | | | **CIELAB 1976** | | | | **CIELAB 1994** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Ku** | **Ks** | **k** | **Kg** | **Ku** | **Ks** | **k** | **Kg** | **Ku** | **Ks** | **k** | **Kg** | **Ku** | **Ks** | **k** | **Kg** |
| $\rho$ | 0.81 | 0.82 | 0.76 | 0.82 | 0.84 | 0.85 | 0.80 | 0.85 | **0.88** | **0.89** | **0.84** | **0.89** | 0.82 | 0.84 | 0.78 | 0.83 |
| $MSE$ | 13.38 | 14.59 | 16.46 | 12.73 | 10.41 | 10.60 | 13.17 | 9.79 | **8.13** | **7.39** | **10.47** | **7.72** | 12.48 | 13.68 | 15.57 | 11.95 |
| $RMSE$ | 3.66 | 3.82 | 4.06 | 3.57 | 3.23 | 3.26 | 3.63 | 3.13 | **2.85** | **2.72** | **3.24** | **2.78** | 3.53 | 3.70 | 3.95 | 3.46 |
| $N_{RMSE}$ | 0.19 | 0.21 | 0.21 | 0.18 | 0.16 | 0.17 | 0.19 | 0.15 | **0.14** | **0.15** | **0.18** | **0.14** | 0.18 | 0.21 | 0.22 | 0.17 |
| $\xi_{abs}$ | 2.62 | 2.69 | 2.94 | 2.54 | 2.25 | 2.24 | 2.57 | 2.20 | **2.05** | **1.93** | **2.30** | **2.01** | 2.52 | 2.56 | 2.88 | 2.44 |
| $M$ | 1.72 | 1.74 | 1.88 | 1.59 | 1.46 | 1.40 | 1.67 | 1.49 | **1.35** | **1.36** | **1.45** | **1.36** | 1.66 | 1.75 | 1.99 | 1.61 |
| $\tau$ | 3.56 | 3.51 | 4.22 | 3.48 | 2.95 | 2.88 | 3.51 | 2.80 | **2.69** | **2.57** | **3.09** | **2.73** | 3.33 | 3.21 | 3.83 | 3.30 |
| $KS(5\%, 1.36)$ | 2.62 | 2.78 | 2.66 | 2.56 | 1.57 | 2.14 | 2.30 | 1.70 | 2.27 | 1.57 | 2.02 | 2.40 | 2.40 | 2.69 | 2.72 | 2.37 |
| $KS(1\%, 1.63)$ | 2.62 | 2.78 | 2.66 | 2.56 | **1.57** | 2.14 | 2.30 | 1.70 | 2.27 | **1.57** | 2.02 | 2.40 | 2.40 | 2.69 | 2.72 | 2.37 |
| $KS(.1\%, 1.95)$ | 2.62 | 2.78 | 2.66 | 2.56 | **1.57** | 2.14 | 2.30 | **1.70** | 2.27 | **1.57** | 2.02 | 2.40 | 2.40 | 2.69 | 2.72 | 2.37 |
| $BC$ | 0.90 | 0.90 | 0.88 | 0.90 | **0.93** | 0.93 | **0.89** | **0.94** | 0.92 | **0.94** | **0.89** | 0.92 | **0.93** | 0.90 | 0.88 | 0.93 |
| $BD$ | 0.10 | 0.11 | 0.13 | 0.10 | **0.07** | 0.08 | **0.11** | **0.06** | 0.09 | **0.06** | **0.11** | 0.09 | **0.07** | 0.10 | 0.13 | 0.07 |
| $HD$ | 0.31 | 0.32 | 0.34 | 0.31 | **0.26** | 0.27 | 0.33 | **0.25** | 0.29 | **0.24** | **0.32** | 0.29 | **0.26** | 0.31 | 0.35 | 0.27 |
| $Hits(< 10\%)$ | 16.2% | 15.4% | 13.7% | 15.8% | 20.1% | 20.3% | 19.7% | 20.5% | **24.0%** | **24.2%** | **25.0%** | **25.0%** | 15.4% | 16.8% | 12.7% | 16.0% |
| $Hits(< 20\%)$ | 32.8% | 30.1% | 27.9% | 33.6% | 42.8% | 36.9% | 40.0% | 40.8% | **46.9%** | **44.3%** | **41.0%** | **47.1%** | 33.2% | 33.6% | 30.9% | 36.1% |
| $Hits(< 30\%)$ | 48.8% | 45.1% | 47.1% | 50.6% | 57.2% | 57.8% | 52.3% | 57.8% | **60.0%** | **61.9%** | **56.6%** | **58.4%** | 51.6% | 49.8% | 48.0% | 52.0% |

Table 6.7: Statistics for image 2 - Continuation of Table 6.6

| | **CIELAB 2000** | | | | **Grayscale** | | | | **RGB Distance** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Ku** | **Ks** | **k** | **Kg** | **Ku** | **Ks** | **k** | **Kg** | **Ku** | **Ks** | **k** | **Kg** |
| $\rho$ | 0.85 | 0.86 | 0.81 | 0.86 | 0.84 | 0.85 | 0.79 | 0.84 | 0.79 | 0.80 | 0.73 | 0.79 |
| $MSE$ | 10.00 | 10.14 | 12.73 | 9.38 | 14.06 | 16.26 | 17.04 | 13.48 | 17.71 | 19.75 | 21.08 | 16.98 |
| $RMSE$ | 3.16 | 3.19 | 3.57 | 3.06 | 3.75 | 4.03 | 4.13 | 3.67 | 4.21 | 4.44 | 4.59 | 4.12 |
| $N_{RMSE}$ | 0.16 | 0.17 | 0.20 | 0.15 | 0.21 | 0.25 | 0.25 | 0.21 | 0.23 | 0.26 | 0.27 | 0.22 |
| $\xi_{abs}$ | 2.21 | 2.18 | 2.52 | 2.15 | 2.69 | 2.83 | 3.03 | 2.62 | 2.97 | 3.09 | 3.35 | 2.88 |
| $M$ | 1.41 | 1.40 | 1.61 | 1.43 | 1.75 | 1.96 | 2.12 | 1.77 | 1.93 | 2.07 | 2.44 | 1.87 |
| $\tau$ | 2.82 | 2.73 | 3.46 | 2.78 | 3.58 | 3.61 | 4.09 | 3.43 | 3.99 | 4.06 | 4.49 | 3.90 |
| $KS(5\%, 1.36)$ | 1.82 | 2.30 | 2.37 | 1.89 | 2.85 | 2.98 | 2.91 | 2.88 | 2.88 | 2.98 | 2.94 | 2.85 |
| $KS(1\%, 1.67)$ | 1.82 | 2.30 | 2.37 | 1.89 | 2.85 | 2.98 | 2.91 | 2.88 | 2.88 | 2.98 | 2.94 | 2.85 |
| $KS(.1\%, 1.95)$ | **1.82** | 2.30 | 2.37 | **1.89** | 2.85 | 2.98 | 2.91 | 2.88 | 2.88 | 2.98 | 2.94 | 2.85 |
| $BC$ | 0.92 | 0.91 | 0.88 | 0.93 | 0.90 | 0.88 | 0.87 | 0.90 | 0.90 | 0.88 | 0.87 | 0.91 |
| $BD$ | 0.08 | 0.09 | 0.13 | 0.07 | 0.10 | 0.12 | 0.14 | 0.10 | 0.11 | 0.13 | 0.14 | 0.09 |
| $HD$ | 0.28 | 0.30 | 0.35 | 0.26 | 0.31 | 0.34 | 0.36 | 0.31 | 0.32 | 0.35 | 0.36 | 0.29 |
| $Hits(< 10\%)$ | 21.1% | 22.3% | 19.9% | 22.1% | 14.8% | 12.3% | 11.3% | 16.0% | 12.3% | 12.3% | 9.4% | 14.5% |
| $Hits(< 20\%)$ | 45.1% | 38.3% | 40.2% | 43.9% | 26.8% | 29.1% | 25.0% | 27.0% | 23.8% | 25.6% | 21.3% | 26.6% |
| $Hits(< 30\%)$ | 57.4% | 61.7% | 54.5% | 57.8% | 48.4% | 44.3% | 40.4% | 49.4% | 43.2% | 41.6% | 35.2% | 45.5% |

Table 6.8: Statistics for image 3

| | **CIELAB L*** | | | | **CMC** | | | | **CIELAB 1976** | | | | **CIELAB 1994** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Ku** | **Ks** | **k** | **Kg** | **Ku** | **Ks** | **k** | **Kg** | **Ku** | **Ks** | **k** | **Kg** | **Ku** | **Ks** | **k** | **Kg** |
| $\rho$ | 0.35 | 0.35 | 0.30 | 0.01 | 0.83 | 0.83 | 0.76 | 0.64 | 0.81 | 0.82 | 0.75 | 0.82 | 0.83 | 0.83 | 0.75 | 0.67 |
| $MSE$ | 58.37 | 46.50 | 68.64 | 67.39 | 10.65 | 8.21 | 12.94 | 18.57 | 9.11 | 8.76 | 12.12 | 9.06 | 10.05 | 8.22 | 12.96 | 17.10 |
| $RMSE$ | 7.64 | 6.82 | 8.29 | 8.21 | 3.26 | 2.87 | 3.60 | 4.31 | 3.02 | 2.96 | 3.48 | 3.01 | 3.17 | 2.87 | 3.60 | 4.13 |
| $N_{RMSE}$ | 0.32 | 0.28 | 0.35 | 0.34 | 0.14 | 0.12 | 0.15 | 0.18 | 0.13 | 0.12 | 0.15 | 0.13 | 0.13 | 0.12 | 0.15 | 0.17 |
| $\xi_{abs}$ | 5.88 | 5.02 | 6.36 | 6.14 | 2.04 | 1.66 | 2.19 | 2.34 | 1.67 | 1.89 | 1.98 | 1.63 | 1.91 | **1.65** | 2.10 | 2.20 |
| $M$ | 4.57 | 3.81 | 4.92 | 4.61 | 1.19 | 0.82 | 1.10 | 1.21 | 0.82 | 1.24 | 0.94 | 0.79 | 0.99 | **0.77** | 1.06 | 1.04 |
| $\tau$ | 9.12 | 7.83 | 10.21 | 8.82 | 2.27 | 1.84 | 2.33 | 2.36 | 1.82 | 2.23 | 2.19 | 1.78 | 2.04 | **1.80** | 2.21 | 2.14 |
| $KS(5\%, 1.36)$ | 13.09 | 11.03 | 13.63 | 11.97 | 5.99 | 1.97 | 5.83 | 5.27 | **0.88** | 3.99 | **0.88** | **0.83** | 5.80 | 1.79 | 4.60 | 5.27 |
| $KS(1\%, 1.63)$ | 13.09 | 11.03 | 13.63 | 11.97 | 5.99 | 1.97 | 5.83 | 5.27 | **0.88** | 3.99 | **0.88** | **0.83** | 5.80 | 1.79 | 4.60 | 5.27 |
| $KS(.1\%, 1.95)$ | 13.09 | 11.03 | 13.63 | 11.97 | 5.99 | 1.97 | 5.83 | 5.27 | **0.88** | 3.99 | **0.88** | **0.83** | 5.80 | **1.79** | 4.60 | 5.27 |
| $BC$ | 0.72 | 0.79 | 0.72 | 0.71 | 0.96 | **0.98** | 0.95 | 0.95 | **0.98** | 0.96 | **0.98** | **0.98** | 0.95 | 0.97 | 0.96 | 0.94 |
| $BD$ | 0.33 | 0.23 | 0.33 | 0.34 | 0.05 | **0.02** | 0.05 | 0.05 | **0.02** | 0.04 | **0.02** | **0.02** | 0.05 | 0.03 | 0.04 | 0.06 |
| $HD$ | 0.53 | 0.46 | 0.53 | 0.54 | 0.21 | **0.15** | 0.22 | 0.23 | **0.13** | 0.21 | 0.16 | **0.14** | 0.22 | 0.17 | 0.20 | 0.24 |
| $Hits(< 10\%)$ | 10.4% | 20.4% | 13.2% | 9.4% | 29.3% | 43.8% | 30.0% | 27.5% | 40.2% | 23.7% | 37.4% | 41.0% | 32.2% | **45.1%** | 35.3% | 31.2% |
| $Hits(< 20\%)$ | 21.3% | 31.4% | 23.4% | 20.0% | 54.9% | 68.0% | 55.8% | 52.3% | 64.9% | 53.4% | 60.5% | 66.4% | 60.5% | **68.6%** | 56.9% | 58.5% |
| $Hits(< 30\%)$ | 30.5% | 38.5% | 31.5% | 28.4% | 70.1% | 77.1% | 69.5% | 67.8% | **77.6%** | 72.5% | 74.2% | **79.8%** | 71.6% | 78.3% | 71.6% | 70.6% |

Table 6.9: Statistics for image 3 - Continuation of Table 6.8

| | **CIELAB 2000** | | | | **Grayscale** | | | | **RGB Distance** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Ku** | **Ks** | **k** | **Kg** | **Ku** | **Ks** | **k** | **Kg** | **Ku** | **Ks** | **k** | **Kg** |
| $\rho$ | **0.85** | **0.85** | **0.77** | **0.83** | 0.36 | 0.54 | 0.30 | 0.06 | 0.77 | 0.75 | 0.75 | 0.52 |
| $MSE$ | **7.76** | **6.95** | **11.02** | **8.86** | 56.72 | 34.31 | 68.18 | 63.77 | 13.37 | 12.54 | 13.53 | 24.05 |
| $RMSE$ | **2.79** | **2.64** | **3.32** | **2.98** | 7.53 | 5.86 | 8.26 | 7.99 | 3.66 | 3.54 | 3.68 | 4.90 |
| $N_{RMSE}$ | **0.12** | **0.11** | **0.14** | **0.12** | 0.31 | 0.24 | 0.34 | 0.33 | 0.15 | 0.15 | 0.15 | 0.20 |
| $\xi_{abs}$ | **1.61** | 1.67 | **1.91** | **1.62** | 5.74 | 4.12 | 6.27 | 5.93 | 2.09 | 1.86 | 2.14 | 2.50 |
| $M$ | **0.75** | 0.98 | **0.78** | **0.72** | 4.29 | 2.28 | 4.71 | 4.26 | 1.02 | 0.78 | 1.05 | 1.04 |
| $\tau$ | **1.72** | 1.93 | **2.03** | **1.72** | 8.71 | 6.98 | 10.01 | 8.53 | 2.15 | **1.78** | 2.24 | 2.24 |
| $KS(5\%, 1.36)$ | 2.53 | 1.62 | 1.91 | 2.64 | 12.87 | 7.19 | 13.27 | 11.84 | 4.69 | **1.39** | 4.04 | 4.19 |
| $KS(1\%, 1.63)$ | 2.53 | **1.62** | 1.91 | 2.64 | 12.87 | 7.19 | 13.27 | 11.84 | 4.69 | **1.39** | 4.04 | 4.19 |
| $KS(.1\%, 1.95)$ | 2.53 | **1.62** | **1.91** | 2.64 | 12.87 | 7.19 | 13.27 | 11.84 | 4.69 | **1.39** | 4.04 | 4.19 |
| $BC$ | **0.98** | 0.97 | **0.98** | 0.97 | 0.72 | 0.88 | 0.72 | 0.72 | 0.94 | **0.98** | 0.96 | 0.94 |
| $BD$ | **0.02** | 0.03 | **0.02** | 0.03 | 0.33 | 0.13 | 0.32 | 0.32 | 0.06 | **0.02** | 0.04 | 0.06 |
| $HD$ | **0.15** | 0.18 | **0.15** | 0.16 | 0.53 | 0.35 | 0.52 | 0.53 | 0.24 | **0.15** | 0.20 | 0.24 |
| $Hits(< 10\%)$ | **46.1%** | 32.0% | **42.9%** | **45.2%** | 10.6% | 19.1% | 13.3% | 10.6% | 32.6% | 43.4% | 34.8% | 30.5% |
| $Hits(< 20\%)$ | **69.2%** | 64.8% | **64.1%** | **70.1%** | 22.5% | 39.7% | 24.3% | 20.9% | 58.0% | 66.9% | 56.3% | 58.3% |
| $Hits(< 30\%)$ | 77.5% | **79.5%** | **76.1%** | 78.5% | 32.2% | 53.7% | 32.2% | 30.3% | 70.5% | 77.8% | 70.5% | 68.7% |

Figures 6.8 and 6.9 display the absolute errors in pixels for Figures 6.1 (a)-(c) using the two best color-differences, namely CIE 2000 and CIE 1976, respectively, as we saw in Table 6.11. These figures are useful to isolate the line regions with more errors for each test image.

## 6.6 Analysis of the Results

Tables 6.2, 6.3 and 6.10 show that the mean values for color-differences, namely CIE 1976, CMC 1984, CIE1994, CIE 2000 and RGB, are similar to those of the ground truth, specially for perceptual color-differences such as CIE and CMC. The mean results for the RGB Euclidean distance oscillate for each natural image, but with slightly better values than those obtained using $L^*$ and the gray scale differences in Figures 6.1 (a) and (c). $L^*$ and the gray scale differences have mean values similar to the ground-truth only in Figure 6.1 (b). The results for the variances, and corresponding standard deviations, show slightly increased differences between ground-truth and the CWLD estimation. The best variance results are obtained with perceptual color-differences, followed by RGB, $L^*$ and the gray scale differences. That happens for several reasons including noise, color-difference errors, density estimation errors, etc. Besides, the CWLD determines the local width considering a relatively large surrounding region, so it can smooth lines, whose width varies abruptly inside the kernel region. Despite that, the statistical analysis highlights the combination of color-differences and kernels provides the best estimates regarding the local line width.

In relation to the correlation statistic, the closer the result is to 1, the stronger is the correlation between the ground truth and the estimation. For gray scale and $L^*$ intensity differences, the correlation results are weak for Figures 6.1 (a) and (c) independently of the kernel used, whereas Figure 6.1 (b) shows strong correlation (>0.7) using these differences (see Tables 6.4, 6.5, 6.6, 6.7, 6.8 and 6.9), also independently of the kernel used. The correlation values are slightly better using the RGB Euclidean distance (especially for Figure 6.1 (c)). A kernel comparison is not needed here, since all of them scored well (>0.7) using perceptual color-differences, still we point out that $K_g$ ($\rho \approx 0.80$, CIE 2000), $K_s$ and $K_g$ (both $\rho \approx 0.89$, CIE 1976) and, $K_u$ and $K_s$ (both $\rho \approx 0.85$, CIE 2000) showed the best results for Figures 6.1 (a), (b) and (c) in Table 6.12, respectively. Besides, if we look at the correlation values for perceptual color-differences, such as the CMC$(l, c)$ and CIE color-differences, they show a strong correlation between the ground truth and the CWLD estimation for all figures. This is an indication that the perceptual color-differences may be more effective than gray-scale differences to obtain reasonable density estimates, and, consequently, good width estimations. Here, CIE 1976 and CIE 2000 outperformed the other color-differences (see also Table 6.11).

In relation to the $MSE$ and $RMSE$, the smaller their value, the better the results are. Again, the worst results are for the gray scale and $L^*$ differences, followed by the RGB Euclidean distance. The best results once more occur when perceptual color differences are used, with CIE 1976 and CIE 2000. The best individual kernel performances for Figure 6.1 (a) were $\mathcal{K}$ ($RMSE \approx 2.89$, CIE 1976), $K_u$($RMSE \approx 3.02$, CIE 2000), $K_g$($RMSE \approx 3.1$, CIE 2000) and $K_s$($RMSE \approx 3.3$, CIE 2000). Conversely, Figure 6.1 (b) results were obtained using CIE 1976 for all kernels in the following sequence: $K_s$($RMSE \approx 2.72$), $K_g$($RMSE \approx 2.78$), $K_u$($RMSE \approx 2.85$) and $\mathcal{K}$ ($RMSE \approx 3.24$) pixels. Finally, the results for Figure 6.1 (c) were obtained using CIE 2000 with $K_s$($RMSE \approx 2.64$), $K_u$($RMSE \approx 2.79$), $K_g$($RMSE \approx 2.98$), and $\mathcal{K}$

Table 6.10: Ground Truth and Estimation data using different kernels and color-differences for Figure 6.1 (c)

| | **Ground Truth** | **CIELAB L*** | | | | **CMC** | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | **Ku** | **Ks** | **k** | **Kg** | **Ku** | **Ks** | **k** | **Kg** |
| **Mean** | 7.86 | 12.33 | 11.10 | 13.06 | 11.13 | 9.25 | 8.25 | 9.07 | 8.82 |
| **Variance** | 25.07 | 33.64 | 30.02 | 34.51 | 31.99 | 25.87 | 22.70 | 23.43 | 24.38 |
| **Std. Dev.** | 5.01 | 5.80 | 5.48 | 5.87 | 5.66 | 5.09 | 4.76 | 4.84 | 4.94 |
| **Max** | 26.00 | 26.21 | 24.47 | 26.33 | 25.35 | 26.08 | 23.71 | 26.13 | 25.28 |
| **Min** | 2.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| | **Ground Truth** | **CIELAB 1976** | | | | **CIELAB 1994** | | | |
| | | **Ku** | **Ks** | **k** | **Kg** | **Ku** | **Ks** | **k** | **Kg** |
| **Mean** | 7.86 | 7.99 | 7.13 | 7.82 | 7.99 | 9.09 | 8.07 | 8.90 | 8.72 |
| **Variance** | 25.07 | 24.01 | 20.87 | 22.22 | 25.15 | 25.41 | 22.29 | 23.13 | 24.15 |
| **Std. Dev.** | 5.01 | 4.90 | 4.57 | 4.71 | 5.02 | 5.04 | 4.72 | 4.81 | 4.91 |
| **Max** | 26.00 | 25.84 | 21.95 | 26.45 | 25.27 | 25.45 | 21.61 | 25.77 | 25.41 |
| **Min** | 2.00 | 0.77 | 0.61 | 0.00 | 0.00 | 0.00 | 0.00 | 0.47 | 0.00 |
| | **Ground Truth** | **CIELAB 2000** | | | | **Grayscale** | | | |
| | | **Ku** | **Ks** | **k** | **Kg** | **Ku** | **Ks** | **k** | **Kg** |
| **Mean** | 7.86 | 8.43 | 7.53 | 8.30 | 8.42 | 12.20 | 10.08 | 12.96 | 11.09 |
| **Variance** | 25.07 | 23.64 | 20.82 | 21.66 | 24.84 | 33.56 | 37.46 | 34.92 | 31.84 |
| **Std. Dev**. | 5.01 | 4.86 | 4.56 | 4.65 | 4.98 | 5.79 | 6.12 | 5.91 | 5.64 |
| **Max** | 26.00 | 24.17 | 20.60 | 25.44 | 25.36 | 26.20 | 21.74 | 26.50 | 25.35 |
| **Min** | 2.00 | 0.27 | 0.21 | 0.49 | 0.00 | 0.00 | 0.30 | 0.00 | 0.00 |
| | **Ground Truth** | **RGB Distance** | | | | | | | |
| | | **Ku** | **Ks** | **k** | **Kg** | | | | |
| **Mean** | 7.86 | 9.07 | 8.06 | 8.87 | 8.50 | | | | |
| **Variance** | 25.07 | 27.63 | 24.30 | 25.16 | 24.57 | | | | |
| **Std. Dev**. | 5.01 | 5.26 | 4.93 | 5.02 | 4.96 | | | | |
| **Max** | 26.00 | 26.14 | 23.74 | 26.49 | 25.15 | | | | |
| **Min** | 2.00 | 0.00 | 0.00 | 0.00 | 0.00 | | | | |

Table 6.11: Best Color-Difference Results for Figures 6.1 (a), (b) and (c)

| **Statistic** | **Figure 6.1 (a)** | **Figure 6.1 (b)** | **Figure 6.1 (c)** |
|---|---|---|---|
| $\rho$ | CIE2000(.79) | CIE1976(0.89) | CIE2000(0.77) |
| $MSE$ | CIE1976(8.35) | CIE1976(7.39) | CIE2000(6.95) |
| $RMSE$ | CIE1976(2.89) | CIE1976(2.72) | CIE2000(2.64) |
| $N_{RMSE}$ | CIE1976,1994&2000(.12) | CIE1976(0.14) | CIE2000(0.11) |
| $\xi_{abs}$ | CIE1976(2.12) | CIE1976(1.93) | CIE2000(1.61) |
| $M$ | CIE1976(1.56) | CIE1976(1.35) | CIE2000(0.72) |
| $\tau$ | CIE1976(2.70) | CIE1976(2.56) | CIE2000(1.72) |
| $KS(5\%)$ | - | - | CIE1976(0.83) & RGB(1.39) |
| $KS(1\%)$ | - | CMC & CIE1976(1.57) | CIE1976,RGB & 2000(1.62) |
| $KS(.1\%)$ | - | CMC & CIE1976(1.57) | RGB,CIE1976,2000 & 1994(1.79) |
| $BC$ | CIE1994(0.92) | CMC & CIE1976(0.94) | CMC, CIE1976 & 2000(0.98) |
| $BD$ | CIE1994(0.09) | CMC & CIE1976(0.06) | CMC, CIE1976 & 2000(0.02) |
| $HD$ | CIE1994(0.29) | CIE1976(0.24) | CIE1976(0.13) |
| $Hits(10\%)$ | CIE1976(49.8%) | CIE1976(25%) | CIE2000(46.1%) |
| $Hits(20\%)$ | CIE1976(75.6%) | CIE1976(47.1%) | CIE2000(70.1%) |
| $Hits(30\%)$ | CIE1994(85.9%) | CIE1976(61.9%) | CIE1976(79.8%) |

Table 6.12: Best Kernel Results for Figures 6.1 (a), (b) and (c)

| **Statistic** | **Figure 6.1 (a)** | **Figure 6.1 (b)** | **Figure 6.1 (c)** |
|---|---|---|---|
| $\rho$ | $K_g$ (0.8) | $K_s$ & $K_g$ (0.89) | $K_u$ & $K_s$ (0.85) |
| $MSE$ | $\mathcal{K}$ (8.35) | $K_s$ (7.39) | $K_s$ (6.95) |
| $RMSE$ | $\mathcal{K}$ (2.89) | $K_s$ (2.72) | $K_s$ (2.64) |
| $N_{RMSE}$ | $\mathcal{K}$, $K_g$ & $K_u$ (0.12) | $K_u$ & $K_g$ (0.14) | $K_s$ (0.11) |
| $\xi_{abs}$ | $\mathcal{K}$ (2.12) | $K_s$ (1.93) | $K_u$ (1.61) |
| $M$ | $\mathcal{K}$ (1.56) | $K_u$ (1.35) | $K_g$ (0.72) |
| $\tau$ | $\mathcal{K}$ (2.70) | $K_g$ (2.57) | $K_u$ & $K_g$ (1.72) |
| $KS(5\%)$ | - | - | $\mathcal{K}$, $K_g$ & $K_u$ (0.88) |
| $KS(1\%)$ | - | $K_s$ & $K_u$ (1.57) | $\mathcal{K}$, $K_g$ & $K_u$ 0.88), $K_s$ (1.39) |
| $KS(.1\%)$ | - | $K_u$, $K_s$ (1.57) & $K_g$ (1.70) | $\mathcal{K}$, $K_g$ & $K_u$ 0.88), $K_s$ (1.39) |
| $BC$ | $\mathcal{K}$ (0.92) | $K_g$ & $K_s$ (0.94) | $\mathcal{K}$, $K_s$, $K_g$ & $K_u$ (.98) |
| $BD$ | $\mathcal{K}$ (0.09) | $K_g$ & $K_s$ (0.11) | $\mathcal{K}$, $K_s$, $K_g$ & $K_u$ (.02) |
| $HD$ | $\mathcal{K}$ (0.29) | $K_s$ (0.24) | $K_u$ (0.13) |
| $Hits(< 10\%)$ | $\mathcal{K}$ (49.8%) | $K_g$ & $\mathcal{K}$ (25%) | $K_u$ (46.1%) |
| $Hits(< 20\%)$ | $\mathcal{K}$ (75.6%) | $K_g$ (47.1%) | $K_g$ (70.1%) |
| $Hits(< 30\%)$ | $\mathcal{K}$ (85.9%) | $K_s$ (61.9%) | $K_g$ (79.8%) |

($RMSE \approx 3.32$) pixels. These results can be explained by the characteristics of each image. Figure 6.1 (a) is the one with straighter shape, favoring $\mathcal{K}$. Figure 6.1 (b) is difficult because the curvilinear feature does not have an uniform color across the line regions, spreading the density distributions among the colors. Figure 6.1 (c) has more curves, supporting monotonic decreasing kernels. This happens because at regions with high curvature the kernel considers much more pixels than it should, since it takes into consideration not only a small part of the curve, but the curve as a whole. That is, the curve fills almost completely the entire kernel region. As the $\mathcal{K}$ is monotonically increasing, the contribution of pixels next to the limits/borders of $\mathcal{K}$ is high. This results in an undesired increase of the density estimate and, consequently, in the obtained width estimate. This does not happen with $K_u$, $K_g$ and $K_s$, since those pixels have small weight to contribute to the density estimate.

The $N_{RMSE}$ indicates the amount of residual variance, which shows how the observed measurements will scatter around the $RMSE$ in the case of a normal distribution. However, the line width distribution does not necessarily follow a normal distribution. We again observe less scattering for perceptual color-differences with the best results for CIE 1976 and CIE 2000. Figure 6.1(a) presented the smallest values of $N_{RMSE} \approx 12\%$ with $K_g$ (CIE 1976), $\mathcal{K}$ (CIE 1976 and 2000) and $K_u$ (CIE 1994), followed by $K_s$ using CIE 1994 with $N_{RMSE} = 14.3\%$. Figure 6.1(b) was particularly hard, because of the large variation of the line color. However, all the individual kernels obtained the best results using CIE 1976, where $K_u$, $K_g$, $K_s$ and $\mathcal{K}$ had $N_{RMSE}$ equal to 14.2%, 14.3%, 14.8% and 18.2%, respectively. Figure 6.1 (c) reveals the following best results obtained using CIE 2000, where $K_s$, $K_u$, $K_g$ and $\mathcal{K}$ with 11.0%, $\approx 12\%$, $\approx 12\%$ and 14%, respectively. By and large, we see that the kernel performance is influenced by the features of the images analyzed. For instance, the results for Figure 6.1 (a) are affected to the noise in the image. The results for Figure 6.1 (b) were affected by the river banks with different colors, as we have already mentioned. Figure 6.1 (c) has shown the best results using CIE 2000, since the river is blue and CIE 2000 has an underlying ability to deal with bluish colors (LUO, 2002), lacking in previous perceptual color differences. However, it is impressive how the CIE 1976 formula, proposed 24 years before CIE 2000, presents such consistent results even today.

Thereafter, we analyze the absolute mean error, $\xi_{abs}$, measured in pixels. Notably, $\xi_{abs}$ reinforces the previous results, with best scores for CIE 1976 and CIE 2000, followed by the other perceptual color-differences analyzed in this work. RGB and intensity differences showed the worst performances. At this time, for Figure 6.1 (a), we have $\mathcal{K}$ with $\xi_{abs} = 1.56$ and $K_s$ with $\xi_{abs} = 1.82$, both using CIE 1976, while $K_u$ and $K_g$ had $\xi_{abs} = 1.87$ and $\xi_{abs} = 2.06$, respectively, both using CIE 2000. Then, for Figure 6.1 (b), $K_s$, $K_g$, $K_u$ and $\mathcal{K}$ had $\xi_{abs} = 1.93$, $\xi_{abs} = 2.01$, $\xi_{abs} = 2.05$ and $\xi_{abs} = 2.3$, respectively, all using CIE 1976. Lastly, Figure 6.1 (c) presents the best results, with $K_u$ and $K_g$ having $\xi_{abs} = 1.61$ and $\xi_{abs} = 1.62$, respectively, both using CIE 2000. While, $K_s$ had $\xi_{abs} = 1.65$ using CIE 1994 and $\mathcal{K}$ had $\xi_{abs} = 1.91$ using CIE 2000.

Using the median and percentile at 75%, both measured in pixels, CIE color-differences show the best results. At this time, we can see how the error is distributed across the samples, since the median and the percentile at 75% display the error values ordered from the smallest to the largest error at specific points of the distribution. This means that, if the median is small, half of the errors is less or equal to the median. In the same way, if the percentile error at 75% is small, 75% of the errors is less or equal to the percentile error. The combined analysis is also important, because we can see if the error does not

grow drastically from the median to the given percentile. The results for Figure 6.1 (a) showed that the best medians for $\mathcal{K}$(using CIE 1976), $K_s$ (using CIE 1994), $K_u$ (using CIE 2000) and $K_g$ (using CIE 2000) were 1.56, 1.82, 1.87 and 2.06 pixels, respectively, while $\mathcal{K}$, $K_s$, $K_u$ and $K_g$ had the Percentile (75%) equals to 2.70, 2.95, 3.11 and 3.27 pixels, respectively, considering the color-differences used previously for medians. For Figure 6.1 (b), the best median and percentile results were obtained both using CIE 1976. The medians were equal to 1.35, 1.36, 1.36 and 1.45 for $K_u$, $K_s$, $K_g$ and $\mathcal{K}$, respectively, and the Percentile were equal to 2.69, 2.57, 2.73 and 3.09 pixels for the same sequence kernel. For Figure 6.1 (c) the median values (using CIE 2000) were $K_g$, $K_u$, $K_s$ and $\mathcal{K}$ with 0.72, 0.75, 0.77 and 0.78 pixels, respectively, while the Percentile results were $K_u$ and $K_g$ (both using CIE 2000) with 1.72 pixels, followed by $K_s$ (using RGB Euclidean distance) with 1.78 pixels, and $\mathcal{K}$ (using CIE 2000) with 2.03 pixels. These results are important, because they show that for an image with wide lines such as Figure 6.1 (a), the error for 75% of the samples can be smaller than 3 pixels. Also, Figure 6.1 (b), with non-uniform color lines, showed a similar values, however the mean width of the line is half of the one from Figure 6.1 (a). Hence these results are not as good as the previous one, with 75% of the samples with errors that are less or equal to 2.57 in the best case. Figure 6.1 (c) presented the best results, with medians with errors smaller than a pixel, and percentile errors of up to two pixels. We see that the proposed CWLD estimate provides useful results to estimate lines.

The Kolgomorov-Smirnov Hypothesis results are provided with three significance levels (5.0%, 1,0%and 0,1%) to determine $K_\alpha$, which are used to accept or reject $H_0$, according to Equations 6.9 and 6.10. For the significance levels 5%, 1% and 0.1%, $K_\alpha$ is equal to 1.36, 1.63 and 1.95, respectively. For Figure 6.1 (a), $H_0$ was rejected considering all significance levels, where the best result was obtained using $\mathcal{K}$ using CIE 2000, i.e., $KS = 2.3$, which is above the threshold needed by any significance level to accept $H_0$. For Figure 6.1 (b), $H_0$ was accepted only with significance levels of $1\%$ and $0.1\%$ using $K_g$ and $K_u$ with CIE 1976 (both with $KS = 1.57$). Still, in Figure 6.1 (b), $H_0$ was accepted with significance level of $0.1\%$ using $K_g$ with CMC 1984 ($KS = 1.70$); and $K_u$ ($KS = 1.82$) and $K_g$ ($KS = 1.89$), both with CIE 2000. In Figure 6.1 (c), $H_0$ was accepted for all the three significance levels with $K_u$, $K_g$ or $\mathcal{K}$ using CIE 1976, and also for $K_s$ with the RGB Euclidean distance. When using $K_s$, $H_0$ was also accepted for the significance levels of $1\%$ and $0.1\%$ for CIE 2000 ($KS = 1.62$), and also for the significance level of $0.1\%$ using CIE 1994 ($KS = 1.79$). RGB presents good results sometimes for some figures and kernels, while the results produced by perceptual color-differences are more consistent. In some of the tests, the CWLD estimation could be considered very similar to the ground-truth. Even in Figure 6.1 (a), where the test almost passed, probably due to the noise. In a nutshell, the results shows that the CWLD is probably similar to the ground truth, depending on the image quality.

Regarding the Bhattacharrya Coefficient, we can see that the best values, i.e. $BC \approx 1$, are those from the perceptual color-differences for the three images. The distances of Bhattacharrya and Hellinger reflect the same behavior, since they are based on the Bhattacharrya coefficient. However, for these measures, the closer they are from zero, the better the results are. Besides, if we analyze the Bhattacharrya Coefficient alone or the Bhattacharrya Distance, the results are not enough for comparison purposes, since they are very similar. On the other hand, the Hellinger distance results are more visible than those of the Bhattacharrya Coefficient or the Bhattacharrya Distance. For instance, $\mathcal{K}$ shows the best results (using with CIE 1994) with $HD = 0.29$ for Figure 6.1 (a), followed by all the

other kernels with $HD \geq 0.33$ using different perceptual color-differences. For Figure 6.1 (b), $K_s$ obtained $HD = 0.24$ (using CIE 1976), $K_g$ obtained $HD = 0.25$ (using CMC 1984), $K_u$ obtained $HD = 0.26$ (using CIE 1976) and $\mathcal{K}$ obtained $HD = 0.31$ (using CIE 1976). For Figure 6.1 (c), $K_u$ and $K_g$ obtained $HD = 0.13$ and $HD = 0.14$, respectively, both using with CIE 1976. While $K_s$ with CMC and $\mathcal{K}$ with CIE 2000, both obtained $HD = 0.15$. Comaniciu et al. (COMANICIU; RAMESH; MEER, 2000) have shown that the Hellinger distance is closely associated with the $MSE$ and also that it is a metric. Probably, that is the reason we could see the results more clearly through the Hellinger Distance than with the Bhattacharrya Coefficient or the Bhattacharrya Distance. Results confirm the scores of perceptual color-differences and also the same tendency of previous results for the kernels, with $\mathcal{K}$ showing the best results for Figure 6.1 (a), while the other kernels are slightly better than $\mathcal{K}$ with the other Figures. We should mention that these statistics are non-parametric. That is, if the probability distributions are the same, but the measurements are in different orders, i.e., the widths are in different positions in the line, the results may be misleading.

In relation to the Hit counts measure, we use $10\%$, $20\%$ and $30\%$ of acceptance. As it is solely based on the relative error (see Section 6.2), if a line has width of ten pixels, only 1 pixel of error is acceptable at 10% acceptance. This is a rigorous test for the thin lines, since it approaches the image discretization limits. This is the case for the mean width (see Tables 6.2 and 6.3) of the measured lines from Figures 6.1 (b) and (c), but not for Figure 6.1 (a). Figure 6.1 (a) presented the best results, since the mean width is wider than the other two Figures (see Table 6.10). For this Figure, $\mathcal{K}$ (using CIE 1976) obtained 49.8%, 75.6% and 85.8%; $K_s$ (using CIE 1994) obtained 44.1%, 70.6% and 84.1%; $K_u$ obtained 41.1%, 70.3% and 85.6%; and $K_g$ obtained 35.1%, 68.7% and 85.2%, all of the them consider acceptance rates of 10%, 20% and 30% respectively. Figure 6.1 (b) presented slightly inferior results, using CIE 1976 for all the kernels, where $K_s$ obtained 24.2%, 44.3% and 61.9%; $K_u$ obtained 24%, 46.9% and 60%; $K_g$ obtained 25%, 47.1% and 58.4% for acceptances of 10%, 20% and 30%; and $\mathcal{K}$ obtained 25%, 41% and 56.6%, all of the them consider acceptance rates of 10%, 20% and 30% respectively. For Figure 6.1 (c) the best results were different among each acceptance level. The best result for the acceptance rate of 10% were obtained by $K_u$ with 46.1%, $K_g$ with 45.2% (both using CIE 2000), $K_s$ with 45,1% (using CIE 1994) and $\mathcal{K}$ with 42.9 (using CIE 2000). For the same figure, the best results for the acceptance rate of 20% were obtained using $K_g$ with 70.1% (both using CIE 2000), $K_u$ with 69.2%, $K_s$ with 68.6% (using CIE 1994) and $\mathcal{K}$ with 64.1% (using CIE 2000). Still in Figure (b), the best results for the acceptance rate of 30% were obtained using $K_g$ with 79.8% (both using CIE 2000), $K_s$ with 79.5% (using CIE 1994), $K_u$ with 77.6% and $\mathcal{K}$ with 76.1% (using CIE 2000). In summary, the best results attest the good performance of CIE perceptual color differences. Figure 6.1 (a) has wider lines than the other images, thus the good results were already expected. Figure 6.1 (b) was the most challenging to the algorithm, because it has some long thin lines and the color problem, discussed previously, that led to the weak results. The best results of Figure 6.1 (c) were obtained using CIE 2000, due to its bluish color, as mentioned previously in this section. However, the thin lines combined with the large number of curves in this image contribute to the reasonable result when compared to Figure 6.1 (a).

Lastly, Figures 6.5, 6.6 and 6.7 present the Error distributions for Figures 6.1 (a), (b) and (c), respectively. For each figure, (a)-(g) show the results obtained using CMC 1984, CIE 1976, CIE 1994, CIE 2000, RGB Euclidean distance, gray scale difference

and $L^*$ differences, respectively. Each chart displays the results using $K_u$, $K_g$, $\mathcal{K}$ and $K_s$. These figures only reinforce the statistical results, where we can see bad performances for the RGB, grayscale and $L^*$ color-differences in Figures 6.5 (e)-(g); in Figure 6.6 (e)-(g); and in Figure 6.7 (e)-(g). They also show the good performances for perceptual color-differences, mainly for CIE 2000 and CIE 1976. These figures also show that there is a small amount of errors with high magnitude, but the error distribution in pixels does not provide enough to visualize the locations of these errors. Figures 6.9 (a)-(c) show the errors along the line using CIE 1976 for Figures 6.1 (a)-(c), respectively. Figure 6.8 (a)-(c) show the same data, but using CIE 2000. We point out these two color-differences, CIE 2000 and CIE 1976, because they produced the best results in the previous tests. Figures 6.9 and Figure 6.8 display line portions with the largest errors for the three images. They are present mainly in the regions of high curvature and those where the color-difference was not enough to separate the line from the background. These errors permit to identify the two constraints of the current version of the CWLD algorithm. First, it is not suited for lines with regions of high curvature features. It has better results in images that do not have lines with sharp angles. Special treatments are required to correct the density estimate in those regions. Finally, we cannot calculate the line width appropriately if the color-difference and the CWLD threshold are not adequate to the given image, i.e., the method will fail. Observe that we have chosen natural images because they illustrate these conditions. Figure 6.1 (a) has considerable compression noise; Figure 6.1 (b) has lines that are not easily separated from the background using colors; and Figure 6.1 (c) has several regions with lines of high curvature.

## 6.7 Limitations

Overall we have shown the importance of color-differences when compared to gray scale images, however we point out that with bad illumination conditions, color differences do not work properly. Furthermore, line detection is poor depending on the background, and further changes should be made to the algorithm to improve its line detection results.

Regarding width estimates, we highlight that algorithm does not perform well when the structures are too curved, since it deforms the underlying density estimate. Similarly, the quality of the color-difference equation also affects the method, along with illumination conditions as above. Also, the pixels at the beginning and end of line structures are also not measured correctly since the kernel is not completely crossed by the structure, cutting out part of the density estimate used by the method to determine the line thickness. Further development is required to deal with these conditions and improve the thickness estimates. Furthermore, the proposed kernel $\mathcal{K}$ suffers from sampling problems at the edges as the radius gets smaller, this is expected since the higher weights of the kernel is at its extremities.

Finally, the method can be computationally costly when applied to all pixels in the image. Still, it is efficient when it is applied for estimation of other higher level information, such as pose estimation or the river width at specific positions, desired pixels of an image. This is where the advantage of the method lies. The algorithm can obtain the thickness (or width) estimate of arbitrary pixel positions using a non-parametric strategy on-the-fly.

Figure 6.5: Error Distribution for Figure 6.1 (a). The color differences shown are the CIE 1976 (a); CMC 1984 (b), CIE 1994 (c); CIE 2000 (d); RGB distance (e); gray-scale difference,i.e., the WLD (f); and CIE $L^*$ difference (g). Observe that the CIE and CMC color differences perform better than 1D based differences.

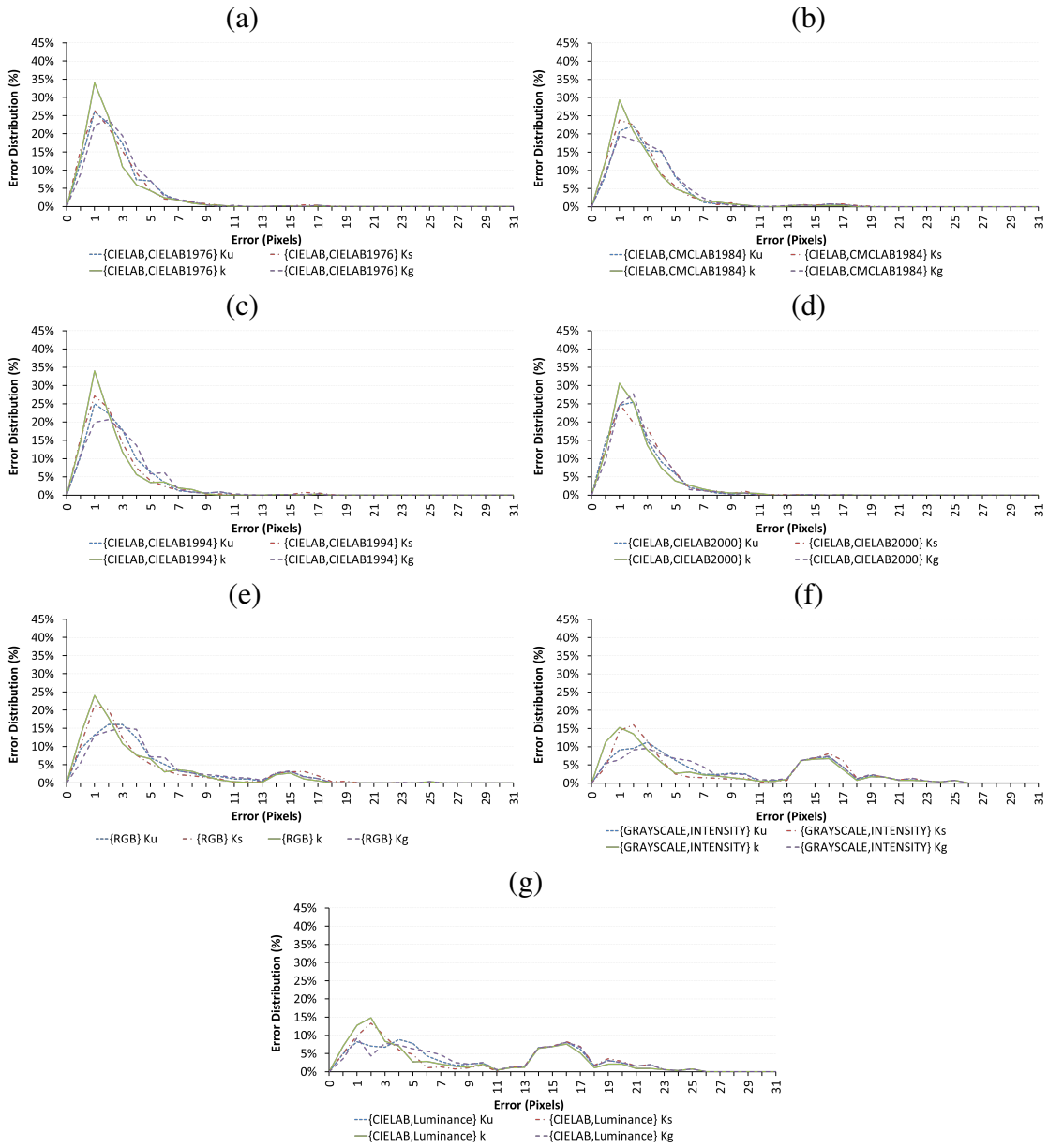


Figure 6.6: Error Distribution for Figure 6.1 (b). The color differences shown are the CIE 1976 (a); CMC 1984 (b), CIE 1994 (c); CIE 2000 (d); RGB distance (e); gray-scale difference,i.e., the WLD (f); and CIE $L^*$ difference (g). Observe that the CIE and CMC color differences perform better than 1D based differences.

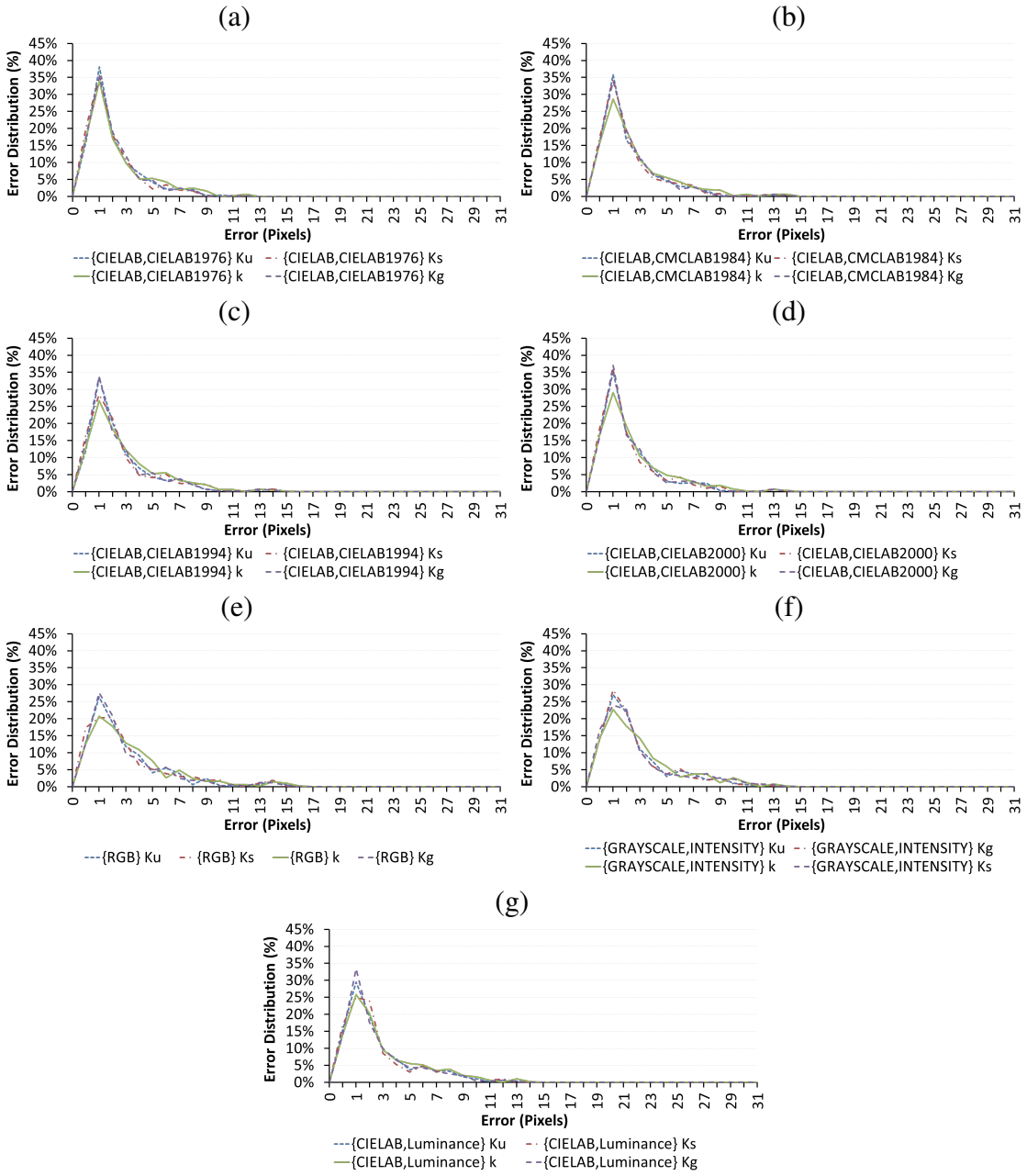


Figure 6.7: Error Distribution for Figure 6.1 (c). The color differences shown are the CIE 1976 (a); CMC 1984 (b), CIE 1994 (c); CIE 2000 (d); RGB distance (e); gray-scale difference,i.e., the WLD (f); and CIE $L^*$ difference (g). Observe that the CIE and CMC color differences perform better than 1D based differences.

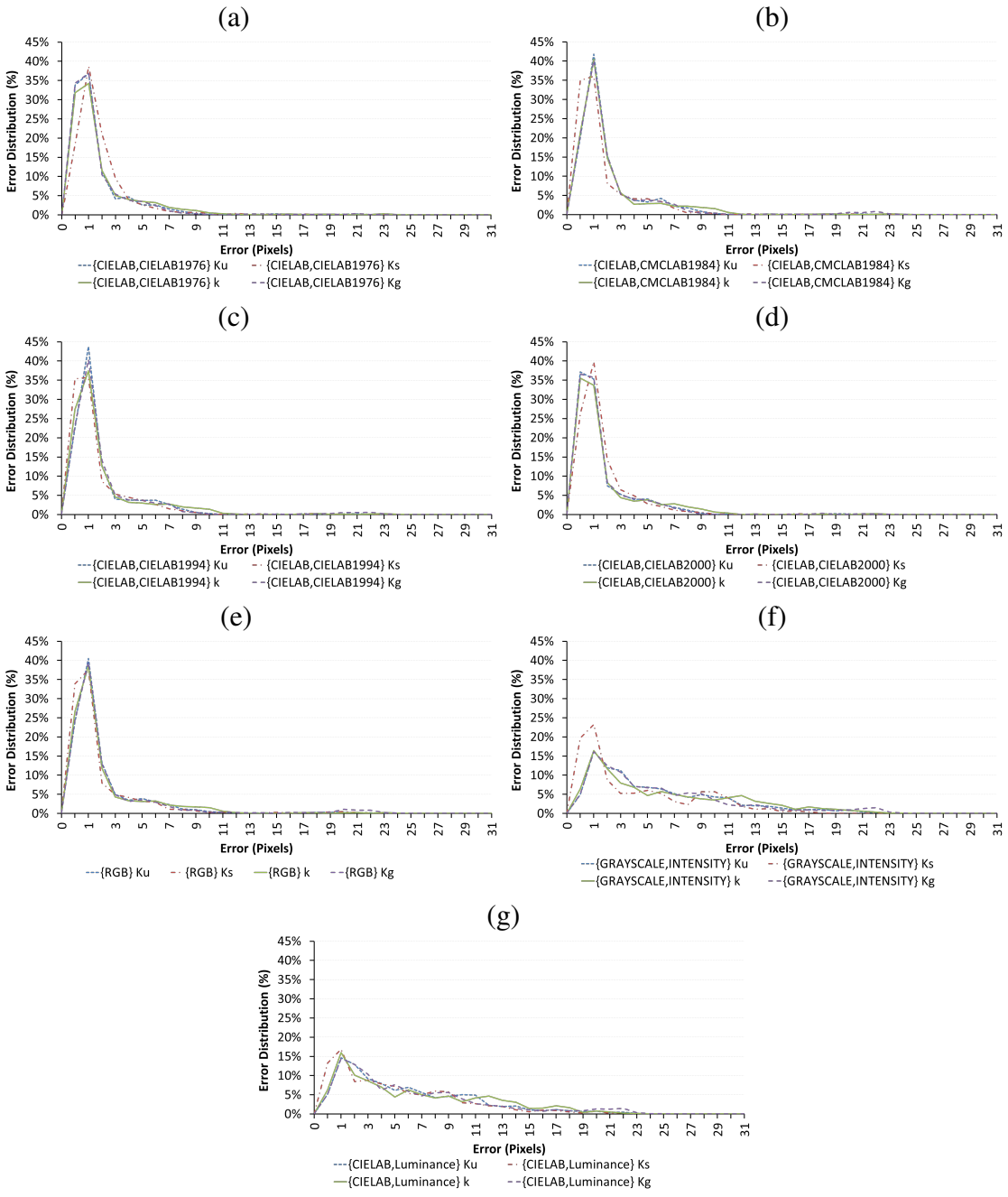
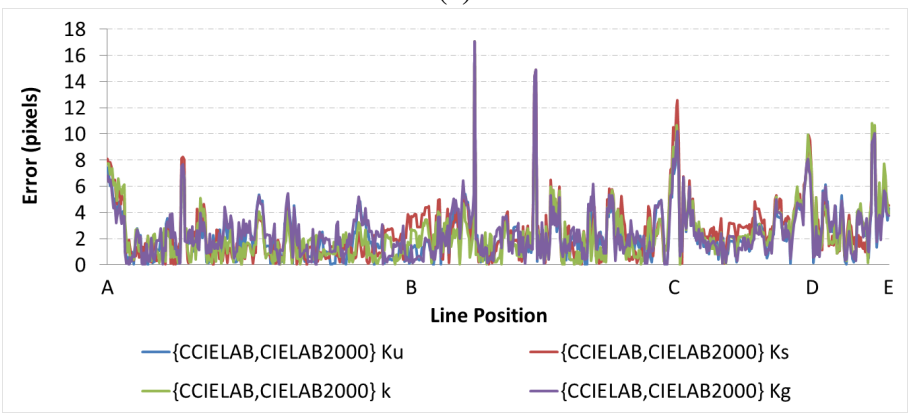
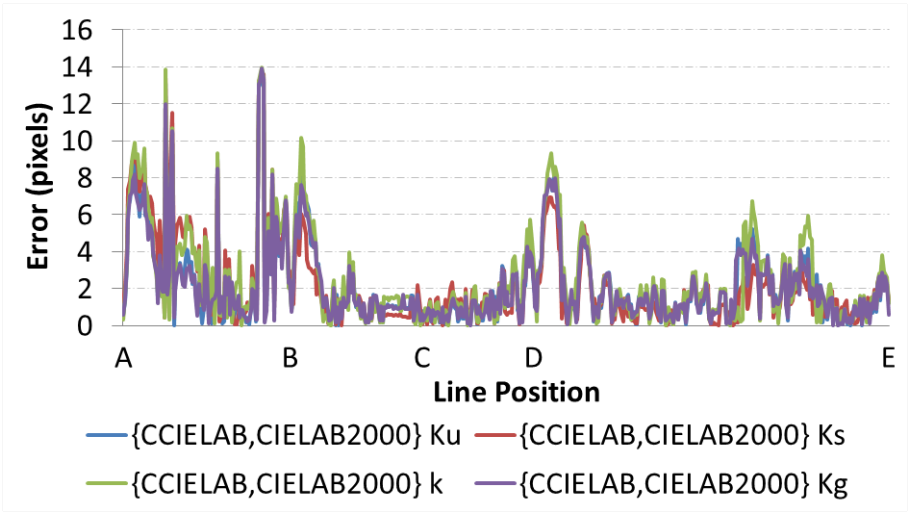


Figure 6.8: The absolute error at each pixel for Figures 6.1 (a), (b) and (c) considering the CIE 2000 color-difference. Observe that regions next to curves present the high incidence of errors for all the images, while regions with low curvature present small errors.

(a)




(b)




(c)

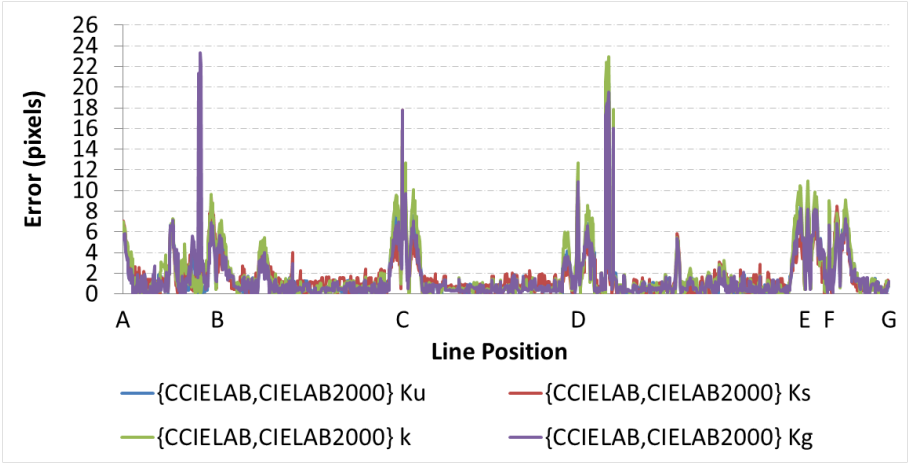
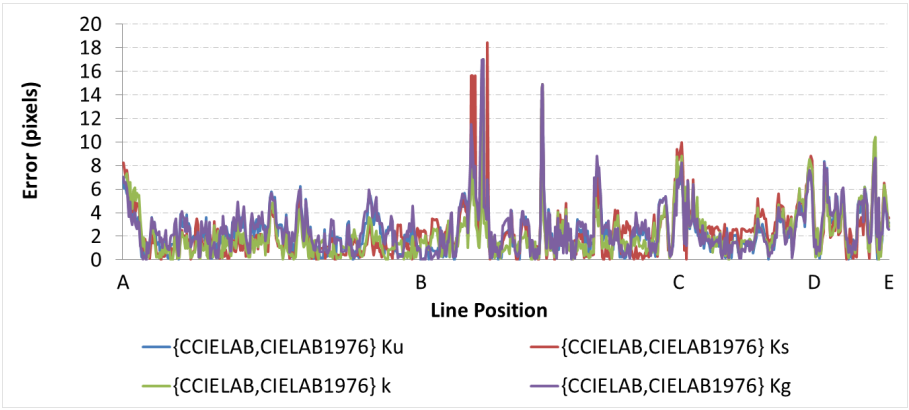
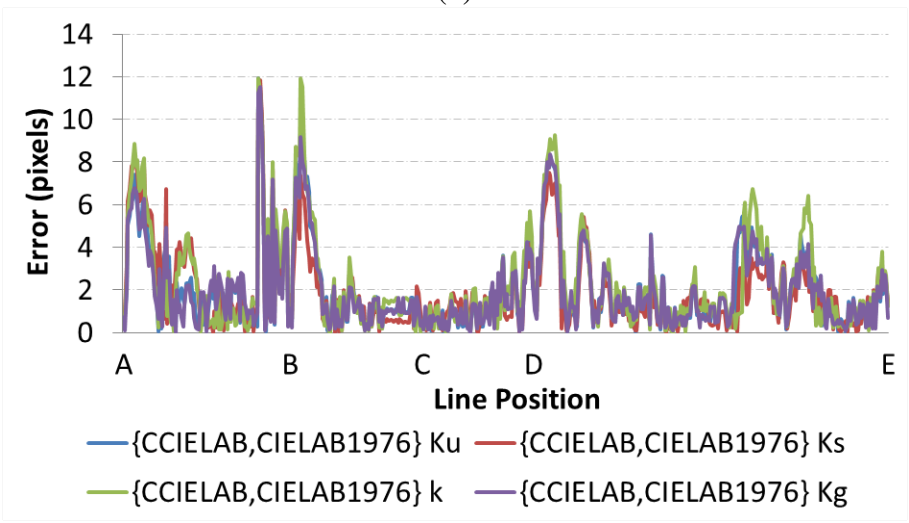


Figure 6.9: The absolute error at each pixel for Figures 6.1 (a), (b) and (c) considering the CIE 1976 color-difference. Observe that regions next to curves present the high incidence of errors for all the images, while regions with low curvature present small errors. Furthermore, the results are very similar to Figure 6.8.

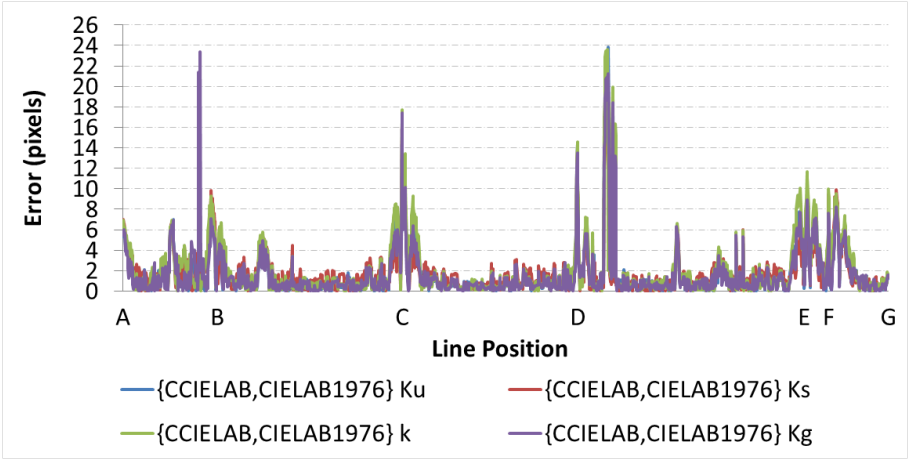
(a)




(b)




(c)



# 7 DISCUSSIONS AND FUTURE WORK

In this Work, we proposed a new method for detecting wide lines in color images. We have shown the robustness of perceptual color differences when compared to a trivial gray scale difference to detect line pixels. We also developed a new kernel that is more efficient than the previous ones to detect line pixels. Experiments confirmed that, on average, the proposed kernel $\mathcal{K}$ detects widelines with smaller kernel radius than $K_u$, $K_g$ and $K_s$. Finally we have developed a new method to obtain the line width, using the density estimate. We tested it with several combinations of kernels and color-differences using different statistical metrics.

The use of different statistical measures is one of the contributions of this Dissertation, applied to assess the effectiveness the method along with the different color-difference equations, and color spaces. We show that the effectiveness of each statistical measure varies. In particular, we could see the results more clearly through the Hellinger Distance than with the Bhattacharrya Coefficient or the Bhattacharrya Distance. This confirms previous findings showng that the Hellinger distance (metric) is a powerful statistical tool, closely associated with the $MSE$. In summary, comparing distributions should be done carefully, since not all of them fit all situations, but they can be useful if the metric is chosen appropriately.

Specifically, analyzing all the statistics available, the perceptual color-differences offered a better density estimate and, as a result, a better width measurement and line detection. In particular, we can see that both CIE 1976 and CIE 2000 consistently presented the best results. CIE 2000 presented the best results for Figure 6.1 (c), where a bluish line is tested. This is not surprising, since this color-difference equation was designed to correct errors in the blue centers present in all the previous formulas. The WLD and the CWLD using $L^*$ presented worse results than the CWLD using color vectors for measuring the line width and line detection. Notably the RGB Euclidean distance was better than the monochromatic differences, but worse than perceptual color differences. However, when the CWLD uses the CIE color-differences, it can be affected by the illumination conditions.

Still, this work shows that color information can indeed be superior to gray scale images, but it also have shown that the behavior of each color-difference equation is not stable across images, what opens space to future improvements of such equations with computer vision in mind. The evolution of color-difference equations and the development of color appearance models will lead to new and enhanced ways to determine the color-difference. Nevertheless, the CWLD can easily take advantage of new perceptually uniform color spaces and new color-difference equations. Even though several kernels and color-differences tested in this work can provide visually similar local density estimates, the resulting values may vary a lot more than one might expect. Moreover, we

have adapted the computation of the brightness contrast threshold in the WLD, for the CWLD, resulting in an intuitive relative metric to analyze the calculated threshold. Our tests have shown that there is much room to improve this threshold. Figure 6.1 (b) is a good example of that, where we need a better criteria to compare the lines.

We also developed and confirmed with experiments that it is possible to obtain a local width estimation (pixel-wise) using only the kernel density data, for each of the kernels studied. However, our method presented the smallest errors with somewhat straight features, whereas the errors were large high curvature regions. We have shown that $K_u$ and $K_g$ sometimes can produce slightly better measurements than $\mathcal{K}$, but at the cost of using a bigger kernel radius. Despite that, all the kernels performed reasonably well estimating the line width at each pixel. The proposed kernel seems to be slightly more affected by noise (compression noise) and curvature interference than the monotonically decreasing kernels, but much less than we suspected. Furthermore, the CWLD in many ways can be seen as a modified medial axis transformation, but instead of marking the center line, it gives the line width at all line pixels. This eliminates the need to accurately detect the center line and edges, which is important mainly when a simple local estimate is required (i.e., test one pixel alone. e.g., in the center of an image while using a UAV camera).

The tests using $K_u$ and $K_g$ with varying radius showed that, insofar as the kernel radius increases, the line pixels are gradually marked from the borders to the center line (see Figure 5.3 in Section 5.3). On the other hand,increasing the kernel radius, $\mathcal{K}$ marks the line pixels mostly in a uniform way, sometimes going from the center line towards the borders much faster than $K_u$ and $K_g$ (see Figure 5.3 in Section 5.3). These two behaviors could be combined to improve detection, diminishing the kernel size at the same time. In addition, with the aid of morphologic algorithms, such as region growing, it is possible to mark line pixels faster or even correct detection errors. Also, $K_g$ results seem to result understimates the actual line thickmess, but measurements using such kernel seem more linear than the other ones with respect to orientation. Still, a scale parameter could be defined to try to correct such errors, but further tests would be required to test the effectiveness of such changes.

The algorithm has shown considerable amount of false positives mainly due to the fact that this is a density based approach. That is, the algorithm sets every low density pixel to a line pixel. Removing small features, i.e. false positives, should be done carefully, since it might lead to the removal of valid line pixels as well, generating holes in segmented lines or removing them completely. Another strategy used by the WLD is the removal of groups of pixels with eccentricity greater than a specific threshold (LIU; ZHANG; YOU, 2007). We plan to investigate these and other pre and post processing methods to minimize false detection and maximize the correct segmentation.

The construction of $\mathcal{K}$ was based on a single condition, i.e., $V_C = V_E = V_K/2$, and even though the results are quite robust to decrease the kernel size. In the future, we intend to investigate other conditions and functions to create a more efficient and precise kernel for width estimation and line detection. As the CWLD is also based on bar-shaped lines, the width measurement could be improved further. This could be achieved by incorporating more information, such as curvature, to the algorithm. It would be intriguing to extend the algorithm to 3D datasets separating surfaces automatically using the same principle of the CWLD/WLD.

Finally, we might use the CWLD width information together with the output of other line detection algorithms to develop new image compression algorithms or even a new background removal algorithm, leading to other branches of investigation. It can be

relevant for image segmentation, feature extraction, non-parametric tracking and other computer vision tasks, particularly to non-parametric methods such as Mean-Shift (COMANICIU; RAMESH; MEER, 2000; COMANICIU; RAMESH, 2000). Line detection can useful for robot localization, however new algorithms should be devised to use thickness information.

# APPENDIX A COLOR CONVERSION

Given a **RGB** color in the normalized color space, it is possible to convert it to the **XYZ** color space applying the following transformation represented by Equation A.1:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = [M] \begin{bmatrix} r \\ g \\ b \end{bmatrix} \tag{A.1}$$

where $\begin{bmatrix} r & g & b \end{bmatrix}^{\tau}$ represents the tristimulus values. They are calculated from the given normalized input **RGB** vector, $\begin{bmatrix} R & G & B \end{bmatrix}^{\tau}$, typically through Equation A.2:

$$\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} R^{\gamma} \\ G^{\gamma} \\ B^{\gamma} \end{bmatrix} \tag{A.2}$$

where $^{\gamma}$ is used to approximate the behavior of the human vision, regarding each one of the tristimilus perception values.

The matrix $[M]$ represents the change from coordinate system and each value corresponds to the maximum value that each of the primaries of the given **RGB** color system can reach. It is important to say that $[M]$ is precomputed and can be obtained from the primaries of the given RGB color space: $\begin{bmatrix} X_r & Y_r & Z_r \end{bmatrix}^{\tau}$, $\begin{bmatrix} X_g & Y_g & Z_g \end{bmatrix}^{\tau}$ and $\begin{bmatrix} X_b & Y_b & Z_b \end{bmatrix}^{\tau}$, all of them represented in **XYZ** space according to Equations A.3 and A.4:

$$[M] = \begin{bmatrix} X_{r,max} & X_{g,max} & X_{b,max} \\ Y_{r,max} & Y_{g,max} & Y_{b,max} \\ Z_{r,max} & Z_{g,max} & Z_{b,max} \end{bmatrix} \tag{A.3}$$

where each corresponding value of Matrix A.3 is the maximum values that each of the components of the primaries can achieve with the given illuminant.

$$[M] = \begin{bmatrix} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{bmatrix} \begin{bmatrix} L_{r,max} & 0 & 0 \\ 0 & L_{g,max} & 0 \\ 0 & 0 & L_{b,max} \end{bmatrix} \tag{A.4}$$

The vector $\begin{bmatrix} L_{r,max} & L_{g,max} & L_{b,max} \end{bmatrix}^{\tau}$ in the diagonal matrix of Equation A.4 represent the maximum luminances for each primary of the given **RGB** coordinate system. To obtain them we must equal the white point, $\begin{bmatrix} X_W & Y_W & Z_W \end{bmatrix}^{\tau}$, and the color white in the **RGB** color space,

$\left[\begin{array}{ccc} 1 & 1 & 1 \end{array}\right]^{\tau}$ manipulating Equation A.1 using Equation A.4. This will result in Equations A.5 and A.6:

$$\left[\begin{array}{c} X_W \\ Y_W \\ Z_W \end{array}\right] = [M] \left[\begin{array}{c} 1 \\ 1 \\ 1 \end{array}\right] \tag{A.5}$$

obtaining:

$$\left[\begin{array}{c} X_W \\ Y_W \\ Z_W \end{array}\right] = \left[\begin{array}{ccc} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{array}\right] \left[\begin{array}{c} L_{r,max} \\ L_{g,max} \\ L_{b,max} \end{array}\right] \tag{A.6}$$

since in Equation A.6, the primaries and the whitepoint are known for the given RGB color space, the vector $\left[\begin{array}{ccc} L_{r,max} & L_{g,max} & L_{b,max} \end{array}\right]^{\tau}$ is given by Equation A.7:

$$\left[\begin{array}{c} L_{r,max} \\ L_{g,max} \\ L_{b,max} \end{array}\right] = \left[\begin{array}{ccc} X_r & X_g & X_b \\ Y_r & Y_g & Y_b \\ Z_r & Z_g & Z_b \end{array}\right]^{-1} \left[\begin{array}{c} X_W \\ Y_W \\ Z_W \end{array}\right] \tag{A.7}$$

this scales the values of each primary to its maximum value.

However, some devices use the rather different **sRGB** color space, in this case a slightly more complex function is used (see Equations A.8 and A.9):

$$\left[\begin{array}{c} r \\ g \\ b \end{array}\right] = \left[\begin{array}{c} \phi(R) \\ \phi(G) \\ \phi(B) \end{array}\right] \tag{A.8}$$

where $\phi(x)$ is represented by:

$$\phi(x) = \begin{cases} x/12.92321 & \text{, if } x \leq 0.0392857 \\ (\frac{x+0.055}{1.055})^{2.4} & \text{, otherwise} \end{cases} \tag{A.9}$$

Given a color, **(X,Y,Z)**, in the **(XYZ)** color space, its conversion to **CIE 1976 L*a*b*** in the Cartesian space is given by Equations A.10, A.11 and A.12 (GONZALES; WOODS, 2007):

$$L^* = 116h\left(\frac{Y}{Y_W}\right) - 16 \tag{A.10}$$

$$a^* = 500\left[h\left(\frac{X}{X_W}\right) - h\left(\frac{Y}{Y_W}\right)\right] \tag{A.11}$$

$$b^* = 200\left[h\left(\frac{Y}{Y_W}\right) - h\left(\frac{Z}{Z_W}\right)\right] \tag{A.12}$$

where $h\left(.\right)$ is the result of Equation A.13:

$$h(x) = \begin{cases} \sqrt[3]{x} & \text{, if } x > 0.008856 \\ 7.787x + 16/116 & \text{, otherwise} \end{cases} \tag{A.13}$$

This same color space can also be represented in as cylindrical coordinate system, the **CIE** $L^*C^*_{ab}h_{ab}$, that can be constructed using Equations A.14 and A.15.

$$C^*_{ab} = \sqrt{a^{*2} + b^{*2}} \tag{A.14}$$

$$h_{ab} = \arctan\left(a^*/b^*\right) \tag{A.15}$$

# APPENDIX B COLOR-DIFFERENCE

In this work we use color-difference equations with different results. A review of the research on color-difference is presented by Luo (LUO, 2002).

## B.1 RGB color-difference

Using the color components of a RGB image, we can compute the color-difference between two colors, $(R_1, G_1, B_1)$ and $(R_2, G_2, B_2)$, through the Euclidean distance between the two color vectors

$$\Delta E^*_{RGB} = \sqrt{(R_2 - R_1)^2 + (G_2 - G_1)^2 + (B_2 - B_1)^2} \tag{B.1}$$

## B.2 CIE Color-differences

CIE color spaces are typically good in this work, because perceptually similar colors are in a narrow interval (typically around $5.0$ units in the CIE $L^*a^*b^*$ space)(FAIRCHILD, 2005). This fact enables the easy interpretation of the color-difference threshold, $t$, required for the CWLD. There are several ways to compute the CIE $L^*a^*b^*$ color-difference. Below we present those that were used in this paper.

### B.2.1 CIE $\Delta E^*_{ab}$ 1976

The simplest color-difference equation is associated with the Euclidean distance of two CIE $L^*a^*b^*$ colors, $(L^*_1, a^*_1, b^*_1)$ and $(L^*_2, a^*_2, b^*_2)$, as shown in Equation B.2 (BERNS, 2000)

$$\Delta E^*_{ab} = \sqrt{(L^*_2 - L^*_1)^2 + (a^*_2 - a^*_1)^2 + (b^*_2 - b^*_1)^2} \tag{B.2}$$

The color distance can also be calculated from the CIE cylindrical coordinate system (CIE $L^*C^*_{ab}h$). To obtain $\Delta E^*_{ab}$ having two colors in the cylindrical coordinate system, $(L^*_1, C^*_1, h_1)$ and $(L^*_2, C^*_2, h_2)$, the Equations B.4, B.5, and B.6 are used (BERNS, 2000).

$$\Delta L^* = L^*_2 - L^*_1 \tag{B.3}$$

$$\Delta C^*_{ab} = \sqrt{a^{*2}_2 + b^{*2}_2} - \sqrt{a^{*2}_1 + b^{*2}_1} \tag{B.4}$$

$$\Delta H^*_{ab} = \frac{a^*_1 b^*_2 - a^*_2 b^*_1}{0.5\sqrt{C^*_1 C^*_2 + a^*_1 a^*_2 + b^*_2 b^*_1}} \tag{B.5}$$

$$\Delta E^*_{ab} = \sqrt{\Delta L^{*2} + (\Delta C^*_{ab})^2 + (\Delta H^*_{ab})^2} \tag{B.6}$$

The regions of **Just Noticeable Difference** (**JND**) are approximated through ellipses. Such ellipses vary according test conditions.

### B.2.2 Color Measurement Committee CMC(*l*:*c*) *Color-Difference Equation*

Color-difference equations proposed before the CMC($l$:$c$) exhibited wrong behaviors for very dark and near neutral colors. Also, the lightness weighting function of previous formulas was not good enough for textiles. Therefore, the CMC decided to improve the color-difference equation by adding weights to each component of the formula together with two constants, $l$ and $c$, to weight lightness and chroma relative to hue (BERNS, 2000).

The formula is presented in Equation B.7:

$$CMC(l:c) = \sqrt{\left(\frac{\Delta L^*}{lS_L}\right) + \left(\frac{\Delta C^*_{ab}}{cS_C}\right) + \left(\frac{\Delta H^*_{ab}}{S_H}\right)} \tag{B.7}$$

where:

$$S_L = \begin{cases} \frac{0.004097L^*}{1+0.01765L^*} & \text{, if } L^* \geq 16 \\ 0.511 & \text{, otherwise} \end{cases} \tag{B.8}$$

$$S_C = \frac{0.0638C^*_{ab}}{1 + 0.0131C^*_{ab}} + 0.638 \tag{B.9}$$

$$S_H = S_C(TF + 1 - F) \tag{B.10}$$

where:

$$F = \sqrt{\frac{C^{*4}_{ab}}{C^{*4}_{ab} + 1900}} \tag{B.11}$$

$$T = \begin{cases} 0.511 & \text{, if } 164^\circ \leq h_{ab} \leq 345^\circ \\ 0.38 + |0.4\cos(h_{ab} + 35)| & \text{, otherwise} \end{cases} \tag{B.12}$$

### B.2.3 CIE $\Delta E^*_{94}$ 1994

In order to improve the color-difference uniformity measurements the CIE $\Delta E^*_{76}$ was modified based on the evaluation of empirical data obtained by experiments with the CMC($l$:$c$) equation. The CMC equation was simplified and the results were improved for some datasets(FAIRCHILD, 2005). Again there are parametric factors to adjust the relative weighting of lightness, chroma and hue. Given two colors in the cylindrical coordinate system CIE $L^*C^*h$, ($L^*_1, C^*_1, h_1$) and ($L^*_2, C^*_2, h_2$), CIE 1994 is obtained through

$$\Delta E^*_{94} = \sqrt{\left(\frac{\Delta L^*}{K_L S_L}\right) + \left(\frac{\Delta C^*_{ab}}{K_C S_C}\right) + \left(\frac{\Delta H^*}{K_H S_H}\right)} \tag{B.13}$$

where:

$$K_L = K_C = K_H = 1 \tag{B.14}$$

for reference conditions.

The lightness, chroma and hue weighing functions $S_L$, $S_C$ and$S_H$, respectively, can be computed by:

$$S_L = 1 \tag{B.15}$$

$$S_C = 1 + 0.045C^*_{ab} \tag{B.16}$$

$$S_H = 1 + 0.015C^*_{ab} \tag{B.17}$$

where the value of $C^*_{ab}$ is the CIE chroma value of the reference color. However, if the reference cannot be deemed:

$$C^*_{ab} = \sqrt{C^*_1 C^*_2} \tag{B.18}$$

where $C_1^*$ and $C_2^*$ are the chroma values for the two colors being compared. In this work we consider the color in the center of the kernel as the reference color.

Unfortunately, some problems are known to this equation, such as the wrong perceptual difference around blue and gray centers (LUO; CUI; RIGG, 2001).

### B.2.4 CIE $\Delta E_{00}^*$ 2000

CIE 2000 addresses some corrections with bluish colors through a rotation term, $R_T$, as well as other compensations for neutral colors, lightness, $S_L$, chroma,$S_C$, and hue, $S_H$. Given two colors in the cylindrical coordinate system CIE $L^*C^*h$, $(L_1^*, C_1^*, h_1)$ and $(L_2^*, C_2^*, h_2)$, and their respective Cartesian coordinates in the CIE $L^*a^*b^*$ space, $(L_1^*, a_1^*, b_1^*)$ and $(L_2^*, a_2^*, b_2^*)$, we can obtain the CIE 2000 color-difference between then through

$$\Delta E_{00}^* = \sqrt{\left(\frac{\Delta L'}{S_L}\right)^2 + \left(\frac{\Delta C'}{S_C}\right)^2 + \left(\frac{\Delta H'}{S_H}\right)^2 + R_T \frac{\Delta C' \Delta H'}{S_C S_H}} \tag{B.19}$$

where the lightness difference, $\Delta L'$, is

$$\Delta L' = L_2^* - L_1^* \tag{B.20}$$

The compensation for lightness is given by

$$S_L = 1 + \frac{0.015(\bar{L} - 50)^2}{\sqrt{20 + (\bar{L} - 50)^2}} \tag{B.21}$$

where

$$\bar{L} = \frac{L_1^* + L_2^*}{2} \tag{B.22}$$

The chroma difference, $\Delta C'$, is

$$\Delta C' = C_2' - C_1' \tag{B.23}$$

where

$$C_1' = \sqrt{a_1'^2 + b_1^{*2}} \tag{B.24}$$

$$C_2' = \sqrt{a_2'^2 + b_2^{*2}} \tag{B.25}$$

and

$$a_1' = a_1^* + \frac{a_1^*}{2}(1 - \upsilon) \tag{B.26}$$

$$a_2' = a_2^* + \frac{a_2^*}{2}(1 - \upsilon) \tag{B.27}$$

where

$$\upsilon = \sqrt{\frac{\bar{C}^7}{\bar{C}^7 + 25^7}} \tag{B.28}$$

and

$$\bar{C} = \frac{C_1^* + C_2^*}{2} \tag{B.29}$$

The compensation for chroma, $S_C$, is given by

$$S_C = 1 + 0.045\bar{C}' \tag{B.30}$$

where

$$\bar{C}' = C_1' + C_2' \tag{B.31}$$

The hue difference, $\Delta H'$, is

$$\Delta H' = 2\sqrt{C_1' C_2'} \sin\left(\frac{\Delta h'}{2}\right) \tag{B.32}$$

where

$$\Delta h' = \begin{cases} h_2' - h_1' & \text{, if } ||h_1' - h_2'|| \leq \pi \\ h_2' - h_1' + 2\pi & \text{, if } ||h_1' - h_2'|| > \pi \quad \text{, and } h_2' \leq h_1' \\ h_2' - h_1' - 2\pi & \text{, if } ||h_1' - h_2'|| > \pi \quad \text{, and } h_2' > h_1' \end{cases} \tag{B.33}$$

and

$$h_1' = \arctan\left(\frac{b_1^*}{a_1'}\right) \mod 2\pi \tag{B.34}$$

$$h_2' = \arctan\left(\frac{b_2^*}{a_2'}\right) \mod 2\pi \tag{B.35}$$

The compensation for hue, $S_H$, can be obtained through

$$S_H = 1 + 0.015\bar{C}'T \tag{B.36}$$

where

$$\begin{aligned} T = 1 - 0.17\cos\left(\bar{H}' - \tfrac{\pi}{6}\right) + 0.24\cos\left(2\bar{H}'\right) + 0.32\cos\left(3\bar{H}'\tfrac{\pi}{30}\right) \\ -0.20\cos\left(4\bar{H}' - 21\tfrac{\pi}{60}\right) \end{aligned} \tag{B.37}$$

and

$$\bar{H}' = \begin{cases} \frac{h_2' + h_1' + 2\pi}{2} & \text{, if } ||h_1' - h_2'|| > \pi \\ \frac{h_2' + h_1'}{2} & \text{, if } ||h_1' - h_2'|| \leq \pi \end{cases} \tag{B.38}$$

The rotation term, $R_T$, used to correct problems with bluish colors is obtained through

$$R_T = -2\upsilon \sin\left[\frac{\pi}{6}\exp\left(-\left(\frac{\bar{H}' - 275^\circ}{25}\right)^2\right)\right] \tag{B.39}$$

This color-difference demands double precision calculations and a careful analysis when applying the square-root of some Equations, since we may fall outside the $\Re$ numbers.

# REFERENCES

ACHAR, S. et al. Self-supervised segmentation of river scenes. In: ROBOTICS AND AUTOMATION (ICRA), 2011 IEEE INTERNATIONAL CONFERENCE ON. **Anais...** Piscataway, NJ, USA: IEEE Press, 2011. p.1050–4729.

BAZIN, J.-C. et al. UAV Attitude estimation by vanishing points in catadioptric images. In: ROBOTICS AND AUTOMATION, 2008. ICRA 2008. IEEE INTERNATIONAL CONFERENCE ON. **Anais...** Piscataway, NJ, USA: IEEE Press, 2008. p.2743 –2749.

BERNS, R. S. **Billmeyer and Saltzman's Principles of Color Technology**. 3.ed. New York: John Wiley & Sons, Inc., 2000. 247p.

CANNY, J. A Computational Approach to Edge Detection. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, IEEE Press, v.PAMI-8, n.6, p.679 –698, nov. 1986.

CASELLES, V.; MONASSE, P. Introduction. In: **Geometric Description of Images as Topographic Maps**. Springer Berlin Heidelberg, 2010. p.1–7. (Lecture Notes in Mathematics, v.1984).

CHIANG, Y.-Y. et al. Automatic and Accurate Extraction of Road Intersections from Raster Maps. **GeoInformatica**, v.13, n.2, p.121–157, June 2009.

COMANICIU, D.; RAMESH, V. Mean shift and optimal prediction for efficient object tracking. **Image Processing, 2000. Proceedings. 2000 International Conference on**, IEEE Press, v.3, p.70–73 vol.3, 2000.

COMANICIU, D.; RAMESH, V.; MEER, P. Real-time tracking of non-rigid objects using mean shift. **Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on**, IEEE Press, v.2, p.142–149 vol.2, 2000.

CRICK, F. **The Astonishing Hypothesis**. 1.ed. New York, NY: Scribner, 1995. 336p.

DUDA, R. O.; HART, P. E. Use of the Hough transformation to detect lines and curves in pictures. **Commun. ACM**, New York, NY, USA, v.15, n.1, p.11–15, 1972.

FAIRCHILD, M. D. **Color Appearance Models**. 2.ed. Wiley, 2005. 408p. (The Wiley Series in Imaging Science and Technology).

FERNANDES, L. A.; OLIVEIRA, M. M. Real-time line detection through an improved Hough transform voting scheme. **Pattern Recognition**, v.41, n.1, p.299 – 314, 2008.

FISCHLER, M. A.; TENENBAUM, J. M.; WOLF, H. C. Detection of roads and linear structures in low-resolution aerial imagery using a multisource knowledge integration technique. **Computer Graphics and Image Processing**, v.15, n.3, p.201 – 223, 1981.

GEMAN, D.; JEDYNAK, B. An active testing model for tracking roads in satellite images. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, IEEE Press, v.18, n.1, p.1 –14, jan. 1996.

GHIDONI, S.; FINOTTO, M.; EMANUELEMENEGATTI. Self-learning visual inspection system for cable crimping machines. In: ROBOTICS AND AUTOMATION (ICRA), 2011 IEEE INTERNATIONAL CONFERENCE ON. **Anais...** Piscataway, NJ, USA: IEEE Press, 2011. p.1 – 4.

GIOI, R. von et al. LSD: a fast line segment detector with a false detection control. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, IEEE Press, v.32, n.4, p.722 –732, apr. 2010.

GHOSH, A. K. (Ed.). **Digital Image Processing**. 2nd.ed. Prentice Hall of India, 2007. 322-323p.

GUAN, S.; LUO, M. R. Investigation of parametric effects using small colour differences. **Color Res. Appl.**, v.24, n.5, p.331–343, 1999.

GUAN, S.; RONNIER, M.; LUO. Investigation of parametric effects using large colour differences. **Color Res. Appl.**, v.24, n.5, p.356–368, 1999.

GUIZILINI, V.; RAMOS, F. Visual odometry learning for unmanned aerial vehicles. In: ROBOTICS AND AUTOMATION (ICRA), 2011 IEEE INTERNATIONAL CONFERENCE ON. **Anais...** Piscataway, NJ, USA: IEEE Press, 2011. p.6213 –6220.

GUO, Z.; HALL, R. W. Parallel thinning with two-subiteration algorithms. **Commun. ACM**, New York, NY, USA, v.32, p.359–373, March 1989.

HWANGBO, M.; KANADE, T. Visual-inertial UAV attitude estimation using urban scene regularities. In: ROBOTICS AND AUTOMATION (ICRA), 2011 IEEE INTERNATIONAL CONFERENCE ON. **Anais...** Piscataway, NJ, USA: IEEE Press, 2011. p.2451 –2458.

JIAN-GUO, X. et al. A Novel Multi-Touch Human-Computer-Interface Based on Binocular Stereo Vision. In: INTELLIGENT UBIQUITOUS COMPUTING AND EDUCATION, 2009 INTERNATIONAL SYMPOSIUM ON. **Anais...** 2009. p.319–323. Binocular hand tracking.

KOLLER, T. et al. Multiscale detection of curvilinear structures in 2-D and 3-D image data. In: COMPUTER VISION, 1995. PROCEEDINGS., FIFTH INTERNATIONAL CONFERENCE ON. **Anais...** 1995. p.864–869.

KWEON, I. S.; KANADE, T. Extracting Topographic Terrain Features from Elevation Maps. **CVGIP: Image Understanding**, v.59, n.2, p.171 – 182, 1994.

LI, S.-X.; CHANG, H.-X.; ZHU, C.-F. Fast curvilinear structure extraction and delineation using density estimation. **Comput. Vis. Image Underst.**, New York, NY, USA, v.113, n.6, p.763–775, 2009.

LI, Y.; LIU, G.; QIAN, X. Ball and Field Line Detection for Placed Kick Refinement. In: INTELLIGENT SYSTEMS, 2009. GCIS '09. WRI GLOBAL CONGRESS ON. **Anais...** 2009. v.4, p.404–407.

LINDEBERG, T. Edge detection and ridge detection with automatic scale selection. In: COMPUTER VISION AND PATTERN RECOGNITION, 1996. PROCEEDINGS CVPR '96, 1996 IEEE COMPUTER SOCIETY CONFERENCE ON. **Anais...** IEEE Press, 1996. p.465 –470.

ZHANG, D. (Ed.). **Medical Biometrics**. Springer Berlin / Heidelberg, 2008. p.49–56. (Lecture Notes in Computer Science, v.4901/2008).

LIU, L.; ZHANG, D. Palm-line detection. In: IMAGE PROCESSING, 2005. ICIP 2005. IEEE INTERNATIONAL CONFERENCE ON. **Anais...** IEEE Press, 2005. v.3, p.III269–III272.

LIU, L.; ZHANG, D.; YOU, J. Detecting Wide Lines Using Isotropic Nonlinear Filtering. **Image Processing, IEEE Transactions on**, IEEE Press, v.16, n.6, p.1584–1595, June 2007.

LIU, W. et al. Vision-Based Real-Time Lane Marking Detection and Tracking. In: INTELLIGENT TRANSPORTATION SYSTEMS, 2008. ITSC 2008. 11TH INTERNATIONAL IEEE CONFERENCE ON. **Anais...** 2008. p.49–54.

LUO, M. R. Development of colour-difference formulae. **Review of Progress in Coloration and Related Topics**, v.32, n.1, p.28–39, 2002.

LUO, M. R.; CUI, G.; RIGG, B. The development of the CIE 2000 colour-difference formula: ciede2000. **Color Res. Appl.**, v.26, n.5, p.340–350, 2001.

MIKSIK, O. et al. Robust detection of shady and highlighted roads for monocular camera based navigation of UGV. In: ROBOTICS AND AUTOMATION (ICRA), 2011 IEEE INTERNATIONAL CONFERENCE ON. **Anais...** Piscataway, NJ, USA: IEEE Press, 2011. p.1050–4729.

MONDRAGON, I. F. et al. 3D pose estimation based on planar object tracking for UAVs control. In: ROBOTICS AND AUTOMATION (ICRA), 2010 IEEE INTERNATIONAL CONFERENCE ON. **Anais...** Piscataway, NJ, USA: IEEE Press, 2010. p.35 –41.

NAYATANI, Y. Simple estimation methods for the Helmholtz-Kohlrausch effect. **Color Research & Application**, v.22, n.6, p.385–401, 1997.

NAYATANI, Y.; SAKAI, H. Confusion between observation and experiment in the Helmholtz-Kohlrausch effect. **Color Research & Application**, v.33, n.3, p.250–253, 2008.

PAKTER, H. et al. Computer-Assisted Methods to Evaluate Retinal Vascular Caliber: what are they measuring? **Investigative Ophthalmology & Visual Science (IOVS)**, v.52, p.810–815, Nov 2010.

QIN-LI et al. Dark line detection with line width extraction. In: IMAGE PROCESSING, 2008. ICIP 2008. 15TH IEEE INTERNATIONAL CONFERENCE ON. **Anais...** IEEE Press, 2008. p.621–624.

REID, A.; RAMOS, F.; SUKKARIEH, S. Multi-class classification of vegetation in natural environments using an Unmanned Aerial system. In: ROBOTICS AND AUTOMATION (ICRA), 2011 IEEE INTERNATIONAL CONFERENCE ON. **Anais...** Piscataway, NJ, USA: IEEE Press, 2011. p.2953 – 2959.

SMIRNOV, N. Table for Estimating the Goodness of Fit of Empirical Distributions. **The Annals of Mathematical Statistics**, v.2, n.2, p.279–281, Jun 1948.

SMITH, S. M.; BRADY, J. M. SUSAN: a new approach to low level image processing. **International Journal of Computer Vision**, v.23, n.1, p.45–78, May 1997.

SMITH, S. M.; BRADY, J. M. SUSAN: a new approach to low level image processing. **International Journal of Computer Vision**, v.23, n.1, p.45–78, May 1997.

STEGER, C. An unbiased detector of curvilinear structures. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, IEEE Press, v.20, n.2, p.113–125, Feb 1998.

SUN, L.; LIU, G. Field lines and players detection and recognition in soccer video. In: ACOUSTICS, SPEECH AND SIGNAL PROCESSING, 2009. ICASSP 2009. IEEE INTERNATIONAL CONFERENCE ON. **Anais...** IEEE Press, 2009. p.1237–1240.

VERMA, O. Fuzzy edge detection based on similarity measure in colour image. In: INDIA CONFERENCE (INDICON), 2010 ANNUAL IEEE. **Anais...**, 2010. p.1 –6.

WANG, Z.; WU, F.; HU, Z. MSLD: a robust descriptor for line matching. **Pattern Recognition**, v.42, n.5, p.941–953, May 2009.

XIN, J. H.; LAM, C. C.; LUO, M. R. Investigation of parametric effects using medium colour-difference pairs. **Color Res. Appl.**, v.26, n.5, p.376–383, 2001.

XU, H.; YAGUCHI, H.; SHIOIRI, S. Testing CIELAB-based color-difference formulae using large color differences. **Optical Review**, v.8, p.487–494, 2001. 10.1007/BF02931740.

ZHANG, M.; LIU, J.-C. Directional Local Contrast Based Blood Vessel Detection in Retinal Images. In: IMAGE PROCESSING, 2007. ICIP 2007. IEEE INTERNATIONAL CONFERENCE ON. **Anais...** IEEE Press, 2007. v.4, p.IV –317–IV –320.