Abstract

In this paper we present an efficient contour tracking algorithm which can track 2D silhouettes of multiple objects in extended image sequences captured by a static camera. We represent contours using cubic B-splines, and our tracking algorithm is based on tracking a lower dimensional shape-space. The tracker is coupled with a multiple model filtering algorithm which caters for objects that move with variable motion. The model based technique we provide is capable of tracking rigid and non-rigid object contours with good accuracy.

1. Introduction

Visual contour tracking is becoming an important area of research within the computer vision and image processing community. Many researchers have provided object tracking algorithms to suit a variety of applications. Among the many algorithms reported in the literature, the work of Baumberg et. al. [1], Blake et. al. [3-5] and Cootes et. al. [6] are noteworthy and have close relation to our method presented in this paper.

Our contribution in this paper is to propose a decomposed non-rigid shape-space tracking procedure coupled with a multiple model filtering (MMF) algorithm. The decomposition technique achieves dimensionality reduction while the MMF caters for tracking variable object motion (model switching). The tracker is capable of tracking multiple objects simultaneously having the prospect of real time implementation possible.

This paper is organised as follows: Section 2 gives an overview of contour and shape space representations. Section 3 provides the shape space decomposition method. Section 4 introduces the multiple model filtering method employed for tracking. Section 5 describes the tracking mechanism. Section 6 outlines occlusion modeling. Section 7 and 8 gives the results and conclusion respectively.

2. Contour Shape Representation

Blake et. al.'s [3],[5] pioneering work in proposing B-splines to represent shape contours has made contour tracking in general much efficient and reliable than the better know "snake" approach [7]. Their proposal of tracking lower dimensional spaces (shape spaces) than the actual contour has computational advantage. The relation between a shape space and a B-spline contour can be given by the following expression,

\[ \mathbf{Q} = \mathbf{WT} + \mathbf{Q}_o \]  \hspace{1cm} (1)

where, \( \mathbf{Q}_o = (X_o, Y_o)^T \) is a template shape (assumed known), and \( \mathbf{Q} = (X, Y)^T \) is any B-spline approximated contour containing the control point coordinates \( X = [x_1, \ldots, x_n]^T \), \( Y = [y_1, \ldots, y_n]^T \), and \( T \) is the 'shape space'. For a planar affine shape, \( \mathbf{W} \) can be defined by:

\[ \mathbf{W} = \begin{pmatrix} 1 & 0 & X_0 & 0 & 0 & Y_0 \\ 0 & 1 & 0 & Y_0 & X_0 & 0 \end{pmatrix} \]  \hspace{1cm} (2)

where \( 1 = [1, 1, \ldots, 0]^T \), \( \theta = [0, 0, \ldots, 0]^T \)

Unfortunately shape matrix (\( \mathbf{W} \)) is limited only to rigid affine transformation (6 degrees of freedom). For non-rigid object tracking, \( \mathbf{W} \) has to be extended to accommodate for non-rigid shape deformations. We provide a method to efficiently extend the matrix \( \mathbf{W} \) based on the work reported in [1], and further propose a decomposition of shape matrix which has proven to be more efficient and convenient. The details of which are explained in the following sections.

2.1 Non-rigid Contour Shape Representation

Cootes et. al. [6] provided a 'shape training' method termed Linear Point Distribution Model (LPDM). Each training shape (in spline space) is represented as \( \mathbf{Q} \). Where each point is the position of the \( i \)-th 'landmark point' (specific point) on the training shape. The training shapes are then aligned using a Generalised Procrustes Analysis technique [1]. The weights are chosen so that more significance is given to more 'stable' landmark points. This process results in a mean shape-vector \( \overline{\mathbf{Q}} \) and a set of aligned training shape vectors \( \mathbf{Q}_i \). Such a process provides a curve representation as,

\[ \mathbf{Q} = \overline{\mathbf{Q}} + \mathbf{Pb} \]  \hspace{1cm} (3)

where \( \mathbf{P} \) is a \( 2N \times m \) matrix whose columns are the \( m \) most significant eigenvectors of the shape deformation covariance matrix (see [1,6]). \( \mathbf{b} = [b_1, \ldots, b_m]^T \) is a shape parameter vector with \( m \) coefficients. By changing \( b \) within limits, different shapes can be obtained. Given an aligned shape vector \( \mathbf{Q} \), the minimum least squares approximation to the shape in the model space is given by a linear projection,

\[ \mathbf{b} = \mathbf{P}^T (\mathbf{Q} - \overline{\mathbf{Q}}) \]  \hspace{1cm} (4)

2.2 Extension of the Shape Matrix

If we assume that the template shape \( \mathbf{Q}_o \) is no longer a fixed shape, but can be given by Eq. (3), then we can rewrite Eq. (1) as follows,

\[ \mathbf{Q} = \mathbf{WT} + (\overline{\mathbf{Q}} + \mathbf{Pb}) \]  \hspace{1cm} (5)
This expression can be expanded, and using simple mathematical techniques can be shown to be [9]:

\[ Q = W_s T_r + \bar{Q} \]  

(6)

where,

\[ W_s = \begin{bmatrix} 1 & 0 & \bar{Q} & P_x & -P_y \\ 0 & 1 & 0 & \bar{Q} & P_y \\ 0 & 0 & 1 & \bar{Q} & 0 \\ 0 & 0 & 0 & 1 & \bar{Q} \end{bmatrix} \]  

(7)

which is of size \((2N)x(6+2mj)\), \(\bar{Q} = (\bar{Q}, \bar{Q})'\) is the mean shape of the training set and the columns of \(P = (P_x, P_y)'\) are the \(m\) most eigenvectors. \(T_r\) is the extended shape-space.

In this expression (Eq. 7), the shape matrix includes non-rigid shape subspaces as additional columns. The interpretation of the elements of \(T_r\) in terms of planar transformation of the mean shape is:

\[ T_r = [u, u, f \cos \theta - 1, f \cos \theta - 1, f \sin \theta - f \sin \theta, f \cos \theta, f \sin \theta]' \]

where translation \([u, u]\), scaling \([f]\) and rotation \([\theta]\) are in relation to the mean shape of the training set.

A potential problem that appears with the form of \(W_s\) while tracking \(T_r\) (as state vector) using a Kalman filter based tracker is that, it causes linear dependence between the elements of \(T_r\), especially for larger inter-frame shape deformations. This leads to unacceptable tracking results [9]. In such a situation \(W_s\) in its current form cannot be used effectively. To avoid such a problem we propose a decomposition method of \(W_s\) into suitable components so that each component of \(T_r\) can be tracked independently. This process is discussed next.

3. Decomposition of \(W_s\)

Where the inter-frame shape changes (rigid and non-rigid) are no longer small, it is best to separate each transformational effect. The result of such a decomposition procedure is given by Eq. (8), which can be shown to be equivalent to Eq. (6). See [9] for details.

\[ Q = \begin{bmatrix} 1 & 0 & T_r & \bar{Q} & P_x & -P_y \\ 0 & 1 & 0 & T_r & \bar{Q} & 0 \\ 0 & 0 & 1 & T_r & \bar{Q} & 0 \\ 0 & 0 & 0 & 1 & \bar{Q} \end{bmatrix} + \begin{bmatrix} P_x & P_y & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \]  

(8)

Each grouped component of \(T_r\), (translation, scaling, rotation/shear and non-rigid shape changes) can now be defined as a separate state vector and tracked independently using a multiple model filter (explained next). This procedure provides better numerical stability and maintains linear independence among the columns of the shape matrix. In this form (Eq. 8), for the rigid part of transformation (assuming a constant velocity model), only a 2 element state vector is required for each state parameter, thus limiting the maximum matrix size to 2x2 in the tracking filter update calculation. In the case of non-rigid shape transformation, we assume that each shape parameter vary independently (and the noise process is isotropic). With this assumption (also supported in [1]) it can be shown that each element of shape parameter vector \(h\) can also be decomposed into \(m\) independent shape parameters. Thus each shape parameter now requires only a 1D Kalman filter for tracking (assuming a constant position motion model), requiring only scalar calculations in the filter update process (details in [1,9]).

The decomposition method reduces the computational burden of the tracking system (described next) to a great extent, thus making real time tracking applications possible. Another distinct advantage of using separate filters is that each filter can employ a combination of different motion models. Such a system provides automatic model switching capability for the tracker.

4. Multiple Model Filtering

Most contour tracking algorithms assume a constant motion model. This assumption can lead to unsatisfactory results when the object of interest moves with variable motion [8]. For example, a pedestrian can initially be walking then running or even stand stationary. To capture all possible motions, the tracker needs to adapt to one or a combination of many motion models. Such a model switching technique can be achieved by incorporating an MMF algorithm into the tracking framework. We have incorporated the Interacting Multiple Model (IMM) algorithm [2] into our tracking process for automatic model switching (due to space details are omitted). The IMM has been proven successful in numerous target tracking projects [2,8]. In this paper we demonstrate how IMM can be applied in a visual tracking scenario.

In the IMM approach, at time \(k\) the state estimate is computed under each possible current model using \(r\) filters (assuming there are \(r\) different motion models in a bank of filters), with each filter using a different combination of the previous model-conditioned estimates (mixed initial condition). Such a system can cope with motion changes very effectively [8].

4.1 Dynamic Model

Each of the filters we employ (ie, translation, scaling, rotation/shear and shape parameter) is combined with an IMM algorithm. Each filter can be operating with a different motion model/s at any one time. The dynamic system for each of the filters is assumed to evolve around an AR process, which in general is given by,

\[ T(t_k) = \sum_{j=1}^{M} A_j T(t_{k-1}) + B_j w_k \]  

(9)

where \(M\) is the order of the model [eg: \(M=3\) is a Constant Acceleration Model (CAM), \(M=2\) is a Constant Velocity Model (CVM), \(M=1\) is a Constant Position Model (CPM)]. \(A_j, B_j\) are all \(N_f \times N_f\) size matrices (where \(N_f\) is the number of columns of \(W_s\)) and \(w_k\) is Gaussian noise.

Equation (9) can be expressed more compactly by defining a 'state vector,

\[ X(t_k) = (T(t_{k-1}), T(t_{k-1}), T(t_k))' \]

which can then be written as,

\[ X(t_k) = AX(t_{k-1}) + Bw_k \]  

(10)

We have employed 3 motion models (\(M=3\)) for the IMM filter bank for our experiments.
5. Tracking Procedure

5.1 Measurement Model

The effectiveness and reliability of the tracker depends on reasonable measurements being applied to the tracker. Since our image sequence is captured by a static camera, measurements of foreground objects (moving) are obtained by image differencing and thresholding using a background frame. This process separates all the moving objects of the scene. Now each moving object of a reasonable size is separated using a ‘search and separate’ procedure [9]. A reference point for each moving object is assigned on the extracted boundary (could be the top most point, bottom most point etc.). Starting from the reference point, N control points are assigned (equally spaced) along the contour of the object in a clockwise manner. This procedure is followed for every frame with the same number of control points for each moving object. We now assume that for a given object, control point N_j for frame k will now be used as measurements for estimating the actual contour shape at frame (k+1).

The measurements for each separate filter (translation, scaling, rotation/shear, and shape changes) are gathered from the overall measurements, so that each tracking filter is fed with only the measurements that are relevant to them. Example the translation filter ‘sees’ only the mean shape translated through the sequence. Similarly the scaling filter ‘sees’ only the mean shape scaled. Similar procedure is followed for the other filters (See [9] for complete details of measurement process).

5.2 Prediction Model

The prediction phase is applied once at each time step. It should be noted that the filter update equations [9] apply to each separate filter for each different motion model in the IMM filter bank. Following the prediction, for a given time step, the measurements are applied (as explained before). For each measurement, the curve estimate is updated as follows:

\[ \hat{X}_{k|k} = \hat{X}_{k|k-1} + K_d U \]

where \( U \) is the innovation: difference in corresponding parametric B-spline control points as applicable to each separate filter. The Kalman gain \( K \) and the state covariance matrix are updated as for a standard Kalman filter (see [9] for details).

6. Occlusion Modeling

Occlusion is detected at the image differencing and thresholding stage. If a connected foreground region exceeds a set height/width ratio or a sudden change in the centroid of the object is observed, then the tracker goes into occlusion mode. During occlusion mode measurements are disregarded. The current object shape is placed onto the estimated centroid position in the next frame. By doing so the tracker keeps track of the number of objects and recovers from occlusion well (see Fig. 1d for illustration).

7. Results

The results of the tracker are displayed in Fig.1 & 2. Figures 1(a-c) illustrates the ability of the tracker to cope with changes in translation, scaling, rotation and non-affine shape changes effectively. Fig.1d shows the tracker’s ability to cope with occlusion. Fig.2(a-d) shows the track accuracy for each affine transformation (for waving hand sequence). Fig. 2(e-g) demonstrates the model switching aspects of the tracker. The tracker automatically switches motion model when the hand changes motion, which can’t be effectively achieved by employing a single motion model based tracker. Further accuracy in shape fitting can be achieved by using more number of control points for each object (currently \( N=32 \)). More number of shape parameters can also result in better quality shape (currently only 10 principal components are used which account for ~90% of non-rigid shape variations). The tracker performance has been compared with Blake et. al.’s tracker [3] and Baumberg et. al.’s tracker [1]. The results show that for large interframe displacement of objects, our tracker produces better quality results.

8. Conclusion

We have provided a novel contour tracking method for applications where the interframe displacements can no longer be considered small. Tracking of a walking person is a classic example of such a case, and we have demonstrated the ability of the tracker to track non-rigid objects that move with variable motion. The tracking algorithm in general can be applied to any deformable object in motion.

References

Fig. 1: Some frames of long image sequences with tracked contours superimposed (white outline). (a) Outdoor pedestrians tracked by using a static camera, note, despite partial occlusion of the person in left (frame 4), the tracker retains the visible shape. (b) Indoor walking man tracking, illustrating the tracker's ability to cope with scale changes. (c) Waving hand moving with variable motion. With automatic model switching capability, the tracker is able to track the silhouette of the hand correctly. (d) Modelling occlusion: Despite occlusion (frame 3), the tracker manages to identify both objects separately and recovers well from occlusion.

Fig. 2: Waving hand sequence results. (a) True and estimated position (x & y) of the object centroid. (b) True and estimated object centroid velocity. (c) True and estimated scaling changes of the hand (in relation to the mean shape). (d) True and estimated rotational changes of the hand (in relation to the mean shape). (e) Motion model switching probabilities for the translation filter; observe that when the hand reaches a stationary position a constant position model is in operation. (f) Model switching probabilities for the scaling filter, a CVM is in operation most of the time. (g) Model switching probabilities for the rotational filter, a CAM is preferred over other models to cope with erratic rotational changes.
A color Contour Detector based on Dempster-Shafer Theory

Michel Chapron
ENSEA-ETIS
6, Avenue du Ponceau
95014 CERGY-France
e-mail : chapron@ensea.fr

Abstract

Segmentation based on contour detection is a relevant stage before image interpretation or pattern recognition. This paper is concerned with a new color contour detector founded on Dempster-Shafer theory. It proposes a method for computing the orientation and magnitude of a gradient defined on color images.

1 Introduction

There are more and more researches on color image processing. The progress of technology (cameras, memories and fast computers) permits to research more easily and efficiently on color image processing. We think that although color image processing, handles more data, it provides a richer representation of objects and can simplify their recognition and positioning. In this paper, we focus our attention on the design of a new chromatic edge detector based on Dempster-Shafer theory. Then, the image can be segmented from the contours. The quality of segmentation process is very relevant for high level image processing such as image interpretation.

At the beginning, some background on color image processing is described, then the proposed method is explained. Finally, the results and conclusion are given.

2 Background

The essential rules of colorimetry have been given by Grassman in 1853. These laws are:
- any color can be created by three other colors and this combination of the three colors is unique,
- if two colors are equivalent, they will be again equivalent after multiplying or dividing the three components which make them up by a same number,
- the luminance of a mixture of colors is equal to the sum of the luminances of each color.

The Commission Internationale de l’Eclairage (1931) [1] determined the tritimus values that represent the color basis in which any color can be expressed. This color basis is for the blue $\lambda_b = 425.8$ nm, the green $\lambda_g = 546.1$ nm and the red $\lambda_r = 700.0$ nm. This is the reason why the three images (red green, blue) are taken for processing color images. This commission also defined linear and non linear transformations between different color basis.

In this paper, we are interested in the problem of the computing the fusion of contours obtained on blue, green and red components of color images. The purpose of this study is to propose a method for combining the different gradient magnitudes of red, green and blue images in order to obtain a more reliable color gradient magnitude. We assume that if there is a color gradient in a pixel, there exists a gradient at least in one of the three component images at this pixel. The basic mass assignment and the mass combination law [1, 2] are computed for finding the maximum of credibility in order to find the gradient magnitude and gradient orientation. A work on region segmentation which uses the dempster-Shafer theory has been presented in [3]. We use a model which looks like the Appriou’s models [4]. This proposal is an alternative to Di Zenzo’s [5] solution for the problem of combining the gradient magnitudes of red, blue and green images. The local maximum gradient pixels provide the contours.

Usual methods of color image segmentation consist in finding the regions of the same color in the picture. For doing this, different kinds of methods can be used. The pixels of the image can be classified into several classes, each of which represents certain values of color features. The drawback of these methods is that they lose geometrical information which is useful for example when several regions at different places in the image have the same color features. There are also methods which consist in splitting or/and merging parts of the image until obtaining regions each of which has the same color features. The drawback of these methods is that they don’t always locate the contours well. Some methods which combine the merging of regions and the use of local histograms give good results but the computation is time-consuming. Other methods which combine the contour and region approaches of segmentation provide good results by consuming time.

A lot of work has been made for getting the contours. A simple way to detect contours is to apply a gradient on every chromatic component and give the sum or the maximum of the gradient magnitudes computed on the red, blue and green images to the result gradient. Below, we propose a new chromatic gradient based upon Dempster-Shafer theory which is an alternative to Di Zenzo’s method [5].
From results of differential geometry, Di Zenzo designed a gradient for multispectral images. He used theoretical results and proposed a combination of three chromatic gradients for getting a global gradient.

With Di Zenzo's notation, we have the following:

Let \( x \) be any pixel, \( f \) the global luminance function on the 3 images, we set:

\[
\mathbf{x} = (x^1, x^2, x^3)
\]

\[
\mathbf{y} = f(x) = (f^1(x), f^2(x), f^3(x))
\]

\( f^1(x), f^2(x), f^3(x) \) correspond to the grey-level value of pixels located at \( x \) in red, green and blue components.

\[
\frac{\partial f_j}{\partial h} \text{ is supposed to be of rank 2 everywhere in the image with } h=1,2 \text{ and } j=1,3 \text{ and }
\]

\[
f_h(x) = \left( \frac{\partial f^1}{\partial h}, \frac{\partial f^2}{\partial h}, \frac{\partial f^3}{\partial h} \right)
\]

\( f_h(x) \) and its first derivatives are assumed to be continuous.

Let \( x \) be the scalar product in \( \mathbb{R}^3 \). For \( h,k=\{1,2\} \)

\[
g_{hk}(x) = f_h(x) \cdot f_k(x)
\]

Let us notice that \( \{ f_h(x) \} | h=1,2 \) is a 2-dimensional vectorial space basis, tangent to \( \mathbb{V}_2 = \{ f(x) \in \mathbb{R}^2 \} \). The 4 numbers \( g_{hk}(x) \) are the components of the symmetric tensor \( g(x)=g(x^1,x^2) \) of rank 2.

We have to maximize:

\[
df^2 = \sum_{h=1}^2 \sum_{k=1}^2 g_{hk} \; dx^h \cdot dx^k
\]

This problem can be seen as the problem of determining the value which maximizes the form:

\[
P(\theta) = g_{11} \cos^2(\theta) + g_{12} \sin(\theta) + \frac{g_{22}}{2} \sin^2(\theta)
\]

The maximum is obtained when \( \frac{dP(\theta)}{d\theta} \) is equal to 0. It corresponds to:

\[
\theta = \frac{1}{2} \arctan \left( \frac{2g_{12}}{g_{11} - g_{22}} \right)
\]

where \( \theta + \frac{\pi}{2} \) are solutions. We keep the one which maximizes \( P(\theta) \). The direction of the gradient is \( \theta \) which corresponds to the maximum. \( g_{11}, g_{12} \text{ and } g_{22} \) can be written for color image processing as follows:

\[
u(x) = \frac{\partial f^3}{\partial x^2} g + \frac{\partial f^3}{\partial x^1} b
\]

\[
v(x) = \frac{\partial f^3}{\partial x^2} g + \frac{\partial f^3}{\partial x^1} b
\]

\[
\begin{align*}
g_{11} &= \left( \frac{\partial f_1}{\partial x^1} \right)^2 + \left( \frac{\partial f_2}{\partial x^1} \right)^2 + \left( \frac{\partial f_3}{\partial x^1} \right)^2 \\
g_{12} &= \left( \frac{\partial f_1}{\partial x^2} \right)^2 + \left( \frac{\partial f_2}{\partial x^2} \right)^2 + \left( \frac{\partial f_3}{\partial x^2} \right)^2 \\
g_{22} &= \left( \frac{\partial f_1}{\partial x^3} \right)^2 + \left( \frac{\partial f_2}{\partial x^3} \right)^2 + \left( \frac{\partial f_3}{\partial x^3} \right)^2
\end{align*}
\]

I took in a previous work the multiscale Canny-Deriche's gradient instead of Sobel's gradient proposed in the Di Zenzo's paper.

3. Proposed method

3.1 Evidential definitions

For each source \( s_k \) (color component image), we have to find the masses \( m_{sk} \). Appriou proposes to give \( N \) a priori probabilities \( P_{sk}(H_i) \) for each hypothesis \( H_i \). The coefficients \( q_{ik} \) represent the degree of confidence placed on the probabilities \( P_{sk}(H_i) \). In order to get the masses from these a priori probabilities, Appriou proposes 2 models:

Model 1

\[
m_{sk}(H_i) = q_{ik} \cdot p_{sk}(H_i)
\]

\[
m_{sk}(H_i) = q_{ik} \cdot (1-p_{sk}(H_i))
\]

Model 2

\[
m_{sk}(H_i) = 0
\]
Appriou shows that these 2 models satisfy the theoretical framework.

The frame of discernment has

The full ignorance is represented by: me = 0, 0 ≤ e ≤ 1.

The belief on A over θ is defined as follows:

Bel (A) = \sum \frac{m_θ (B)}{B ⊂ A}

The plausibility is defined as:

Pl (A) = 1 - Bel (A) = \sum \frac{m_θ (B)}{A ∩ B ≠ φ}

We have:

Bel (A) ≤ P_θ (A) ≤ Pl (A) where P_θ (A) is the probability of A.

No a priori knowledge is needed. The Dempster Shafer’s rule increases the belief on events which are consistent with the data from different sources (component images) and on the contrary the belief of events decreases when they are not consistent with the data.

The mass assignment is performed as follows:

\[ m_θ (A) = \prod_{s_1} m_θ (A) \oplus \prod_{s_2} m_θ (A) \oplus \ldots \oplus \prod_{s_M} m_θ (A) \]

\[ m_θ (A) = K_θ \sum \frac{M}{A_1 \cap A_2 \cap \ldots \cap A_M = A} \prod_{k=1}^{M} m_θ (A_k) \]

With:

\[ K_θ = \frac{1}{1 - k_θ} \]

\[ k_θ = \frac{\sum_{A_1 \cap A_2 \cap \ldots \cap A_M = φ} M}{\prod_{k=1}^{M} m_θ (A_k)} \]

For color image processing M is equal to 3. k_θ represents the degree of conflict between the sources.

This combination is commutative and associative.

The decision-making procedure consists in choosing the most credible hypothesis H_i such that:

\[ Bel (H_i) = \max_{i \in [1, N]} {Bel (H_i)} \]

3.2 A method for computing the orientation and magnitude of total gradient

12 different 5x5 gradient windows are applied to each pixel of the red, green and blue images. The maximum absolute value of these 12 different windows provides the gradient magnitude and the orientation belonging to the interval \[-\pi, \pi / 2\].

In order to find the best orientation to each pixel, we keep for each color component the orientation found above plus its 2 orientation neighbours. The belief masses attributed to the 3 orientations of each color component is proportional to their 3 gradient magnitudes.

Then, the Dempster-Shafer’s normalized mass combination is applied to the orientation masses obtained on each color image component. The referential set of orientations is the same on the 3 basis colors. This allows to get the global orientation by computing the maximum of credibility.

For obtaining the total gradient module, a simple method would be to take the maximum of the 3 color gradient magnitudes. A more robust method is to get the global gradient magnitude by the belief theory. For each color image, we keep the maximum of gradient magnitude and its two neighbours (in the sense of direction). The 3 masses attached to each pixel in a color image are normalized.

Then, the masses from the 3 color components are combined by using Dempster-Shafer’s combination.

Taking 3 directions in each image plane reduces the combinatorial explosion which is a drawback of Dempster-Shafer theory.

Finaly, the overall gradient magnitude is found by computing the maximum of credibility.

3.3 Detection of local maximum of gradient magnitudes

It permits to find the contour pixels from the global gradient. For doing this, Dempster-Shafer’s theory is again utilized. On the direction previously obtained, we can determine on each color image the 3 normalized masses attached to the 3 hypothesis: H corresponds to the hypothesis that the current pixel is a local gradient maximum, H̄ corresponds to the hypothesis that the current pixel is not a local gradient maximum and { H, H̄ } corresponds to the lack of knowledge. The mass m(H) is computed by testing the gradient magnitude of the current pixel with the four pixels on the global direction obtained above in the 5x5 neighborhood. The 2 tests with the
nearest neighbors have a weight 0.2 while the 2 farthest have a weight 0.1. These tests consist in comparing the gradient magnitude of the current pixel with the gradient magnitudes of its neighbors. If all the 4 tests show that the pixel processed has a gradient magnitude greater or equal than its four neighbors, the mass value of \( m(H) \) is 0.8. For the computation of \( m(\tilde{H}) \), we do the same except the test which is less than instead of greater or equal when comparing the gradient magnitude of middle pixel with its 4 neighbors. The mass \( m(\overline{H}, \tilde{H}) \) is equal to 0.2. It corresponds to the uniform places of the input image and to small differences when comparing the gradient magnitudes. Testing is faster than computing all the differences which will be integrated in a formula for obtaining the mass. Finally, the maximum of credibility is computed and if it is high the pixel is a local gradient maximum and therefore it is a contour pixel.

4. Results and conclusion

Dempster-Shafer's theory provides the values orientation and magnitude of a color gradient. Figures 1 and 2 show an input image and the result of the proposed color gradient. This proposed technique can be easily generalized to multispectral images for finding the contours. It works well. And another type of marginal gradient can be taken.

5. References