Classifying Deformable and Non-deformable Video Objects

Wael F. Youssef, Siba Haidar, Philippe Joly

Classifying Deformable and Non-deformable Video Objects

W.F. Youssef*, S. Haidar †, P. Joly #

* SAMOVA Team, IRIT, Paul Sabatier University, Toulouse, France, waelfyoussef@gmail.com
† Faculty of Sciences, Lebanese University, Beirut, Lebanon, siba.haidar@ul.edu.lb
# SAMOVA Team, IRIT, Paul Sabatier University, Toulouse, France, joly@irit.fr

Keywords: video analysis; video database indexing; object classification; fundamental matrix; motion classification.

Abstract

This paper presents a fully automated approach to classifying deformable and non-deformable moving objects in a video surveillance scene. We estimate an object's motion using Marzat optical-flow algorithm. We filter the motion vectors and attempt to find the transformation that represents the correct mapping between the two positions. The Fundamental transformation is estimated using the Normalized Eight-Point Algorithm. We introduce a new type of graph to set the thresholds between deformable and non-deformable motion. Furthermore, we use temporal consistency to classify deformable and non-deformable objects. For experiments, we used a varied corpus of real surveillance videos. Our proposed approach for motion classification achieved a precision rate of 92 percent.

1 Introduction

In the recent years, the concerns about security in public places are rising, especially with the increase of terrorist's threats and the spread of all kind of crimes. "Modern and forever advancing surveillance camera technology can be a valuable tool in the management of public safety and security, in the protection of people and property, in the prevention and investigation of crime, and in bringing crimes to justice"[1]. But this valuable tool has two main limitations: 1- The all-time shortage of human operators compared to the needed number of actively monitored cameras; 2- The extreme difficulty faced when forensically investigating the enormous recorded video database. Therefore, the recent need identified for video surveillance systems and research is the semantic video understanding and indexing by automatic video analysis. For that purpose, it is especially important to recognize and study the video content—i.e., the background, actions, objects, and their movements—to better understand their meaning. Accordingly, object properties are of considerable importance. One property that significantly facilitates the understanding of object movement is object deformability. In many research works, object deformability is a mandatory prior piece of information which is not actually automatically extracted. The existing Intelligent Video Analytic Softwares (IVAS) are hardly dealing with all kinds of video surveilled objects (Humans, animals, machines, etc…). For that, detecting whether an object is deformable or non-deformable would allow a tracking system and an IVAS to rely on more appropriate measurements. There are few related works that deal with this problem [1, 2].

This study presents a new fully automated method for classifying deformable and non-deformable objects. It aims mainly to deal with video-surveillance content—specifically in scenes recorded with a static grayscale or colored camera and where there is only one moving object in the scene.

A deformable object is an object that, when in motion, can undergo shape deformations, for example, a walking man, or a running animal. A non-deformable object, by contrast, has a rigid shape, for example, a passing car, an opening door. We define temporal motion as a fragment of an object motion for a small number of successive frames. "Non-rigid motion" is standardly used to refer to all articulated, elastic, and fluid motion, denoted here "deformable motion". Likewise, rigid motion is denoted as "non-deformable motion". Importantly, deformable objects can have both deformable and non-deformable motion, whereas non-deformable objects are restricted to non-deformable motion.

In Section 2, we explain our approach. We then present, in Section 3, the experiments we have done in order to validate and evaluate our method.

2 PROPOSED APPROACH

In the real world, a general moving object has displacement, for example, from position A_{3D} to position B_{3D}. Its features correspond at both positions, as do the points along its surface. This displacement can be represented by 3D motion vectors $\{\vec{V}_i, i: 0\ to\ n\ldots\}$. In a video, using a general projective camera, this object is projected on image planes (of different positions A_{2D}, B_{2D}, C_{2D} …). Subsequently, 3D motion vectors $\vec{V}_i$ are projected to 2D motion vectors x_i from frame position A_{2D} to frame position B_{2D}, where each vector represents the displacement of a pixel from one image to another. This gives the corresponding points $x_i \leftrightarrow x'_i$, where x_i and x'_i are the two extremities of the vector $\vec{v}_i$.

When a static camera is used, the background estimated motion in the frames will be a null vector. Moreover, because there is only one moving object in the scene, the motion-estimation will point out the object movement represented by motion vectors.

This process begins by deciding, for each temporal motion, whether the displacement between time t_1 and t_2 is deformable or not. In the case of non-deformable object motion, there will be a particular transformation to map x_i to its corresponding

[1] Surveillance Camera Code of Practice. This code is issued in England and Wales by the Secretary of State under Section 30 of the 2012 Act.

x'_i. Later, we will attempt to calculate this transformation, which is the Fundamental matrix. The temporal displacement may be deformable or non-deformable. However, in each of the above cases, the object can be either. Thus, we studied the temporal consistency of the displacements to determine whether the object is deformable or non-deformable.

To summarize, first, we detect object movements and estimate the motions vectors in the scene. Then, we filter the motions vectors. Next, we search for the Fundamental transformation matrix, if there is one, which satisfies these movements. Subsequently, we determine whether the transformation correctly maps the two sets of corresponding points. By reference to this, the decision is made about the detected temporal motion as to whether it is deformable. Finally, from the sequence of the temporal deformability of movements, we can infer the deformability of the moving object. This step will ultimately classify the object moving through the scene, as deformable or non-deformable.

2.1 Motion estimation

For our study, a reliable method is needed—one that can produce a very dense, accurate (to the extent of using sub-pixels), and regular field of vectors representing the pixel displacements of a moving object. Moreover, the capability to track each moving object pixel through frames is required. This must be combined with the ability to estimate any kind of movement, even slow movement or object rotation. In addition, for further application and interpretation of an action, it is not possible to sacrifice the availability of dense and regular information in order to avoid missing any part of the object's body.

Many approaches for motion estimation (Optical flow approaches, feature-based approach, block matching…) were well examined and tested. To achieve our main goals with best results, when no prior information about the content of the scene is available—and with a minimum number of hypotheses, assumptions, and constraints—an optical-flow method is adopted. Also, we found that Marzat's algorithm [3] suits better this type of studies. Marzat's algorithm presents a pyramidal implementation of the Lucas-Kanade method [4] with regularized least squares (i.e., a multi-resolution approach) and in plus an iterative and temporal refinement. Still, estimating motion with Marzat's algorithm requires filtering to ameliorate its results and to remove unreliable motion vectors.

2.2 Motion filtering

The purpose of this step is to eliminate all unreliable motion vectors data that have not been already filtered by Marzat's algorithm. The false-positive appearance of vector movements in uniform areas is the first case that can be detected. The second case appears when the detected vectors are parallel to the local texture, meaning that any estimation of those vectors will be erroneous. In addition, especially small vectors are insignificant to further interpretation. Then, three simple filters are used: the small-vectors filter, uniformity filter, and texture filter.

1) Small-vectors Filter: All insignificant vectors with a very small abscissa and ordinate (<0.5) are eliminated (e.g., the motion of tree leaves), noise, or poor detections.

2) Uniformity Filter: Where there are uniform areas in the frame, Marzat's algorithm detects false-motion vectors. Thus, if a vector exists in a uniform area, it will be eliminated.

3) Texture Filter: Marzat's method uses the Lucas-Kanade approach, which is based on motion-vector estimation according to a gradient calculation that can result in false estimations for vectors, especially along edges where vectors appear to be parallel with the local texture. Thus, we find intensity variations in the vector surrounding block, in the direction and orientation of this motion's vector. If the average intensity variation is low, the motion's vector is in the same direction and orientation as the local texture. These vectors must be eliminated.

2.3 Transformation

We consider a kinematics theory of non-deformable bodies. For general 3D non-deformable static bodies, a well-known transformation exists between two corresponding points (X_i and X_i') taken from two different camera positions at two different times. This case is equivalent to the case of one static camera taking two images of a 3D non-deformable moving body at two different times. Thus, when a non-deformable object is observed in two perspective-camera views, its feature correspondences satisfy an epipolar constraint for a general non-deformable body. The transformation is called the Fundamental matrix. Where :

$$X_i'^T . F . X_i = 0, i = 1..N \quad (1)$$

In this study, we use the normalized 8-point algorithm (N8PA) [5, 6] to estimate the Fundamental matrix, because it provides adequate results and because it is quick and easy to implement. In real images, the position of points x_i and x_i' is perturbed by noise, leading to errors in image measurements for both images. Moreover, the F estimation may not be accurate. Therefore, the epipolar constraint (1), applied to 2D, is not fully satisfied, then: $x'_i{}^T . F . x_i = \varepsilon \neq 0$ *where* $i = 1 \cdots N$ and ε is an error. We did use the *Symmetric Epipolar Distance* [6] to calculate this error:

$$\mathrm{Er}_i^F = \mathrm{d}(\mathrm{x'_i}, \mathrm{F.x_i})^2 + \mathrm{d}(\mathrm{x_i}, \mathrm{F^T.x'_i})^2 \quad (2)$$

2.4 Deformable and Non-deformable Motion

Ideally, the transformation F is a perfect mapping of the corresponding points, and F can correctly map all N points x_i to x'_i and vice versa. However, because of real images errors, the estimation of F will not be perfectly accurate. In such cases, two issues are taken into consideration:

- *The error margin when mapping the corresponding points* ($\mathrm{x_i} \leftrightarrow \mathrm{x'_i}$): calculated using the Symmetric Epipolar Distance (Er_i^F), or by the mean distance (Er'^F_i):
$$Er'^F_i = (d(x'_i, F.x_i) + d(x_i, F^T.x'_i))/2 \quad (3)$$
if $Er_i^F \leq \gamma_F$ (respectively, $Er'^F_i \leq \gamma'_F$) then F correctly maps the couple $x_i \leftrightarrow x'_i$; where γ_F and γ'_F are the *mapping error thresholds* calculated later.
- *The error margin in the percentage of correctly mapped points:* establishing the acceptable percentage of points

that are not correctly mapped, even though both sets of corresponding points are considered to be correctly mapped, in general.

To consider the two sets of corresponding points $x_i \leftrightarrow x'_i / i = 1..N$, as correctly mapped, it will be sufficient if the percentage of the correctly mapped points is more than a certain threshold: $\delta_F = (N' \times 100/N)$ with $N' \leq N$ is the number of correctly mapped points.

Moreover, δ_F should be generalized as much as possible so that they can be applied for all types of objects (deformable, non-deformable, small, medium, and large, with texture, smooth, etc.) and movements (slow, medium, fast, small, large, in all directions, etc). This must be accomplished in such a way that, whatever the object is, and for any temporal movement between the two frames im_{n-1} and im_n, the two sets $2N$ of filtered and corresponding points can be found. Then, F can be estimated; subsequently, the percentage of correctly mapped points p_F can be found. Finally, if $p_F \geq \delta_F$, then the transformation represents a correct mapping. As a result, the temporal movement of the object is classified as non-deformable motion. Else, it is deformable. *For that,* δ_F should be investigated as to whether they can be affected by the following two parameters:

- *Number of motion vectors.*
- *Average Length of the Motion Field (*ALMF*).*

Based on tests, it was clear that the number of motions vectors does not seriously affect the motion non-deformability threshold δ_F. Only a few vectors (8 vectors) are needed to define and represent the true temporal displacement of the object. Thus, the density of the motion field can be reduced in order to diminish the time required for filtering.

Concerning the length of the motion, initially in the experiments, the mapping error threshold for F is fixed regardless of the motion length, and different lengths of motion can significantly affect δ_F in such a way that the smallest average length of the motion field will have the highest motion non-deformability threshold for F to minimize the errors in discriminating between non-deformable and deformable. Therefore, seeking the generality, a normalization step is added to normalize the length of the motion field after motion filtering and before calculating the transformation F. For that, all motion vectors are normalized to an average motion-field length equal to n (n=1, 2, 3, 4 … round (original average length)).The Fundamental matrix F_n was calculated for each of the normalization level n. Therefore, for each normalization level n, the mapping error threshold (γ_F^i, γ'^i_F) must be found in a way to lead to the ultimate motion non-deformability threshold (δ_F^i, δ'^i_F).

In paragraph 3 the method of searching for thresholds is explained and the ultimate couple mapping error threshold (γ_F^i, γ'^i_F) and the motion non-deformability threshold (δ_F^i, δ'^i_F) are found in a way that maximizes the success (the percentage of success) of the algorithm. The ultimate thresholds are shown the Table 1. It should be noted here that, for small object movement, deformable motion can be confused with non-deformable motion in the real world. Furthermore, the length of the motion vectors and the difference in length among motion vectors are very small. Thus, the Fundamental matrix F and the motion non-deformability thresholds are unreliable. Moreover, by having especially long movement vectors, errors in estimating the motion vectors and in estimating F will be duplicates, and the motion non-deformability thresholds will be again unreliable. Following the experiments, the ALMF should fall between seven and ten. For that, the motion vectors inputted during the third step (viz., transformation) should have an average length between seven and ten. By changing (i.e., by eloigning or approaching) the input-compared frame im_i (i.e., the frame compared with the current frame im_n) and repeating the first and the second steps (viz., motion estimation and motion filtering), the desired average length of the motion field can be obtained.

Normalization	***1***	***2***	***3***	***4***	***5***
$(\gamma'^n_F, \delta'^n_F)$:	(0.6, 83.36)	(1, 79.13)	(1.4, 76.92)	(1.8, 76.04)	(2.2, 75.52)
$\%S$:	81.56	82.16	82	82.05	82.04
Normalization	***6***	***7***	***8***	***9***	***10***
$(\gamma'^n_F, \delta'^n_F)$:	(2.8, 76.65)	(3.8, 80.91)	(4.8, 82.56)	(5, 81.06)	(6, 90.76)
$\%S$:	82.32	82.93	82.68	80.98	80.51
Normalization	***1***	***2***	***3***	***4***	***5***
$(\boldsymbol{\gamma_F^n}, \boldsymbol{\delta_F^n})$:	(0.6, 81.31)	(2.2, 80.16)	(3.8, 76.6)	(6.6, 76.25)	(15, 81.31)
$\%S$:	81.8	82.58	82.09	82.41	82.79
Normalization	***6***	***7***	***8***	***9***	***10***
$(\boldsymbol{\gamma_F^n}, \boldsymbol{\delta_F^n})$:	(25, 83.08)	(35, 83.02)	(47, 82.61)	(51, 81.26)	(72, 81.18)
$\%S$:	82.92	83.12	82.64	81.45	82.44

Table 1: Ultimate thresholds: where *(x,y): x is the mapping error threshold, and y is the motion non-deformability threshold; below these thresholds is the corresponding percentage of success (%) Deformable and Non-deformable Objects.*

2.5 Deformable and Non-deformable Objects

Until now, object motion has been classified independently for each frame. Now, the entire series of the object's apparent motion should be considered. When an object appears in frames, the series of its temporal motion (Motions[2]: $X_i, X_{i+1}, X_{i+2} \cdots, X_j \cdots, X_n$) will be studied and classified as deformable or non-deformable motion (like $D_i, D_{i+1}, ND_{i+2} \cdots, D_j \cdots, ND_n$), where D/ND stand for deformable/ non-deformable.

Two criteria should be taken into consideration:

- *Errors in classifying the temporal motion:* the temporal motion can be misclassified owing to errors in the motion-estimation algorithm and transformation-estimation errors caused by image noise, etc.
- *A deformable object can have non-deformable motion:* A deformable object can be mistaken for a non-deformable object if it acts as a non-deformable object for a period. This occurs when the articulations of a deformable object

[2] The motion (i.e., the temporal motion) is denoted according to the frame of its motion vectors and the destination frame. For example, the displacement of the object from frame X_k (the suitable corresponding frame of X_j for the study) to frame X_j ($X_k \rightarrow X_j$) is called motion X_j.

are hidden at the time of motion or have the same displacements as an entire body.

First, the temporal consistency will be studied when the motion is classified for all series of movements to correct classification errors, and to exclude the inconsistent non-deformable movements in a deformable object. Second, object deformability will be inferred.

In this study, we used the temporal-consistency algorithm proposed by [7]. Their goal was to improve the results obtained for an object detector operating independently on each frame of a video document; the results for the object detector are smoothed along the time dimension using a temporal window of size *N,* centered on the subject frame. Then, only the objects whose number of appearances is above a threshold (N_2) are validated. Since the aim is classification, there are no misdetections, but only correct or false classifications. As a result, the algorithm in this work was subject to a few modifications: e.g., X and Y used in the maximization equation are assumed to be independent in [7], which is not the case in our study. Therefore the equation becomes:

$$\underset{N,N_2}{Arg\ max}\, P[(Y < N_2)\ \cap (X \geq N_2)\ ] = \underset{N,N_2}{arg\ max}\, P(X \geq N_2)\, P\left(\frac{Y < N_2}{Y \leq N - N_2}\right) \quad (4)$$

where X is the number of correct classifications in N frames, and Y is the number of false classifications. In [7], p_d is the probability of success, $q_d = 1 - p_d$ is the probability of failure, p_f is the probability of a false alarm in a frame, and $q_f = 1 - p_f$; in our case (without misdetection), $p_f = q_d = q$, and $q_f = p_d = p$.

Finally, the maximization equation will be:

$$Arg\ max_{N,N_2}\, P[(X \geq N_2)\ \cap\ (Y < N_2)] = \begin{cases} \underset{N,N_2}{arg\ max}\left(\sum_{i=N_2}^{N} C_N^i p^i q^{N-i}\right)\left(\sum_{i=0}^{N_2-1} C_N^i q^i p^{N-i}\right),\ N_2 \leq {}^{N}/_{2} \\ \underset{N,N_2}{arg\ max}\left(\sum_{i=N_2}^{N} C_N^i p^i q^{N-i}\right)\left(\sum_{i=0}^{N-N_2} C_N^i q^i p^{N-i}\right),\ N_2 > {}^{N}/_{2} \end{cases} \quad (5)$$

To find the optimal values for N and N_2 that suit our aim, the numerical resolution proposed in [7] was used to maximize the expression. We considered the case of the Symmetric Epipolar distance as the distance measure, normalization level two, the mapping error threshold γ_F^2 =2.2, and the motion non-deformability threshold δ_F^2= 80.16, table 1 give us a probability of success $p = 82.58$ and a probability of failure $q = 100 - 82.58 = 17.42$.

However, the set of solutions was a plateau, and a solution was found that can be generic to several applications. Thus, the couple (N, N_2) taken is: (N, N_2) = (*11, 6*), where, $P[(Y < N_2)\ \cap (X \geq N_2)\ ]$ is maximized to *0.988454*).

This step can be reiterated as needed, until the final output is completely smooth and stable. When temporal-consistency was applied, it increased the percentage of success by more than 6%, see the examples in Table 2, below. When applied a second time, the *percentage of success* (percentage of true classification) increased to more than 91.8%.

The final step in classifying the object is simple; we classify the object as deformable or not by looking on the motion-classification series of its appearance. If all the persistent motion classifications are non-deformable, then the object is non-deformable. However, the existence of one sub-series of deformable classifications is sufficient for the object to be classified as deformable.

	% of true classification	*% of false classification as deformable*	*% of false classification as non-deformable*
Before temporal consistency	**82.58**	**9.06**	**8.36**
After temporal consistency	**89.025**	**6.025**	**4.95**

Table 2: The temporal-consistency amelioration results: *tested on 75 different videos (2141 frames), using the Symmetric Epipolar distance, and the normalization level 2, where γ_F^2 =2.2 and δ_F^2 = 80.16.*

3 EXPERIMENTS

Because we could not find any real public dataset particularly dedicated to the study of deformable and non-deformable object classification with which to compare our proposed method, we created our own dataset by filming[3] some of the videos, and collecting others from **real police video-surveillance cameras**. In addition, we did not find, in any related research, any source code that could be used to test our dataset for the purpose of comparison. We should also mention that, because many of the videos used in our experiments were taken from real police video-surveillance cameras, they were not diffusible.

We tested our approach on 75 color videos containing 30 different scene types with more than 2,100 tested frames. All videos were taken using a static (surveillance) camera. The objects in these videos had diverse properties. They differed in: nature, resolution, distance from the camera, motion speed and lighting conditions. Thus, this dataset is considerably diverse, and this makes it ideal for our purposes, insofar, as the motion in these videos is especially difficult to classify. In Fig. 1 and Fig. 2, we show two typical examples: The "Highway 2" scene, and the "Walking" scene.

For better understanding, we explain how to derive the motion non-deformability thresholds when the mapping error thresholds are fixed. Then thresholds are improved when we apply the ultimate thresholds with variable mapping error thresholds.

3.1 Mapping Error Thresholds Fixed to 1

In all the experiments, we used both error distances. For simplicity we will be talking of only one. Let the *mapping error thresholds* $\gamma_F^n = 1$, where n is the normalization level (n =1…10). For each object motion in the scene, the percentage of correctly mapped points (p_F^n) is calculated. Let m be the number of motions tested for any given object appearance in the scene, and let M_m denote the set of all these motions: $M_m = \{X_1, \ldots X_i \ldots X_m\}$. For each X_i we have $p_F^n i$.

The set M_m contains deformable and non-deformable motion. Each motion is classified manually, as deformable or not, by reference to its movement between the two corresponding frames, and in a critical and rigorous way. For

[3] http://www.irit.fr/recherches/SAMOVA/CORPORA/DND/DNDO.zip (30/08/2016).

example, if only a small part of a human body (e.g., a part of a hand) is moving in a manner different from the body, regardless of whether this motion was correctly estimated with the Marzat optical flow, the object is considered deformable.

Let D_r denote the sub-set of M_m, with all the manually classified deformable motions. Similarly, sub-set ND_s contains all the manually classified non-deformable motions, where r and s are the cardinalities for D_r and ND_s, respectively, such that $r + s = m$.

The *motion non-deformability thresholds* (δ_F^n) differentiate between these two sub-sets. Ideally, all motions in ND_s must have $p_F^n i \geq \delta_F^n$. Similarly, all motions in D_r must have $p_F^n i < \delta_F^n$. When the *motion non-deformability thresholds* (δ_F^n) are found, they can be used for any kind of video, object, or motion. However, when working with real images, we cannot find the *motion non-deformability thresholds* that completely split the two sub-sets D_r and ND_s. Therefore, we settle for thresholds that conform to the following two conditions concurrently:

- The biggest number of motions in ND_s have their own percentages of correctly mapped points ($p_F^n i$), above the corresponding threshold. In other words, we retain the "best maximum" of non-deformable motions above the threshold, and an "acceptable minimum" of non-deformable motions below the threshold.
- Similarly, the biggest number of motions in D_r have their own percentages of correctly mapped points ($p_F^n i$), below the corresponding threshold.

We introduce a new kind of graphs. We call it the "Best Maximum – Acceptable Minimum Graph". A graph for the motion non-deformability threshold (δ_F^n) is obtained as follows:

- The sub-set ND_s of the non-deformable motions is sorted in descending order, according to the percentage of correctly mapped points for each frame in the sub-set.
- On the other hand, the sub-set D_r of the deformable motions is sorted in ascending order, according to the percentage of correctly mapped points for each frame in the sub-set.
- A percentage is given for each element in the two sub-sets, representing its placement within the sub-set. This value is called the "Placement Percentage". For example, the 5^{th} element in ND_s will be given the percentage $(5 \times 100)/s$, and the 5^{th} element in D_r will have the percentage $(5 \times 100)/r$.

Fig. 3 shows the graph for a normalization level of *2,* using the mean distance, with a *mapping error threshold* $\gamma'^2_F = 1$.
Here the requested threshold t is a value of y=t, where:
((maximum of points in Series 1 are above line y=t) ⋂ (maximum of points in Series 0 are below line y=t)).

With this type of graph, the intersection of the two curves represents the best existing solution, where the maximum number of non-deformable motions (in ND_s) have their percentages of correctly mapped points above this coincidence point, and the maximum number of deformable motions (in D_r) have their percentages of correctly mapped points below this coincidence point.

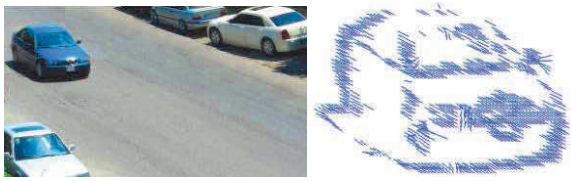

Fig. 1. Scene from "Highway 2": (left) Frame 97, (right) Zoomed motion vectors between frames 97 and 96; (755×2 corresponding points).

Fig. 2. Scene from "Walking": (left) Frame 83, (right) Zoomed motion vectors between frames 83 and 80; (365×2 corresponding points).

For example, in Fig. 3, if we settle for y= 70, rather than the intersection point, we know that 97% of non-deformable motions are above this threshold, and consequently well classified. However, only 62.6% of deformable motions are below this threshold, meaning that only 62.6% are well classified. Alternatively if we take y=t=79.13 (the ordinate of the intersection point), then 82.16% of deformable and 83% of non-deformable motions are well classified, and this is the optimal percentage.

Let $I(s,t)$ be the intersection point, with t denoting the requested threshold. Notice that the abscissa, s, for the point of intersection $I(s,t)$ represents, in this case, the percentage of success for the entire algorithm, insofar as the number of deformable motions and the number of non-deformable motions that are tested are approximately the same. Moreover, the Placement Percentage is the same for both series *ND* and *D*. Accordingly, we calculate the *motion non-deformability thresholds* δ'^n_F for each normalization level (see Table 1).

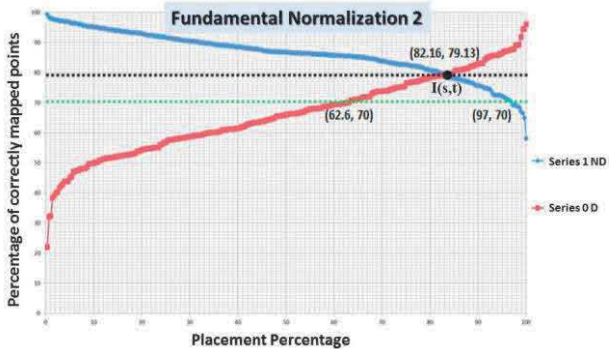

Fig. 3. Graph for F_2: with a *mapping error threshold* $\gamma'^2_F = 1$, y=70, intersecting with Series 0 at 62.6, and Series 1 at 97.

3.2 Ultimate Thresholds

In the previous subsection, the *mapping error thresholds* were fixed to *1*. When the *mapping error thresholds* are variable, the search for the ultimate *motion non-deformability thresholds* can be done by finding the best intersection points by reference to the "Best Maximum – Acceptable Minimum" graph. To do this, we studied the variation of the curves'

intersection points, according to the variation in the *mapping error thresholds*.

For each normalization level n, each type of distance measure, and for F and H, we generated a graph of the variation in the curves with intersection points according to the variations in the mapping error thresholds. For example, for the normalization *3*, the mean distance, and a *mapping error threshold* of γ'^{3}_{F}, varying between *0.2* and *10* with intervals of 0.2 $(\gamma'^{3}_{F} = 0.2{:}\ 0.2{:}\ 10)$, the graph will take the form in Fig. 4. In Fig. 4, it is clear that the *mapping error threshold* $\gamma'^{3}_{F} = 1.4$ is the threshold that maximizes the *percentage of success* to *82%,* which corresponds to the ultimate *motion non-deformability threshold* of *76.92 %.* Furthermore, for each normalization level n, we calculate the ultimate corresponding couple (*mapping error threshold* and *motion non-deformability threshold*) that maximizes the *percentage of success*, using different distance measurements. The values from this calculation are found in Table 1.

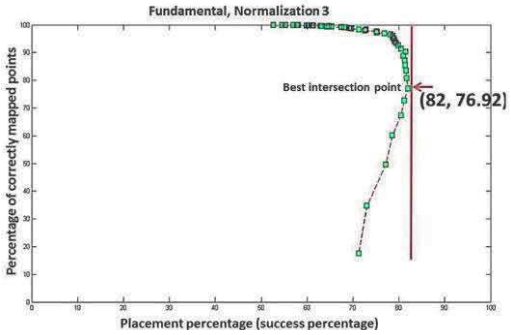

Fig. 4. Graph of the variation of curves: with intersection points according to variable *mapping error thresholds*, for F, normalization 3, mean distance, and a *mapping error threshold* of $\gamma'^{3}_{F} = 0.2{:}\ 0.2 : 10$.

The thresholds in Table 3, and the ultimate thresholds in Table 1, confirm that when the *mapping error threshold* is fixed, the best *motion non-deformability thresholds* will have decreasing values proportional to the normalization level (see Table 3). However, if the *mapping error threshold* is variable in the appropriate way, the ultimate *motion non-deformability thresholds* will have approximately the same value, regardless of the normalization level or the distance type used (see Table 1). This ensures high stability and reliability with regard to our algorithm.

Normalization	***1***	***2***	***3***	***4***	***5***
mean distance $(\%S, \delta'^{n}_{F})$:	(79.95, 92.49)	(82.16, 79.13)	(80.45, 67.22)	(79.23, 57.87)	(77.57, 49.82)
Normalization	***6***	***7***	***8***	***9***	***10***
mean distance $(\%S, \delta'^{n}_{F})$:	(74.86, 42.16)	(72.95, 36.62)	(72.2, 31.47)	(71.08, 27.3)	(73.52, 23.21)

Table 3: Temporal motion non-deformability thresholds: for each normalization when the mapping error threshold is fixed to 1 / where (x,y): x is the percentage of success (%S), and y is the motion non-deformability threshold.

On the basis of our experiments we recommend using the Symmetric Epipolar Distance, normalization level *2*, the *mapping error threshold* $\gamma^{2}_{F} = 2.2$, and the *motion non-deformability threshold* $\delta^{2}_{F} = 80.16$ (note: this is not to suggest that other thresholds are undesirable**)**. As an example, when we compared our results for the two scenes above (viz., "Highway 2," where $p^{2}_{F} = 94.1722$, and "Walking," where $p^{2}_{F} = 68.7671$) with the corresponding threshold ($\delta^{2}_{F} = \mathbf{80.16}$), we can easily infer the deformability of each corresponding motion. The scene from "Highway 2" was classified as non-deformable motion, and the scene from "Walking" was classified as deformable motion—and both classifications were correct.

4 CONCLUSION

In this study, we proposed a threshold-based decision-making system aimed at determining whether an object or its motion corresponds to a non-deformable model using geometric projection modeling. The method relies on estimating parameters of a standard geometric transformation, which could be considered as a model of non-deformable object motion. The accuracy of this transformation in representing the object motion is then analyzed to infer the actual deformability (or non-deformability) of the object. We improved the results using temporal consistency, reaching a relatively high rate of precision (approximately *92%*). Such a precision rate is largely sufficient to address new topics where knowledge about object deformability is an input.

This study can provide the video surveillance research a rigorous and precise algorithm, which can be built on when examining the video analysis and indexing. In the future, we intend to generalize this approach by addressing multiple objects on the basis of multiple-target-tracking tool. Major issues in this perspective will be to take into account partial or complete occultation and possible interactions between objects.

References

[1] A.J. Lipton, "Local application of optic flow to analyse rigid versus non-rigid motion," Carnegie Mellon University, The Robotics Institute, 1999.

[2] Ross Cutler and Larry S. Davis, "Robust real-time periodic motion detection, analysis, and applications," Pattern Analysis and Machine Intelligence, IEEE Transactions 22(8), (2000) 781-796.

[3] J. Marzat, "Estimation temps réel du Flot Optique," *Master recherche adisibi*, **ENSEM, Nancy**, (2008).

[4] B. D. Lucas, T. Kanade, "An iterative image registration technique with an application to stereo vision," *IJCAI* **81**, pp. 674-679, (1981).

[5] H.C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature 293* **(5828)**, pp. 133–135, (1981).

[6] R. Hartley, A. Zisserman, "Multiple View Geomerty in Computer Vision", *Cambridge University Press*, **second edition**, (2003).

[7] G. Jaffré, P. Joly, "Improvement of a Temporal Video Index Produced by an Object Detector," *in Computer Analysis of Images and Patterns*, *Springer Berlin Heidelberg*, pp. 472-479, (2005).