Comprehensive evaluation of tracking systems by non-photorealistic simulation

Christine Dubreu, Antoine Manzanera, Eric Bohain

Comprehensive evaluation of tracking systems by non-photorealistic simulation

Christine Dubreu[a], Antoine Manzanera[b], Eric Bohain[c]

[a] Cedip Infrared Systems and ENSTA, Paris, France;
[b] ENSTA, Paris, France;
[c] Cedip Infrared Systems, Paris, France

ABSTRACT

As more and more research effort is drawn into object tracking algorithms, the ability to assess the performance of these algorithms quantitatively has become a fundamental issue in computer vision. Because tracking systems have to operate in widely varying conditions (different weather conditions, background and target characteristics, etc), a large test bed of video sequences is needed in order to obtain a comprehensive evaluation of a tracker across the whole range of its operating conditions. However, it is very unlikely that a dataset of real video sequences representative of the whole range of operating conditions of a tracker together with its ground truth could be obtained, and building a realistic synthetic dataset of such sequences would require costly advanced simulation platforms.

In the new evaluation method proposed in this paper, the operational criteria of the tracking system are turned into objective measures and used to generate a synthetic dataset, non-photorealistic, but statistically representative of the whole range of operating conditions. The assessment of an algorithm using our method provides both a quantitative evaluation of the algorithm and the borders of its validity domain. The performance measurement of an algorithm on a synthetic sequence is shown to be consistent with the measurement on a real sequence with the same criteria. The benefit of this approach is twofold: it provides the developer with a way to concentrate on the weaknesses of his algorithm, and helps the system designer to choose the algorithm that best fits the operating constraints.

Keywords: Image processing, tracking, evaluation, simulation

1. INTRODUCTION

In recent years, there has been a considerable interest in automatic visual surveillance of wide area scenes. One of the major challenges of visual surveillance is the development of reliable tracking systems, and a variety of algorithms has been developed for the tracking of objects in environments of different complexities. Thus, performance evaluation has become an increasingly important issue, since it enables researchers to assess the reliability and robustness of their algorithms under widely varying conditions, and users to compare different algorithms and to choose the one that best fits their operating constraints.

The conventional assessment process consists in running algorithms on a dataset of sequences and comparing the results with a ground truth, typically generated by manual or semi-automatic examination of the video sequences. However, this method has two disadvantages : (i) the generation of ground truth data is highly time-consuming and subject to error and uncertainty, and (ii) it is very difficult to produce a dataset of sequences which covers the whole range of operating conditions of a system, and then very difficult for the user to circumscribe the validity domain of the system, with respect to the operating criteria (Figure 1(1)).

To handle this problem, some users employ physical simulation to produce a dataset of photo-realistic sequences. This allows them to evaluate the system on any configuration of the operating criteria, by translating these criteria within physical parameters, including 3D geometric, mechanical, and photometric modeling (Figure 1(2)). But not many users and researchers have access to such simulation platforms.

The purpose of our research is to propose a much simpler way to produce a dataset able to represent any operating condition, and for which a ground truth is automatically generated. The assumption which is made here (and that will be verified by experiments) is that the operating criteria can be turned into a set of formal parameters operating directly in the image domain to produce sequences that are not photo-realistic, but statistically representative of the operating conditions (Figure 1(3)).

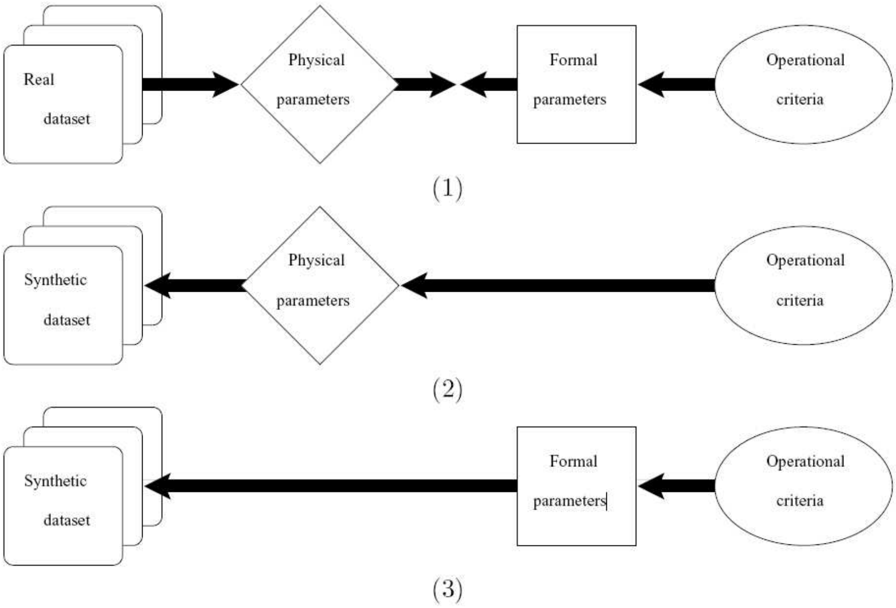

Figure 1. Evaluation data set production paradigm. (1) Real data set, (2) Photo-realistic, physical-based, synthetic data set, (3) Non photo-realistic, formal-based, synthetic data set.

The problem addressed in this paper is: given the operating criteria, what is the good (i.e. minimal, or easiest to obtain) set of formal parameters we should consider? This set should be constituted of: (1) parameters defining the inherent statistical properties of the objects involved, i.e. the target(s) and background. These will be referred to as *static parameters*, and (2) parameters defining the relative interaction between these objects, and their temporal behavior, i.e. their deformation, illumination, and relative motion, the *dynamic parameters*. Derivation of these parameters from the operating criteria and the way they are used to produce a synthetic sequence is presented in Section 2.

In Section 3, we validate the approach by evaluating two tracking algorithms on both real sequences and synthetic sequences with the same set of parameters generated with our method, and compare the results. Then, we illustrate the interest of our synthesis method, by showing that we are able to change any parameter value of a given scene at our convenience in order to generate a scene representative of any desired operating condition of a tracking system.

In Section 4, we discuss the further potentialities of the method, and also the way it can be extended and improved.

2. SYNTHESIS OF A SCENE

2.1. Set of discriminating perceptual parameters

We assume (and verify *a posteriori* that this is a correct assumption) that a set of discriminating parameters representative of a scene can be defined, from which a scene representative of any operating conditions of a tracking system can be generated and used for the assessment of this system. The aim of this paragraph is to find the static discriminating parameters of this set, i.e. the parameters inherent to the objects present in the scene independently of their orientation, motion or illumination, that fully characterize these objects.

Some of the parameters that will be determining in the behavior of a tracking system are obvious. These are the size and shape of the target, and the brightness and relative contrast of the target and background. In our application, a tracking algorithm has to perform the same way on two scenes generated with the same set of static and dynamic parameters. As the dynamic parameters may obviously contain parameters defining the objects motion, including rotation and scale, the static parameters must take into account the way the appearance of an object will be affected by those transformations, which rely on texture features, in particular coarseness, directionality and regularity. These features are related to the size, shape and organization of homogeneous regions within the texture of an object.

Therefore, the parameters used in our static model of a scene will be the size, shape, color, contrast, coarseness, directionality and regularity of the objects present in it. These are perceptual textural features which have to be estimated using computational measures, in order to get a statistical model of the objects.

2.2. Static synthesis of a scene

Texture modeling has been extensively studied in Computer Vision, and a bibliography of the literature concerning this topic was written by Tuceryan & Jain.[1] Many approaches concern models aimed at faithfully reproducing a given texture so that the synthetic texture and the sample one from which it was generated are not visually discriminable. But this is beyond the scope of this paper, and the reader can refer to the work of Haralick[2] , Derin[3] , Julesz[4] and Gagalowicz[5] for more information.

The problem we aim to solve is : given the set of parameters described in 2.1, synthesize a texture field which parameters are equal or almost equal to this set of parameters. We describe a sequential procedure to synthesize such a texture. Although the generated texture is not visually representative of an object, we will show that it is statistically representative of it, and that it can be reliably used for an accurate assessment of tracking algorithms.

The first step of this sequential procedure is to generate a synthetic texture with values on the whole histogram. In this paper, a basic texture is taken as a gaussian function, as shown in figure 2.a.

Then, the second-order features (coarseness and directionality) of the desired texture are considered. By stretching the basic texture along the two axes, a texture as directional and as coarse as desired can be obtained. This pattern is then replicated in order to get an image of fixed size, whatever the desired coarseness and directionality, as shown in figure 2.b.

Then the first-order features (grey levels distribution) are considered, a white noise is added to the obtained pattern, and its histogram is mapped to the desired histogram, as shown in figure 2.c.

After these steps, the pattern still looks very regular and artificial. In order to get a texture which is less regular, this pattern is used as a sample to synthesize an image using the algorithm described by Li-Yi Wei.[6] This algorithm, derived from Markov Random Fields texture models, generates texture through a sequential deterministic neighborhood searching process.

The algorithm starts with an input texture sample S and a white random noise I. We force the random image I to look like the sample by transforming it pixel by pixel in a raster scan ordering, i.e. from top to bottom and left to right. To determine value of each pixel p of I, its spatial causal neighborhood $N(p)$ is compared against all possible neighborhoods $N(p_i)$ from S. The value of the input pixel p_i with the most similar $N(p_i)$ is assigned to p. We use a simple L_2 norm to measure the similarity between the neighborhoods. The goal of this synthesis process is to ensure that the newly assigned pixel p will maintain as much local similarity between I and S as possible. The size of the neighborhood is a tuneable parameter corresponding to the regularity of the texture: in order to obtain a texture which is not too regular, the size of the causal neighborhood has to be small relatively to the coarseness of the texture (so that only local similarity is ensured), but if the texture is wanted to be very ordered, a larger neighborhood has to be used.

Figure 2 shows some synthetic textures generated with this method. It should be noted that some very different types of textures can be generated from the same initial pattern and different sets of parameters, and that although the generated texture still looks artificial, it is statistically representative of this set of parameters.

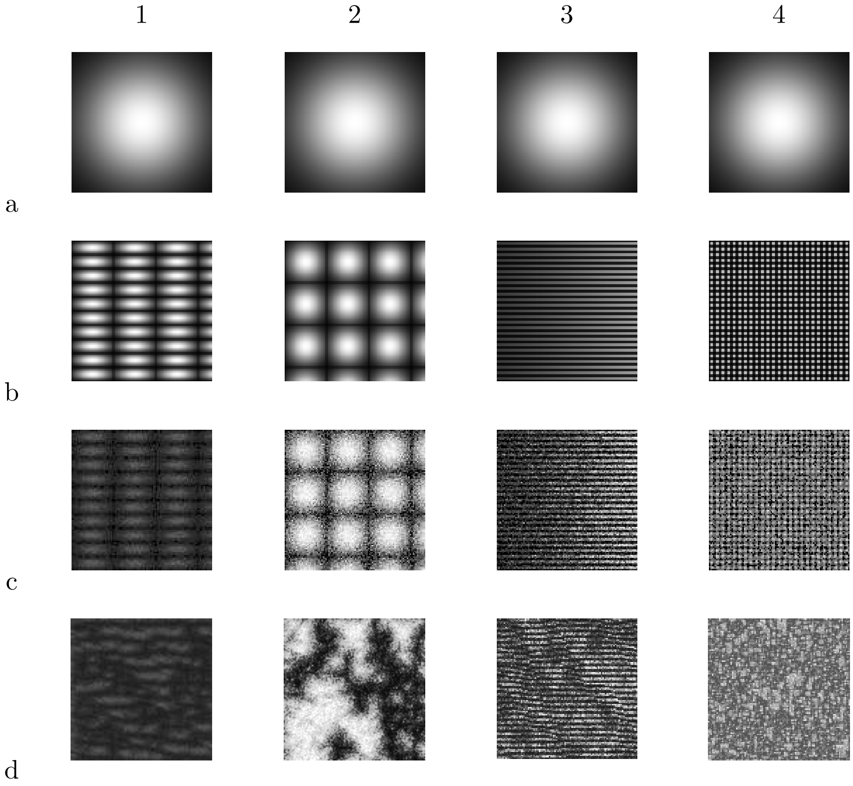

Figure 2. Texture Synthesis with different sets of static parameters using the same initial pattern. (a) Initial pattern for the synthesis of four textures. (b) Fixed-size pattern obtained by stretching of the initial pattern to get the desired coarseness and directionality. (c) Fixed-size pattern after white noise addition and histogram matching. (d) Result of texture synthesis with fixed-size neighborhood using (c) as sample.

The textures generated with this method are simple and homogeneous textures, but more complex textures, i.e. textures made of several primitives, e.g. trees and roads, can be generated using the method of texture synthesis from multiple sources described by Wei.[7]

Once the target(s) and background texture fields are created using this method, the target(s) field(s) are mapped on an ellipse with eccentricity and size corresponding to the input shape parameters, and the background field is mapped to a rectangle which size is the desired size of the video sequence. Thus, from the set of static parameters defined in 2.1, a scene with the same parameters was generated.

2.3. Set of discriminating dynamic parameters

After the static target(s) and background textures are generated, their deformation and relative displacement have to be modeled in order to get a dynamic synthetic scene statistically representative of a real scene. The background motion is modeled by a composition of a rotation, a translation, and a scale :

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \rho \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} T_{\mathrm{X}} \\ T_{\mathrm{Y}} \end{pmatrix} \tag{1}$$

Therefore, the parameters $\theta^{\mathrm{B}}_{\mathrm{max}}$, $\rho^{\mathrm{B}}_{\mathrm{max}}$, and $\mathrm{T}^{\mathrm{B}}_{\mathrm{Xmax}}$ and $\mathrm{T}^{\mathrm{B}}_{\mathrm{Ymax}}$, defining respectively the maximum rotational angle, scale, and displacement of the background between two consecutive frames have to be defined. Similarly, the target(s) motion is defined by the parameters $\mathrm{T}^{\mathrm{T}}_{\mathrm{Xmax}}$ and $\mathrm{T}^{\mathrm{T}}_{\mathrm{Ymax}}$, and its deformation by $\theta^{\mathrm{T}}_{\mathrm{max}}$, $\rho^{\mathrm{T}}_{\mathrm{Xmax}}$ and $\rho^{\mathrm{T}}_{\mathrm{Ymax}}$.

The displacement and deformation of the objects present in the scene are chosen to be modeled in 2 dimensions. This is consistent with the target(s) and background dynamics between two frames. This model would also enable us to model static occlusions, i.e. occlusions of the target(s) by a part of the background, by adding a 'transparency' parameter to the model of target.

2.4. Dynamic synthesis of a scene

From the models of objects obtained in 2.2, and the displacement parameters defined in 2.3, a synthetic dynamic scene is computed in which the model of dynamic target, i.e, the static model of target to which a dynamic motion and deformation is applied, is superimposed to the dynamic model of background texture.

A 2D deformation is applied to the target(s) at each frame. The values of the rotation angle, translation, and scale along x and y are chosen at random in the range defined in 2.3.

Similarly, a displacement and a deformation are applied to the background. To avoid discontinuities and re-synthesis of new pixels of the background at each frame generation, the background texture is symmetrized and infinitely replicated in the two directions x and y. Then the deformation and displacement can be applied to it, and no pixel has to be re-synthesized. The deformation and displacement of the background are generated in such a way that it is continuous; there is no sudden change in the angle of rotation, scale, or displacement vector, which is consistent with what is commonly found in real sequences.

2.5. Synthesis of the database

To get a dataset representative of the wide range of operating conditions of a tracking system, a large number of scenes will be synthesized with different input values. Any parameter value of a given scene can be modified; thus, a scene with any input parameters can be generated, and the contrast, motion, and deformation of the target(s) or background can be controlled.

In this paper, different of choices have been made for the simulation : only simple-textured objects are generated, their motion is simulated as described previously, and only one target is synthesized. This is because the aim of our method is to validate the assumption that only a few parameters are necessary to synthesize scenes which can be used to evaluate low-level object tracking algorithms, but some more elaborate sequences could be generated thanks to synthesis of complex textures from several samples, and control of static and dynamic occlusions by generation of multiple targets.

Indeed, our method can be combined with the one of Black,[8] also based on a synthetic dataset generation, and where dynamic occlusions can be modeled. This method constructs sequences containing complex motion scenes, by superimposing motion from isolated targets, and allows the generation of a large variety of datasets representing different tracking scenarii by controlling the occurrence and duration of dynamic occlusions.

3. VALIDATION OF THE METHOD

3.1. Validation protocol

To validate the assumption that the input parameters defined in 2.1 and 2.3 are sufficient to characterize a scene and to complete our purpose, we extract these parameters from a wide set of real sequences, and use them to generate synthetic sequences. The motion of the target(s) and background are extracted from the ground truths provided with the real scenes, the mean and contrast of the objects present in the scene are given by their histogram. Their coarseness, directionality and regularity are obtained by averaging the size and shape (ratio width/height) of regions obtained by manual segmentation of these objects.

Thus, we dispose of a wide set of pairs composed of a real and a synthetic scene with the same parameters. Figure 3.1 shows such pairs of scenes.

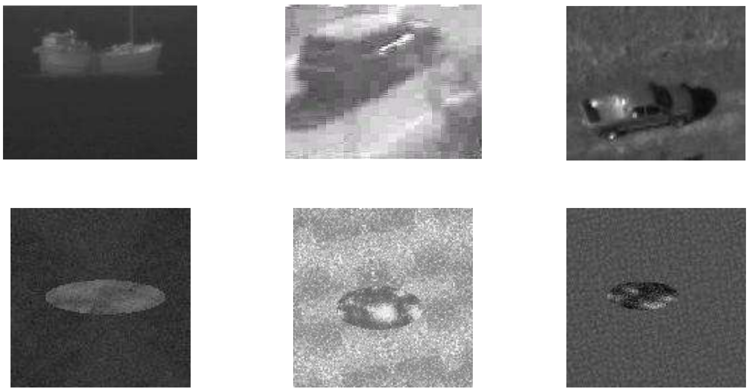

Figure 3. Pairs of images composed of a real one, and a synthetic one generated from the estimated set of parameters of the real one. The synthetic image, although non photorealistic, is statistically representative of the real one.

The real scenes we used are some infrared and visible real sequences, taken from naval, airborne, and ground cameras, for which a ground truth has been manually generated, as well as the scenes of the VIVID database,[9] which is a database of infrared and visible video sequences together with their ground truth aimed at evaluation of tracking systems.

Tracking algorithms are then applied to the real and the corresponding synthetic datasets, and the performance of the different algorithms on the different scenes are compared.

3.2. Algorithms and metric

Two algorithms are evaluated on the scenes. The first one is a correlation algorithm, and the second one is a centroid algorithm. The correlation algorithm relies on a search of the position for which the correlation value between the reference image A and a rectangular patch B in the current frame, given by (2), is maximal. The search strategy is based on a gradient descent with a diamond pattern.[10]

$$r = \frac{\sum_i \sum_j (A_{ij} - \bar{A})(B_{ij} - \bar{B})}{\sqrt{\left(\sum_i \sum_j (A_{ij} - \bar{A})^2\right)\left(\sum_i \sum_j (B_{ij} - \bar{B})^2\right)}} \tag{2}$$

The second algorithm is the centroid algorithm described by Albus & al,[11] in which a probability map is used to determine whether pixels belong to the target or not, and to determine the target's centroid. This algorithm uses concentric gates to determine relevant regions: the outer region, which contains mostly background pixels, and the inner region mostly target pixels. Therefore, a probability map can be computed from the smoothed histograms of these regions to segment the target :

$$P(k) = \frac{H_S^I(k)}{H_S^I(k) + H_S^O(k)}, \forall f \in 0...\Delta - 1, \tag{3}$$

where H_S^I and H_S^O are respectively the smoothed histogram of the inner and outer region, and Δ is the number of grey levels in the image. $P(k)$ is the probability for a pixel of intensity k to belong to the target.

A metric has to be defined for the evaluation of tracking algorithms. A diverse range of measures and procedures to establish a performance metric has been used in tracking evaluation.[12] The choice will inevitably depend on the target application, as the priorities will vary for different applications. The metric used in this paper is the standard deviation of the distance to the ground truth.

3.3. Results and validation of the model

The performance of the algorithms for a few of the real and corresponding synthetic scenes are shown in the following table:

texture pair	σ_R^{corr}	σ_S^{corr}	σ_R^{centr}	σ_S^{centr}
1. (visible cars)	0.5572	0.5401	1.034	0.9903
2. (infrared boat)	0.1473	0.1397	0.3577	0.3342
3. (visible plane)	1.0176	0.9775	0.945	1.032

σ_R^{corr} is the standard deviation of the distance to the ground truth for the correlation algorithm applied to the real sequence of the pair, σ_S^{corr} is this measure for the correlation algorithm applied to the corresponding synthetic scene. σ_R^{centr} and σ_S^{centr} are the values of this standard deviation for the centroid algorithm.

We can see that globally, the results of the algorithms on the synthetic scenes are consistent with the results of the algorithms on the corresponding real scenes for each of the two algorithms. This comforts all the assumptions made so far, and in particularly the discriminability of the set of parameters chosen, and the consistency of the database generation method.

Therefore, the low-level performance of a tracking system can be evaluated in any conditions covered by the operating criteria described in this paper, without having to find a real video sequence representative of this conditions, and without having to use costly and heavy photorealistic simulation methods.

3.4. Applications

Now that evaluating an algorithm on a synthetic scene generated with this method is known to be equivalent to evaluating this algorithm on a real scene with the same characteristics, we can assess the performance of an algorithm in any situation by running it on synthetic sequences. This is very useful for the determination of the validity domain of an algorithm, i.e. the domain on which the algorithm will perform in a satisfactory way. Figures 4, 5 and 6 show the measure of the standard deviation of the distance to the ground truth for several sets of synthetic scenes of 500 frames in which a parameter is varied, the other ones remaining constant.

In Figure 4, the contrast between the target and the background is varied, all the other parameters remaining constant. This means that for all the sequences, the displacement of the background, target, and their texture and deformation are exactly the same. Only their relative contrast is modified. For this particular case, the deformation of the target was chosen to be very low, and the background motion to be close to 15 pixels between two consecutive frames. The correlation algorithm performs well (the standard deviation of the error is low) since a very low deformation is allowed, and since deformation is the main thing, with occlusions, that would cause a correlation algorithm to fail. The centroid algorithm is highly dependent on the contrast between the target and the background, since the probability of a pixel to belong to the target depends on the inner and outer histograms. As expected, the figure shows that the centroid algorithm is worse than the correlation one for low contrasts, but it gives a quantitative information about the performance of these algorithms. If one knows the maximal standard deviation of the error accepted for an algorithm, then one can deduce the minimum contrast that would ensure the standard deviation to be acceptable, and therefore get the validity domain of this algorithm.

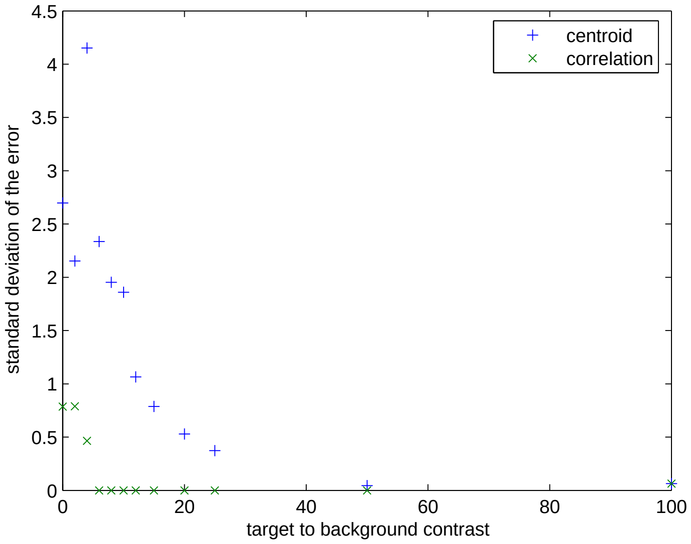

Figure 4. Robustness of the correlation and centroid algorithms to the change of contrast between the target and the background

Similarly, figure 5 gives the maximal target displacement allowed between two frames, and figure 6 its maximal rotation, the other parameters being fixed. These figures are consistent with the fact that generally, the centroid algorithm is more robust to rotation and large displacements of the target than correlation. It is the case for the specific set of fixed parameters chosen to construct these figures.

Thus, a multi-dimensional (more than 2 dimensions) circumscription of the validity domain of an algorithm can be found, and we are able to tell whether an algorithm is likely to work or not in some given conditions.

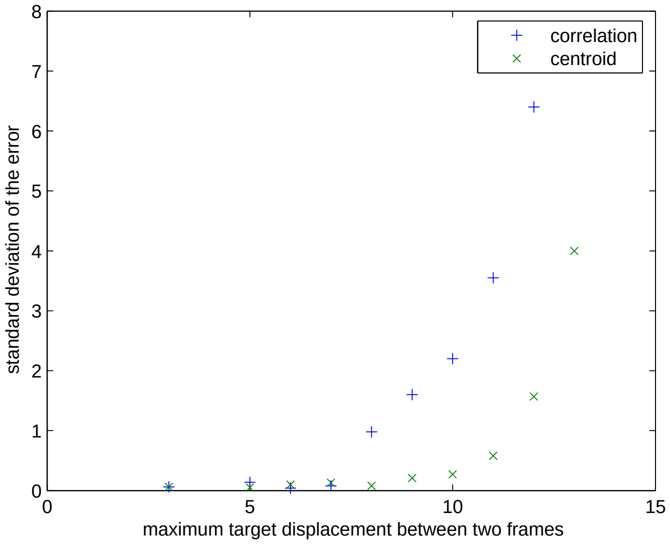

Figure 5. Robustness of the correlation and centroid algorithms to the maximum displacement of the target between two frames

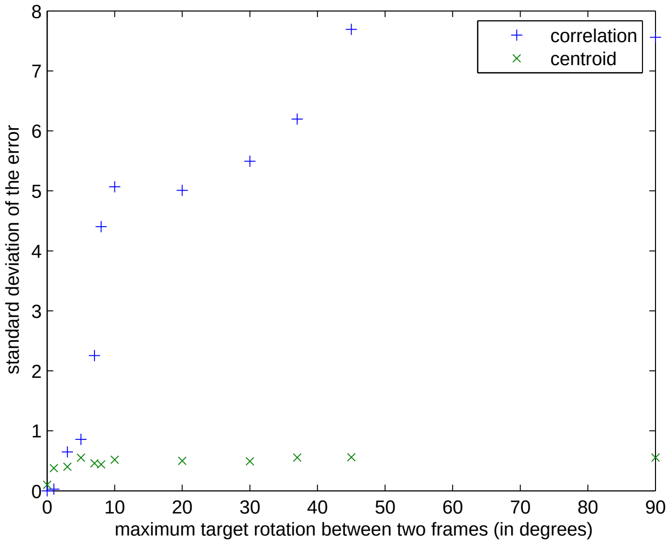

Figure 6. Robustness of the correlation and centroid algorithms to the maximum value allowed for the target rotation (in degree)

This is particularly useful for specifying the range of operation of a system .

4. CONCLUSION AND FURTHER WORK

Our method provides an efficient way to quantitatively evaluate low-level object tracking methods without having to use costly and heavy simulation platforms. It relies on the characterization of the parameters of a scene that will be discriminating for this task, and on the synthesis of a scene, which, although non photo-realistic, is statistically representative of the real scene. This enables us to get the validity domain of algorithms and to quantitatively compare the performance of different algorithms.

The evaluation method presented in this paper is not suitable for high level algorithms evaluation, since it does not deal with static or dynamic occlusions, or complex target(s) and background textures. Nevertheless, it is possible to simulate more elaborate scenes, by increasing the number of discriminating parameters. This will enable us to get a more accurate evaluation and circumscription of the validity domain of algorithms, at the cost of more input parameters.

REFERENCES

1. M. Tuceryan, A. Jain : *Texture Analysis*, in "The Handbook of Pattern Recognition and Computer Vision (2nd Edition)" by C. H. Chen, L. F. Pau, P. S. P. Wang, pp207-248, World Scientific Publishing Co, 1998.

2. R. M. Haralick : "Statistical and Structural Approaches to Textures", Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (PETS), January 2005.

3. R. T. Collins, X. Zhou, and S. K. Teh : "An Open Source Tracking Testbed and Evaluation Web Site", IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2005), January, 2005.

4. B. Julesz : "On Perceptual Analysis Underlying Visual Texture Discrimination", Biological Cybernetics, Vol 28, pp 167-175, and Vol 29, pp 201-214, 1978.

5. A. Gagalowicz : "Vers un modèle de Textures", PhD thesis, University of Paris VI, 1983.

6. L. Y. Wei : "Texture Synthesis by Fixed Neighbourhood Searching", PhD thesis, November, 2001.

7. L. Y. Wei : "Texture Synthesis from Multiple Sources", Sisgraph 2003, San Diego.

8. J. Black, T. Ellis, P. Rosin : "A Novel Method for Video Tracking Performance Evaluation", Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (PETS), January 2005.

9. R. T. Collins, X. Zhou, and S. K. Teh : "An Open Source Tracking Testbed and Evaluation Web Site", IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 2005), January, 2005.

10. A. Barjatya : "Block matching algorithms for motion estimation", 2004.

11. J. E. Albus, L. J. Lewins, J. R. Schacht : "Centroid Tracking using a probability map for target segmentation" SPIE, Acquisition, Tracking, and Pointing, April 2002.

12. T. Ellis : "Performance Metrics and Methods for Tracking in Surveillance", Proceedings of the third IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS 02), pp 26-31, September 2000.