

# Distance Maps from Unthresholded Magnitudes

Luis Anton-Canalis<sup>1</sup>, Mario Hernandez-Tejera<sup>1</sup>, and Elena Sanchez-Nielsen<sup>2</sup>

<sup>1</sup> SIANI - University of Las Palmas de Gran Canaria

<sup>2</sup> University of La Laguna,

{lanton,mhernandez}@siani.es, enielsen@ull.es

**Abstract.** A straightforward algorithm that computes distance maps from unthresholded magnitude values is presented, suitable for still images and video sequences. While results on binary images are similar to classic Euclidean Distance Transforms, the proposed approach does not require a binarization step. Thus, no thresholds are needed and no information is lost in intermediate classification stages. Experiments include the evaluation of segmented images using the watershed algorithm and the measurement of pixel value stability in video sequences.

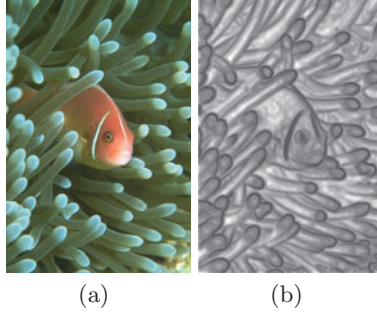
**Keywords:** Distance Transform, Thresholding, pseudodistances.

## 1 Introduction

The Distance Transform (DT), originally proposed by Rosenfeld and Pfaltz [10], has been widely used in computer vision, image processing, pattern recognition and robotics. Applied to an image, a DT assigns a value to each image point that represents the minimum distance to some locus of points, usually belonging to one of the two groups created by a binary partition. Classic DT (i.e. Euclidean Distance Transform) has been applied to object recognition [3] [6], path planning [12], active contour modeling [15] and segmentation [2] [9].

DT usually operates on binary feature maps obtained by thresholding operations like thresholding or edge detection. However, any binarization process based on thresholds systematically creates groups of points in which binary membership may fluctuate through time due to small light changes or image noise. As video processing applications require features to be detected consistently in the spatiotemporal domain, processes based on thresholds should be avoided. In this context, computing distance maps directly from unthresholded magnitude values should increase the stability of further processing stages relying on them.

Different solutions that do not require a binarization step have been proposed to compute pseudo-distances from a given image point to some locus of points. In [7], pseudo-distance maps are computed applying PDEs to an edge-strength function of a grayscale image, obtaining a robust and less noisy skeletonization; in [11], pseudo-distances are weighted by edge magnitude and length; in [1], an intuitive solution named Continuous Distance Transform (CDT) is proposed, where pseudo-distances to brightness and darkness saturation regions are computed directly from grayscale images.



**Fig. 1.** DMUM pseudo-distance values 1(b) directly computed from the unthresholded Sobel gradients of image 1(a)

In this paper, a novel method for computing distance maps directly from unthresholded magnitude values is proposed. No critical information is lost in an intermediate classification step and different magnitude values can be employed (i.e. depth or gradient, see Fig.1). Output values represent the size of the smaller area around each pixel enclosing a relevant amount of accumulated magnitude values that depends solely on the image being studied. A single pass on the magnitude values image is needed, and the integral image [14] is used to speed up area sums. A formal definition and a concise algorithm to compute this Distance Map from Unthresholded Magnitude values (DMUM) are proposed.

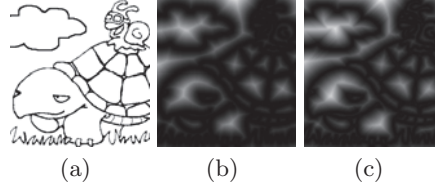
Section 2 describes DMUM, proposing a straightforward algorithm. Section 3 analyzes DMUM values in watershed segmentations and its stability in video sequences and finally Section 4 summarizes the approach and proposes future research lines and applications.

## 2 Distance Maps from Untresholded Gradients

In classic DT each point receives a value that represents a *distance to the closest object point*. However, a suitable definition of *object* is necessary. It is usually achieved through plain thresholding or edge detection, which leads to a classification step where points are considered object or non-object. This is not a trivial problem: an unsuitable classification mechanism may produce an inaccurate binarization, both removing information and adding noise. In DMUM, the *distance to the closest object point* concept is replaced by the *distance to the closest most relevant considered magnitude value region* in a mapping function that relies directly on unthresholded magnitude values. For the sake of simplicity, the rest of this work will work with gradient magnitudes, but many other features can be used instead, like depth maps.

Let us define the set of points bounded by a closed ball of radius  $r$  centered at point  $p$ :

$$B_r[p] \triangleq \{x \in \mathbb{S}^2 | d(x, p) \leq r\} \quad (1)$$



**Fig. 2.** 2(a) Original binary image, 2(b) classic chessboard DT values and 2(c) DMUM values

where  $d(x, p)$  is a distance function. If  $\mathbb{S}^2 \equiv \mathbb{R}^2$  then  $B_r[p]$  is defined in a continuous metric space, while  $\mathbb{S}^2 \equiv \mathbb{Z}^2$  defines  $B_r[p]$  in a discrete metric space. For optimization purposes the Chebyshev distance ( $L_\infty$  norm) is adopted, defining a square around  $p$ .

Given an image  $i \in \mathbb{Z}^2$  and its normalized gradient image  $g^n$ , being its maximum value 1.0, consider the sum of gradients for a given  $B_r[p]$  in  $i$  as:

$$\phi_r(p) = \gamma \cdot \sum_{\forall x \in B_r[p]} g^n(x) \quad (2)$$

being the  $\gamma$  parameter a scaling factor that will be explained later. Radius  $q$  is defined as:

$$q(p) = \arg \sup_r \{\phi_r(p) > 1.0\} \quad (3)$$

that is, the minimum radius  $r$  of a ball  $B_r(p)$  that encloses a sum of gradient  $\phi_r(p)$  that exceeds 1.0. Then, the value of the Distance Map from Unthresholded Magnitudes at each point  $p$  is given by:

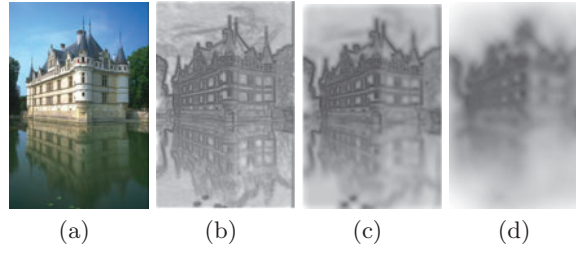
$$\Gamma(p) = q - \sum_{r=1}^q \phi_r(p) \quad (4)$$

DMUM values  $\Gamma(p)$  will be higher for points placed in sparse gradient regions and lower for points with a high gradient or points placed in dense gradient regions. Its application to a binary image produces an output similar to that produced by DT, as seen in Fig.2. Measuring pixel value differences between Fig. 2(b) and Fig. 2(c) returns a dissimilarity smaller than 1.5%.

The accumulation of  $\phi_r(p)$  values in Eq.4 introduces a linear weighting factor. The first and smallest area contributes  $q$  times to this accumulation, because it is also contained on bigger areas. The second one contributes  $q - 1$  times, the third one  $q - 2$  times and so on, while the area that satisfies the condition contributes only once. While raw  $q$  values could be used as the desired output, resulting maps would feature isohypses of heights instead of more continuous values. This linear weighting modifies DMUM values according to local magnitude conditions. Fig. 3 shows this effect. The  $\gamma, \gamma \in (0, 1.0]$  parameter in Eq.2 allows DMUM output values to be softened while computing output values, so no previous filtering is needed. Because region areas grow exponentially ( $1 \times 1$ ,  $2 \times 2$ ,  $3 \times 3 \dots q \times q$ ), the



**Fig. 3.** 3(a) Original image. 3(b) Raw DMUM  $q$  values. 3(c) The accumulation of innermost regions results in smoother DMUM maps.



**Fig. 4.** 4(a) Original image. DMUM computed with  $\gamma = 1.0$  4(b), 0.1 4(c) and 0.01 4(d).

$\gamma$  parameter dampening effect will be much stronger in small areas with low gradient accumulations. Fig.4 shows the effect of different  $\gamma$  values on the same image. Notice how high gradient values are preserved, effectively working as an anisotropic filter.

DMUM values represent the pseudo-distance from a given image point to the closest most relevant gradient accumulation. Due to the limit condition in Eq. 3,  $\phi_r(p) \leq 1.0$ ,  $\phi_r(p) \in [0, q]$  and  $\Gamma(p) \geq 0$ , being zero only for those points which gradient equals 1.0. This satisfies the positive definiteness condition of a metric. However no symmetry and thus no triangle inequality can be defined. Therefore, DMUM map values can only be considered a pseudo-distance, though most works consider both terms interchangeable.

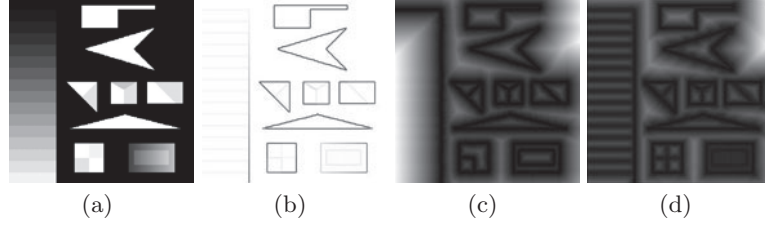
## 2.1 DMUM Algorithm

Computing the DMUM value for each pixel  $p$  in  $i$  consists of finding the smaller area  $B_q(p)$  which gradient sum  $\phi_q(p)$  is equal or higher than 1.0. As only squared regions around  $p$  are considered, due to the adoption of the Chebyshev distance, the sum of gradient values can be costlessly computed using the integral image [14] of  $g$ . The following pseudocode summarizes the whole process:

```

Require: RGB image source sized  $n \times m$ 
Require:  $\gamma \in (0..1]$ 
Ensure: Floating point image DMUM sized  $n \times m$ 
   $DMUM[i, j] \leftarrow 0, i \in [0..n), j \in [0..m)$ 
   $gradient \leftarrow gradient(source)/max(gradient(source))$ 
   $integral \leftarrow integral(gradient)$ 
   $max\_r \leftarrow min(n/2, m/2)$ 
  for  $i \in [0..n)$  do
    for  $j \in [0..m)$  do

```



**Fig. 5.** 5(a) Image with both binary and grayscale regions. 5(b) Sobel gradient. 5(c) DT values. 5(d) DMUM map.

```

 $r \leftarrow 1$ 
 $enclosed\_gradient \leftarrow gradient[i, j] \cdot \gamma$ 
 $gradient\_accum \leftarrow enclosed\_gradient$ 
while  $enclosed\_gradient \leq 1.0$  and  $r \leq max\_r$  do
     $gradient\_accum \leftarrow gradient\_accum + enclosed\_gradient$ 
     $r \leftarrow r + 1$ 
     $enclosed\_gradient \leftarrow (integral(i + r, j + r) + integral(i - r, j - r)) - (integral(i + r, j - r) + integral(i - r, j + r))$ 
     $enclosed\_gradient \leftarrow enclosed\_gradient \cdot \gamma$ 
end while
 $DMUM[i, j] \leftarrow r - gradient\_accum$ 
end for
end for
return  $DMUM$ 

```

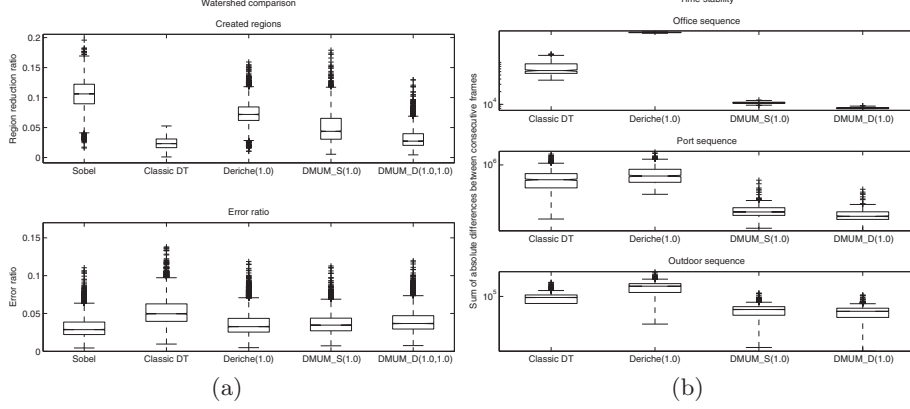
### 3 Results

The main difference between classic DT (computed from a Canny edge map) and the proposed approach is shown in Fig.5. Although both operations behave similarly in binary regions, details on grayscale parts are removed in the DT image. Ideally, the perfect set of Canny parameters would produce an optimal distance map, but finding them is not easy. Each image, even consecutive frames of a video sequence, may need different values. Besides, unevenly lighted images may require parameters to be adjusted locally. However, DMUM is computed from local gradient values (see Eq.3) so the original image structure is better preserved (See Fig. 5).

It was already shown in [1] that distance maps applied directly to grayscale values increase object recognition rates using Chamfer distances. The solution proposed in [?] was specifically designed for OCR applications and computationally expensive. DMUM offers a more general solution that is two orders of magnitude faster and introduces an anisotropic filtering mechanism. Experiments were focused on the analysis of the structure of distance maps and their stability in video sequences.

#### 3.1 Spatial Structure Analysis

Watershed algorithms[13] [2] perform an oversegmentation of a given image, creating groups of pixels that share similar features and reducing the amount of



**Fig. 6.** Watershed region-error ratios and time stability measures

data needed to describe the image. The sorting and region growing mechanism in the basic watershed algorithm reveals relevant morphological relations between pixels.

A total of 5074 images were watershed-segmented, including the Berkeley Segmentation Dataset [8], showing a wide range of indoor and outdoor images and different sizes and noise factors. Sobel gradients, chessboard DT images, Deriche [5] gradients and DMUM values computed both from Sobel ( $DMUM_S$ ) and Deriche ( $DMUM_D$ ) were used as input for the watershed algorithm. Both the number of final regions and the sum of absolute differences between pixels from the segmented image and the original image were measured. These values were normalized with respect to the maximum number of regions (that equals the number of pixels) and the maximum possible error (sum of maximum error for every pixel on each channel) respectively, obtaining the region reduction ratio and the error ratio respectively.

Region reduction and error ratios for each method are depicted in Fig. 6. Classic DT creates the lowest number of regions, but also the highest error, which reveals the loss of information suffered in the binarization step required to compute the classic DT. As DMUM computes pixel values from local regions, differences between neighbouring pixels tend to be smoother, while noisy pixels have a smaller influence on the final outcome.

While  $DMUM_S$  creates less regions than *Deriche*, the application of a Tukey-Kramer multiple comparison test on error ratio values reveals that Deriche and  $DMUM_S$  error ratios are significantly similar, with a confidence level of 99%.  $DMUM_D$  creates even less regions with a slightly higher error, but introduces the computational complexity of computing Deriche gradients.

### 3.2 Temporal Stability Results

It is important for most operators working on video sequences to be stable in time, meaning that the same values should be returned if operational conditions

have not changed significantly. Even small light changes like those coming from the almost imperceptible flicker of a fluorescent bulb may affect the scene, and they certainly affect some image operators like the Canny edge detector.

Three different video sequences were used for measuring stability. The first one is an indoor sequence with no visible changes. As nothing moves, the difference between consecutive frames would be ideally zero, although visually unnoticeable light changes introduce pixel value variations. The second sequence shows a video from a static security camera placed above a harbor, showing images from dawn until dusk. Light conditions change smoothly and some objects cross the scene. Finally, the third sequence is an outdoor video that includes dynamic objects and backgrounds, different settings and changes in lighting conditions.

Classic DT, Deriche gradients,  $DMUM_S$  and  $DMUM_D$  were computed on each frame. Temporal coherence was measured computing the sum of absolute differences between pixel values of consecutive frames for each method, avoiding prior filtering. Fig. 6 depicts measured differences. DMUM values are clearly more stable in video sequences than Deriche and classic DT on the three sequences, showing also a smaller variance.

A new Tukey-Kramer multiple comparison test applied to the time stability test results reveals that  $DMUM_S$  and  $DMUM_D$  values are significantly similar to each other, and different from DT and Deriche. Once again,  $DMUM_S$  seems more appropriate for real time applications due to the computational cost of computing Deriche gradients.

## 4 Conclusions

This paper describes a new method to compute distance maps from unthresholded magnitudes that includes an inexpensive anisotropic filter. It is suitable for still images and real time applications. The approach is conceptually simple and can be easily reproduced following the proposed algorithm. Similarly to classic DT, DMUM computes a pseudo-distance from any pixel to the closest most relevant gradient accumulation.

Two different experiments were performed. A first test compared watershed segmentations created from five different methods: Sobel gradients, DT, Deriche gradients and both DMUM computed from Sobel and Deriche values. The number of created regions and the per-pixel error between segmented images and their original values was measured. It was statistically proved that the proposed approach obtains a better region-to-error ratio than the rest of considered methods, suggesting that pixel value relations are more natural in DMUM images and supporting the goodness of unthresholded methods. The proposed operator is also more stable in video sequences, obtaining the lowest pixel value differences between consecutive frames. This stability is critical in object detection or tracking schemes. It was also shown that  $DMUM_S$  was statistically as stable as  $DMUM_D$ . Being Sobel gradients much simpler to compute,  $DMUM_S$  results appropriate for real-time applications. Further image processing stages would certainly benefit from DMUM increased stability, as there exists a stronger certainty that changes in values correspond to true changes in the scene.

Different optimizations are being considered in order to improve overall speed, considering that DTs usually take place in the first stages of a visual system. Further research related to DMUM includes the application of values to Hausdorff matching for object classification and tracking, and its application to depth maps to guide object segmentation.

## References

1. Arlandis, J., Perez-Cortes, J.-C.: Fast handwritten recognition using continuous distance transformation. In: Sanfeliu, A., Ruiz-Shulcloper, J. (eds.) CIARP 2003. LNCS, vol. 2905, pp. 400–407. Springer, Heidelberg (2003)
2. Beucher, S., Meyer, F.: The morphological approach of segmentation: the watershed transformation. *Mathematical Morphology in Image Processing*, 433–481 (1993)
3. Borgefors, G.: Hierarchical chamfer matching: A parametric edge matching algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 10(6), 849–865 (1988)
4. Canny, J.: A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8(6), 679–698 (1986)
5. Deriche, R.: Using canny’s criteria to derive a recursively implemented optimal edge detector. *International Journal of Computer Vision* 1(2), 167–187 (1987)
6. Huttenlocher, D.P., Klanderman, G.A., Rucklidge, W.J.: Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15(9), 850–863 (1993)
7. Jang, J.H., Hong, K.S.: A pseudo-distance map for the segmentation-free skeletonization of gray-scale images. In: *Proceedings of ICCV*, pp. II18–II23 (2001)
8. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: *Proceedings of ICCV*, vol. 2, pp. 416–423 (July 2001)
9. Roerdink, J.B.T.M., Meijster, A.: The watershed transform: definitions, algorithms and parallelization strategies. *Fundamenta Informaticae* 41(1-2), 187–228 (2000)
10. Rosenfeld, A., Pfaltz, J.L.: Distance functions on digital pictures. *Pattern Recognition* 1(1), 33–61 (1968)
11. Rosin, P.L., West, G.A.W.: Saliency distance transforms. *Graphical Models and Image Processing* 57(6), 483–521 (1995)
12. Taylor, T., Geva, S., Boles, W.W.: Directed exploration using a modified distance transform. In: *Proceedings of DICTA*, p. 53. IEEE Computer Society, Washington, DC (2005)
13. Vincent, L., Soille, P.: Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. Pattern Anal. Mach. Intell.* 13(6), 583–598 (1991)
14. Viola, P.A., Jones, M.J.: Rapid object detection using a boosted cascade of simple features. In: *Proceedings of CVPR*, pp. I511–I518 (2001)
15. Yang, R., Mirmehdi, M., Xie, X.: A charged active contour based on electrostatics. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) *ACIVS 2006*. LNCS, vol. 4179, pp. 173–184. Springer, Heidelberg (2006)