A vectorial Color Edge detector using spatiocolorimetric neighborhood hypergraph and perceptual color distance

S. Rital*, H. Cherifi* and D. Aboutajdine**(IEEE Senior member)

* LIRSA, University of Burgundy, Faculty of Sciences Mirande
Dijon, France
email: soufiane.rital, hocine.cherifi@u-bourgogne.fr

** GSCM, department of physics, B. P. 1014 Science Faculty of Rabat
Rabat, Morocco.
email: aboutaj@fsr.ac.ma.

ABSTRACT
In this paper, we present an edge detection approach in color image using neighborhood hypergraph. The edge structure is detected by a structural model. The Color Image Neighborhood Hypergraph (CINH) representation is first computed using a perceptual color distance, then the hyperedges of CINH are classified into noise, edge or others based on hypergraph properties. To evaluate the algorithm performance, experiments were carried out on synthetic and real color images corrupted by $\alpha$-stable noise. The results show that the proposed edge detector finds the edges properly from color images.

KEY WORDS
Graph, Hypergraph, Color space, Neighborhood hypergraph, noise detection, color distance

1 Introduction
Edge detection is a front-end processing step in most computer vision and image understanding systems. The accuracy and reliability of edge detection is critical to the overall performance of these systems. Earlier developments of edge detection are mostly based on direct application of the difference operation and could encounter difficulties when images are corrupted by noise. Much research has been carried out in the effort to detect edge structures in the presence of noise. One type of edge detector employs smoothing before using the difference operation, so as to offset the effects of noise.

The use of color in edge detection increases the amount of information needed for processing which complicates the definition of the problem. A number of approaches have been proposed from processing individual planes to true vector-based approaches. Multidimensionality adds one important step, image recombination, which can be inserted at different places. This insertion translates into performing some sets of operations on each color component. The intermediate results are then combined into a single output. The point at which recombination occurs is key to understanding the different categories of color edge detection algorithms: output fusion methods, multidimensional gradient methods, and vector methods.

In output fusion methods, gray-scale edge detection is carried out independently in each color component; combining these results yields the final edge map. Multidimensional gradient methods are characterized by a single estimate of the orientation and strength of an edge at a point. The first such method belongs to Robinson [6], who also appears to have published the first paper on color edge detection. He computed 24 directional derivatives (8 neighbors $\times$ 3 components) and choose the one with the largest magnitude as the gradient. However, it was Di Zenzo [11] who wrote the classic paper on multidimensional gradients. His method was derived algebraically, but it is perhaps better explained in terms of matrices. A $2 \times 2$ matrix is formed from the outer product of the gradient vector in each component. These matrices are summed over all components, and the square root of the principal eigenvalue (i.e., the principal singular value) becomes the magnitude of the gradient. The corresponding eigenvector yields the gradient direction. This approach was used in various forms by Cumani [4]. In vector methods, the decomposition and recombination steps nullify each other; the vector nature of color is preserved throughout the computation. How to represent and use these vectors has varied greatly. Perhaps the most compelling work in vector methods so far has been that of Trahanias and Venetsanopoulos [10]. Their method used the median of a set of vectors, which is the vector in that set whose distance to all other vectors is minimized. Once the vector median has been determined, vectors in a neighborhood are sorted by their distances from the vector median, and various statistics are measured and used for edge detection. The algorithms that incorporate more vector op-
erations are preferable to those with fewer [7].

In this paper, we propose a structural approach to edge detection which is robust even when the color image is corrupted by noise. This approach can be classified into vector method category. The method is based on the structural model in color image neighborhood hypergraph representation. In hypergraph [2] the vertices \( V \) correspond to objects and hyperedges \( E \) represent the interrelations between these objects. Various applications have been proposed for gray scale image using Image Neighborhood Hypergraph representation (INH) [3][5].

The proposed color edge detector executes in two phases. The color image neighborhood hypergraph is first computed. Then, the edge and neighborhood structures are detected by the structural model. In section 2, We first briefly review some background on hypergraph theory. Then, we define the neighborhood hypergraph associated with any color image, denoted by Color Image Neighborhood Hypergraph (CINH). In section 4, we introduce a perceptual color distance using in CINH generation. The proposed edge detector algorithm is presented in Section 5 and its performance is illustrated in Section 6. Finally, the paper ends with a conclusion in Section 7.

2 Background on hypergraph theory

As our main interest in this paper is to use combinatorial models, we will introduce basic tools that are needed. A hypergraph \( H \) on a set \( X \) is a family \( (E_i)_{i \in I} \) of non-empty subsets of \( X \) called hyperedges with:

\[
\bigcup_{i \in I} E_i = X, \quad I = \{1, 2, \ldots, n\}, \quad n \in \mathbb{N}.
\]

Given a graph \( G \), the hypergraph having the vertices of \( G \) as vertices and the neighborhood of these vertices as hyperedges (including this vertices) is called the neighborhood hypergraph of \( G \). To each graph we can associate a neighborhood hypergraph : \( H_G = (X, (E_x = \{ x \} \cup \Gamma(x))) \), where \( \Gamma(x) \) is the neighborhood of each vertex \( x \).

Let us note: \( H_G = (X; (E_i)_{i \in I}) \). A chain is a succession of the hyperedges \( E_x \). It is disjointed if the hyperedges \( E_x \) are not connected two by two. An hyperedge \( E_i \) is isolated if and only if \( \forall j \in I, j \neq i \) \( E_i \cap E_j \neq \emptyset \) then \( E_j \subseteq E_i \).

3 Color Image Neighborhood Hypergraph

In this paper the color image will be represented by the following mapping : \( I : \mathcal{X} \subseteq \mathbb{Z}^2 \rightarrow \mathcal{C} \subseteq \mathbb{Z}^n \), \( n = 3 \) for color image). Vertices of \( \mathcal{X} \) are called pixels, elements of \( \mathcal{C} \) are called color. A distance \( d \) on \( \mathcal{X} \) defines a grid (a graph connected, regular, without both loop and multi-edge). Let \( d' \) be a distance on \( \mathcal{C} \), we have a neighborhood relation on an image defined by :

\[
\forall x \in \mathcal{X}, \quad \Gamma_{\lambda,\beta}(x) = \{ x' \in \mathcal{X}, x' \neq x \}
\]

\[
d'(I(x), I(x')) < \lambda \quad \text{and} \quad d(x, x') \leq \beta \quad (1)
\]

The neighborhood of \( x \) on the grid will be denoted by \( \Gamma_{\lambda,\beta}(x) \). So to each color image we can associate a hypergraph called Color Image Neighborhood Hypergraph CINH : \( H_{\Gamma_{\lambda,\beta}} = (X, \{(x) \cup \Gamma_{\lambda,\beta}(x)\}_{x \in X}) \). On a grid \( \Gamma_{\beta} \), to each pixel \( x \) we can associate a neighborhood \( \Gamma_{\lambda,\beta}(x) \), according to predicate \( \lambda \). The predicate \( \lambda \) may be completely arbitrary provided it is useful for a task domain. It may be defined on a set of points, it may use colors, or some symbolic representation of a set of colors, or it may be a combination of several predicates, and so on. The thresholding can be carried out in two manners. In the first case, the threshold is given for all the pixels of the image, whereas in the second case, the threshold is generated locally then applied in an adaptive way to the unit of the pixels.

4 A Perceptual Color Distance

Before computing the color image neighborhood hypergraph representation, we must first compute the distance between each pair of colors in the neighborhood \( \Gamma_{\lambda,\beta}(x) \). For this distance measure to agree with human perception, we must find a good combination of a color space and a function on this space. There is general agreement that the organization of color in our perceptual systems is three-dimensional, but the actual assignment of coordinates to colors depends on the task involved. As a result, many color spaces exist (see [8] for details on many of them). Few of these spaces were designed to mimic perceived color distances, however. In particular, the Red, Green, and Blue (RGB) color space has hardly ever been advocated as a good space for measuring color distances. One of the few color spaces that was designed from a perceptual standpoint is the CIE-L*a*b* color space [8]. CIELab was constructed from the results of psychophysical color similarity experiments. The Euclidean distance between two nearby colors in this space is intended to be equivalent to their perceptual distance. In [12], X.M. Zhang et al. described a spatial extension to the CIELAB color metric named S-CIELab. The S-CIELAB extension includes a spatial processing step, prior to the CIELAB \( \Delta E \) calculation, so that the results correspond better to color difference perception by the human eye. In [9], G. Sharma presented a new color-difference formula CIEDE2000. The euclidean Distance (ED) is the metric usually used in N-dimensional vector space.

CIELab is not ideal; in particular, the experiments that led to its creation used large uniform patches rather than pixelized elements, which has a noticeable effect on our perception. However, we have found it to be effective in providing an accurate measure of perceptual color distance. Unfortunately, simply measuring Euclidean distance in CIELab is insufficient. An important caveat is that the equivalence between Euclidean and perceptual distances
holds for small distances only. For larger distances, the most we can say about a pair of colors is that they are different. For feature detection, this is exactly what is required; once two colors are far enough apart that we can perceive contrast between them, their actual Euclidean distance is irrelevant. This key observation turns out to be independent of our choice of color space. We are not interested in any physical properties of the color stimuli; only the amount of perceptual dissimilarity between two stimuli is important. Stimuli that are infinitely far apart should have a distance of one, not infinity. Including the requirements that the function be smooth, monotonic, and a metric leads to a choice of

\[
d_{ij} = 1 - \exp(-E_{ij}/\xi)
\]  

(2)

In other words, the ground distance between color \(i\) and color \(j\) is an exponential measure, with steepness governed by \(\xi\) (we use \(\xi = 14.0\), of the Euclidean distance \(E_{ij}\) between them in CIELab. This function also has the advantage of being roughly linear for small distances, which is why the choice of a color space is still relevant. Theoretical justification for this measure can be found in the work of A. Mark Ruzon et al. [7].

5 Edge detector algorithm

The most usual edge detector give the good edges maps results. However, they are often less robust in the presence of noise. Consequently, the most algorithms are based in a previously smoothing action in order to improve the edge detection and to reduce the effect of noise. This pre-filtering process seems not to be adequate when the noise suppression does not preserve useful information. In this section, we describe a structural edge detection algorithm based on color image neighborhood hypergraph representation and structural model of edge and noise. These structural models have two goals : firstly, the structural noise model may be used for both : noise detection in order to apply an edge detection algorithm without noise interferences and edge preservation. Secondly, the structural edge model may be used to edge detection. In the figure 1, we illustrate the block diagram of the proposed algorithm. It starts firstly with a CINH generation and secondly by a noise and edge classifications. This classification is based on two structural models.

![Figure 1: The block diagram of the proposed color edge detector algorithm.](image)

To model a noisy hyperedge \(E_{\lambda,\beta}^{\text{noise}}(x)\) and an edge \(E_{\lambda,\beta}^{\text{edge}}(x)\) in color image, we propose the following definitions:

**Noise model** We say that \(E_{\lambda,\beta}(x)\) is a noise hyperedge if it verifies one of the two conditions :

1. The cardinality of \(E_{\lambda,\beta}(x)\) is equal to 1: \(E_{\lambda,\beta}(x)\) is not contained in disjoined thin chain having \(\omega\) elements at least.
2. \(E_{\lambda,\beta}(x)\) is an isolated hyperedge and there exists an element \(y\) belonging to the open neighborhood of \(E_{\lambda,\beta}(x)\) on the grid, such that \(E_{\lambda,\beta}(y)\) is isolated. (i.e. \(E_{\lambda,\beta}\) is isolated and it has an isolated hyperedge in its neighborhood on the grid).

In figure 2, we present a graphical illustration of noisy hyperedge definition.

![Figure 2: Graphical illustration of the noise model.](image)

**Edge model** For every \(E_{\lambda,\beta}(x)\).

a. If \(E_{\lambda,\beta}(x)\) is not isolated, contained in a chain having \(\omega\) elements at least and \(E_{\lambda,\beta}(x) \neq E_{\lambda,\beta}^{\text{noise}}(x)\) defined in the first definition (Def. 5) then \(E_{\lambda,\beta}(x) = E_{\lambda,\beta}(x)\);

b. If \(E_{\lambda,\beta}(x)\) is isolated, contained in a disjoined chain having \(\omega\) elements at least and \(E_{\lambda,\beta}(x) \neq E_{\lambda,\beta}^{\text{noise}}(x)\) defined in the first definition (Def. 5) then \(E_{\lambda,\beta}(x) = E_{\lambda,\beta}(x)\);

b. If \(E_{\lambda,\beta}(x)\) is an isolated hyperedge and there exists an element \(y\) belonging to the open neighborhood of \(E_{\lambda,\beta}(x)\) on the grid, such that \(E_{\lambda,\beta}(y)\) is isolated. (i.e. \(E_{\lambda,\beta}\) is isolated and it has an isolated hyperedge in its neighborhood on the grid).

In figure 4, we present a graphical illustration of edge definition. In color image and more generally, a digital image, we find several types of edges : step, concave slope, convex slope, roof, valley and staircase edges. Among these types of edge one often finds step and roof edges. These two types of edges are detected respectively with conditions (a) and (b) of edge model. (Fig. 3).

6 Results and Discussion

We shall present a set of experiments in order to assess the performance of the color edge detection algorithm we have
discussed so far. Our goal in the first experiment is to evaluate the noisy hyperedges detection. In the second experiment, we evaluate the color edge structural model in no corrupted color images. Finally, we evaluate the color edge detection in corrupted color images. The distance measure between two vectors in a given color space is defined by equation 2. We used 256 × 256-pixel ”Peppers”, ”Logo” and ”Fruit” images, all of them being true-color images (24 bits/pixel). We tested the performance of the noisy hyperedge detection described above in the presence of α-stable noise.

An important requirement for most image processing problems is the specification for the corrupting noise distribution. The most widely used model is the gaussian random process. However, in some environments, the gaussian noise model may not be appropriate. A number of models have been proposed for non-Gaussian phenomena.

In recent years, further research into signal modeling has led to the realization that many natural phenomena can be better represented by distributions of a more impulsive nature. One type of distribution that exhibits heavier tails than the Gaussian is the class of α-stable distributions.

The α-stable distribution is a useful model of noise distribution. For a symmetrical distribution, the characteristic function is given by: $\psi(t) = e^{i\omega t - \gamma |t|^\alpha}$, where: (1) $\alpha$ is the characteristic exponent satisfying $0 < \alpha \leq 2$. The characteristic exponent controls the heaviness of the tails of the density function. The tails are heavier, and thus the noise more impulsive, for low values of $\alpha$ while for a larger $\alpha$ the distribution has a less impulsive behavior. (2) $\beta$ is the location parameter ($-\infty < \beta < +\infty$). (3) $\gamma$ is the dispersion parameter ($\gamma > 0$), which determines the spread of the density around its location parameter.

In the evaluation and comparison of the noisy hyperedge detection, two criteria are employed namely: probability of the decision ($\hat{P}_d$) and probability of the false alarm ($\hat{P}_f$). Let’s consider an image corrupted with a noise source and the true locations of the noise are stored for comparison. For each detector under consideration the $\hat{P}_d$ and $\hat{P}_f$ are computed by comparing them with the true set of noise. A good detector will have a high detection probability and low false alarm probability.

### 6.1 Noise removal

In this section, we first evaluate the use of distance $d_{ij}$ (Eq. 2) in two color spaces RGB and CIELab using the ROC curves of the noisy hyperedge detection ($\hat{P}_d = f(\hat{P}_f)$) at many threshold $\lambda$ in [0, 255] with $\beta = 1$ and $\omega = 5$. The used Peppers color image is corrupted by 6% of α-stable noise ($\alpha = 0.5, \gamma = 1$). The best detector should have $\hat{P}_d$ rates as large as possible, and false alarm rates as small as possible, i.e. a ROC curve that bows away from the diagonal line as much as possible. From figure 5, we can see that in the CINH$_{Lab}$, the detection give a significant results over CINH$_{RGB}$. The results demonstrate the improvement introduced in terms of performance using a distance $d_{ij}$ to compute CINH representation and superiority of CIELab space consequently.

After color space comparison, we evaluate the performance of the proposed noise model after estimating the noisy hyperedges. The objective of the filtering is to remove the noisy hyperedges while preserving the noise-free patterns. In figure 6, we present the results of the noise detection in Fruit color image corrupted by 6% of alpha stable noise with two parameters: $\alpha = 2$ representing a Gaussian distribution noise and $\alpha = 0.5$ representing a more impulsive distribution noise. These two results are compared with the Vector Median Filter (VMF) [1]. The VMF operates using 3 × 3 square processing windows. From the error image between the filtered image and the original image, we note that the proposed algorithm has better edge preservation properties than the VMF filter.
6.2 Edge detection

In this section, we present the results of edge detection in no corrupted color image in order to evaluate the structural model of edge initially on synthesized color images then on real color images. In figure 7(a,b,c), we illustrate first the edge detection results of the synthetic logo color image. This figure contains two algorithms outputs: the one proposed and the Cumani algorithm. These two algorithms could both detect significant edges presented in the Logo color image. Nevertheless, the Cumani algorithm did not detect the junctions of the majority of the objects in the image. In figure 7(a’,b’,c’), we illustrate the edge detection results of Peppers color image. According to these results, we note that the Cumani algorithm detected more of false edges than the proposed algorithm. Let us note that the results of the proposed algorithm or the Cumani algorithm are obtained one adjusting their parameters in order to obtain less false edges and more significant edges. The detection of less false edges and the detection of the junctions favoured the proposed algorithm.

6.3 Edge detection in noisy color image

Two $\alpha$-stable noise distributions with $\alpha = 0.5, \gamma = 1$ and $\alpha = 2, \gamma = 35$ and 6% are added to the Logo and Peppers color images in order to evaluate robustness of the proposed method to noise effects. The two parameters $(\alpha = 2, \gamma = 35)$ and $(\alpha = 0.5, \gamma = 1)$ represent the $\alpha$-stable noise with: Gaussian and more Impulsive noise behaviors respectively. The Gaussian distribution is added to Logo color image while the impulsive distribution is added to Peppers color image. The edge detection results of this two color images are illustrated in figure 8. According to these results, we note that we found the two previous re-
marks are still confirmed: the false edges and the miss of the junctions by Cumani algorithm. Concerning the robustness of the two algorithms to noise effects. We note that the Cumani algorithm is robust to the Gaussian noise while the proposed algorithm is robust to the impulsive noise.

![Figure 7](image)

Figure 7. The edge detection results of Cumani and proposed algorithms. (a,a’) the original color images. (b,b’) the outputs of Cumani algorithm with the parameters ($\sigma = 3, TH = 2$) and ($\sigma = 3, TH = 10$) respectively. (c,c’) the outputs of the proposed algorithm with the parameters ($\lambda = 12, \omega = 5, \beta = 1$) and ($\lambda = 32, \omega = 5, \beta = 1$) respectively.

![Figure 8](image)

Figure 8. The edge detection results of Cumani and proposed algorithms. (a,a’) the corrupted color images with the parameters ($\alpha = 2, \gamma = 35$) and ($\alpha = 0.5, \gamma = 1$) respectively. (b,b’) the outputs of Cumani algorithm with the parameters ($\sigma = 2, TH = 6$) and ($\sigma = 3, TH = 10$) respectively. (c,c’) the outputs of the proposed algorithm with the parameters ($\lambda = 20, \omega = 5, \beta = 1$) and ($\lambda = 35, \omega = 5, \beta = 1$) respectively.

7 Conclusions

A new algorithm for edge detection based on structural model is proposed for color image corrupted with noise. The algorithm includes two phases. first CINH generation based on a perceptual distance and edge or noise classification. Simulation results have shown that it is consistent and reliable even when image quality is significantly degraded by impulsive noise or Gaussian noise. It is effective in both cancellation of noise while preserving details and features detection.

References