Parametric Evaluation of Video Motion Tracking Data Sets

Mukesh Motwani, Rakhi Motwani, and Frederick Harris, Jr

Abstract—**Video tracking is a complex problem because the environment, in which video motion needs to be tracked, is widely varied based on the application and poses several constraints on the design and performance of the tracking system. Current datasets that are used to evaluate and compare video motion tracking algorithms use a cumulative performance measure without thoroughly analyzing the effect of these different constraints imposed by the environment. But it needs to analyze these constraints as parameters. The objective of this paper is to identify these parameters and define quantitative measures for these parameters to compare video datasets for motion tracking.**

Index Terms—**Benchmarking, datasets, design, metrics, video tracking.**

doi: 10.3969/j.issn.1674-862X.2010.03.004

1. Introduction

Video tracking is a very challenging problem since the tracking environment is unique for every situation. Top down design approach of a tracking system consists of the analysis of the coverage area leading to positioning of the camera, specification of pan and tilt unit, specification of the camera, number of cameras used, and design of the algorithm. There are several video tracking algorithms to choose from as highlighted in survey[1]. Currently, there is no published research which gives a guideline as to which algorithm is best suited for particular type of video motion tracking application. The publicly available datasets, such as performance evaluation of tracking and surveillance (PETS)[2],[3] which are currently used for testing the performance of motion tracking algorithms, present very limited scenarios which may not be relevant to the scenario of the application to be designed. Fig. 1 (a) is an image from a PETS dataset and Fig. 1 (b) is an image from the dataset used at the call for real-time event detection solutions (CREDS) used to test the performance in indoor environments. There are no publicly available video datasets for applications such as monitoring movement of ships in dockyard.

The current evaluation systems compare algorithms in very specific environments against a single metric related to deviation of the tracked path from the actual ground truth. This metric is cumulative and does not identify the cause of deviation or measure the deviation due to specific outlier. This can result in the failure of the evaluated algorithm in an environment with certain unaccounted conditions such as occlusion since there is no object in the scene and thus absence of ground truth. The current evaluation metrics based on track (path of object of ground truth is compared with path of object of tracker) or frame based evaluations[4],[5] are not sufficient to cover all the operating scenarios for tracking algorithms. Thus, no generic evaluation metrics exist which can be used to test the performance of tracking algorithms in the presence of outliers. This lack of analysis results in a flawed method of comparison which leads to poor selection of tracking algorithms for a system. It is a challenge to arrive at a true comparison metric for tracking systems. The metric of deviation from ground truth path can still be used and is a valid measure provided image sequences account for these variations in isolation. Thus, there is a need to create a cumulative metric which is derived from these subjective metrics corresponding to different constraints in the environment. It is imperative to identify these constraints in the environment for video tracking algorithms to be benchmarked. There is also a need to create datasets that

(a)

(b)

Fig. 1. Image sequences from various datasets: (a) PETS and (b) CREDS.

Manuscript presented at 2010 International Conference on Signal Acquisition and Processing (ICSAP 2010), Bangalore, India, February 9-10, 2010; recommended by TPC of ICSAP 2010; revised July 2010.

M. Motwani, R. Motwani, and F. Harris, Jr. are with the Computer Science & Engineering Department, University of Nevada, Reno, 13644 NV 89507, USA (email: mukesh@cse.unr.edu, rakhi@cse.unr.edu, and fredh@cse.unr.edu).

enable to test the performance of tracking algorithms, which have scenarios with these specific constraints in isolation to compute these subjective metrics corresponding to the scenarios.

The scope of this paper is limited to proposing parameters based on which motion tracking systems can be designed and compared. Using these heuristic parameters, the datasets that are already available can also be graded for the strength of each outlier. Thus, the metrics also serve as a means to compare the existing datasets. These parameters would enable to meaningfully compare tracking algorithms as well as the test datasets in spite of wide variation in testing environments.

2. Motion Background

The background of the object to be tracked in the video can be either static or moving. In case the background is static, the complexity of the background needs to be considered, such as whether the background is simple or if it contains cluttered components. If the background is not static and contains multiple motions, then the tracking system needs to decide if the motions are of objects of interest or are simply motions which should be ignored. If the background motions are not of importance, then the tracking algorithm needs to see if the motions occur close to or away from the object of interest. Associated motions in the background due to the object motion such as due to shadows or due to object debris such as left by the trail of water of a boat need to be accounted. Based upon the above scenarios of the background, the following parameters are proposed.

2.1 Multiple Motion Traceability

Consider scenario where there are multiple ships which are part of a fleet as shown in Fig. 2 with all of them moving in the same direction. The tracking algorithm may require tracking all of them. In any frame, there are multiple objects moving in the same direction at the same time. Special metrics are required to handle this parameter as conventionally used video sequences do not test this parameter[6]. The quality of multiple motions in dataset Q_{multiple} can be given as

$$\begin{cases} Q_{\text{multiple}} = 1 \quad \text{if} \quad N_{\text{tracked}} > 1 \\ Q_{\text{multiple}} = 0 \quad \text{if} \quad N_{\text{tracked}} \leq 1 \end{cases}. \tag{1}$$

The value of Q_{multiple} is zero if a scenario does not contain multiple tracked objects.

Fig. 2. Tracking multiple objects.

2.2 Immunity to Changes in Background (Distracters)

Consider the image shown in Fig. 3. If the car on the left is the object being tracked, the motion of the people or vehicles should not affect the tracking algorithm.

If there is motion in the frame other than that of the desired objects, the tracking algorithm should not deviate from its current targets. The background distracters are objects which belong to the background. It can be assumed that in this scenario the objects are of relatively small size as compared to the object to be tracked. The background objects are characterized by relative slow and local motions that need to be ignored by the algorithm in comparison to the motion of the tracked object. If N_{Distract} indicates the number of distracter objects in frame then the quality of the distracters outlier for the dataset used for testing is given as

$$Q_{\text{backgnd}} = \frac{N_{\text{Distract}}}{N_{\text{Distract}} + N_{\text{Tracked}}} \tag{2}$$

where N_{Distract} is the number of distracter objects in the frame and N_{Tracked} is the number of objects to be tracked.

The quality of the distracters can be measured in terms of the number of moving background objects. Larger the number of background objects tolerable by the tracking algorithm, better its ability is. The denominator indicates the total number of moving objects which is the sum of the number of background objects and the object being tracked itself. The value varies between 0 and 1 with 0 indicating no distracters in the frame and 1 indicating large number of distracters.

2.3 Ability to Distinguish Other Close by Motions

In Fig. 4, if the target to be tracked is the two wheelers at the centre of the frame, the motion of the vehicles around should not cause the tracking algorithm to lose track of the target no matter how small the motion may be.

Motion in the vicinity of the object being tracked may cause the tracker to lose association with the object to be tracked and may instead track the object close to it.

Fig. 3. Background objects-distracters.

Fig. 4. Distinguish motion of the object of interest.

Morphological based algorithms may not perform well in this case. The ability to reject motion of other objects in the vicinity is directly proportional to the size of the distracter object S_i calculated by the number of pixels occupied by the object and inversely proportional to the distance D_i of each object from the tracked object. Thus, as more larger objects come closer to the tracked object, the quality of the outlier increases as

$$Q_{\text{closeby}} = \frac{\left(\sqrt{W^2+H^2}\right)\left(\sum_{i=1}^{n} S_i / D_i\right)}{N_{\text{distract}} WH} \tag{3}$$

where S_i is the size of object i, D_i is the distance of object i from the tracked object, N_{distract} is the total number of distracter objects in the frame, W is the number of columns in frame, and H is the number of rows in frame.

The summation needs to be carried out for all the distracter objects in the vicinity. The multiplication by $\sqrt{W^2+H^2}$ in the numerator and division by the total number of distracter objects 'N_{distract}' and the total number of pixels in the frame WH in the denominator of (3) is in order to normalize the value of the strength between 0 and 1. This metric highlights the problem of loss of association of the tracked object. For example, if a distracter object comes in the vicinity of the object being tracked, the quality of the video sequence would be greater than zero. The larger the object and the closer the distracter object is to the tracked object, the quality factor will gradually close to 1. The video sequence should contain other foreground objects which move as fast as the tracked object and as close to the tracked object without occluding it.

2.4 Shadows

Shadows can cause problems in tracking especially when they become larger than the object being tracked. Shadows are longer in the evenings than in the noon in an outdoor environment, as shown in Fig. 5.

Shadows cause errors in spatial detection. The detection area in the presence of shadows is larger than the ground truth area of the object. To find out the performance of the motion tracking algorithm, a test sequence containing shadows of varying lengths is to be used.

Fig. 5. Presence of shadows.

Fig. 6. Cluttered background objects.

The quality of the disturbance caused by shadows Q_{shadow} can be measured in terms of the relative size of the shadow as compared to the object itself. This is measured through the number of pixels occupied by the shadow divided by the total number of pixels occupied by the shadow and the object together, as shown in (4):

$$Q_{\text{shadow}} = \frac{S_{\text{shadow}}}{S_{\text{shadow}} + S_{\text{tracked}}} \tag{4}$$

where S_{shadow} is the size of shadow computed by the number of pixels occupied by shadow and S_{tracked} is the size of the object to be tracked computed by the number of pixels occupied by object.

2.5 Presence of Complex or Cluttered Static Background

In Fig. 6, the lady is walking in a cluttered background with clothes of different colors.

This can confuse algorithms based on tracking color. Thus, the tracking algorithm needs to take into account the complexity of the background of the environment. The video to be used to assess the performance towards this parameter should contain a cluttered background with various colors. Consider N_{clutter} to be the total number of cluttered objects in the background and N_{tracked} to be the number of objects to be tracked. The quality of the clutter is given by (5) as follows:

$$Q_{\text{clutter}} = \frac{N_{\text{clutter}}}{N_{\text{clutter}} + N_{\text{tracked}}}. \tag{5}$$

This value ranges between 0 and 1 with 0 indicating a simple plain background whereas Q close to 1 indicating a very cluttered environment.

2.6 Merge and Split for Deformable/Non-Rigid Bodies

Consider a scenario in which people come together, shake hands, and move away. At the point when the bodies merge, the tracking algorithm can detect the combined bodies as one. Also, when they split, the tracking algorithm may follow the wrong body upon splitting up.

A video which simulates the above stated condition is to be used to test performance towards this parameter. Assume N_{tracked} to be the number of objects to be tracked in the ith frame and N_{distract} to be the distracter objects which merge with the tracked objects in the ith frame. The quality of the merge and split, $Q_{\text{mergesplit}}$ can be expressed as follows:

$$Q_{\text{mergesplit}} = 1 - \frac{\sum_{i=1}^{N_{\text{frames}}} N_{\text{distract}_i}}{\sum_{i=1}^{N_{\text{frames}}} N_{\text{tracked}_i}} \tag{6}$$

where N_{frames} is the number of frames of video sequence over which summation is performed.

It is assumed that the distracter objects, which may be static or moving, do not disappear from the scene. The quality of the condition is measured in terms of its occurrence. Hence, if the multiple merging and splitting occurs in the scene, the value of the quality will tend towards 1.

3. Velocity of Objects

For any tracking system, velocity of the object or variation in rate of change of velocity is an important factor to be considered in the design. These factors are described below.

3.1 Objects Moving Too Fast or Too Slow

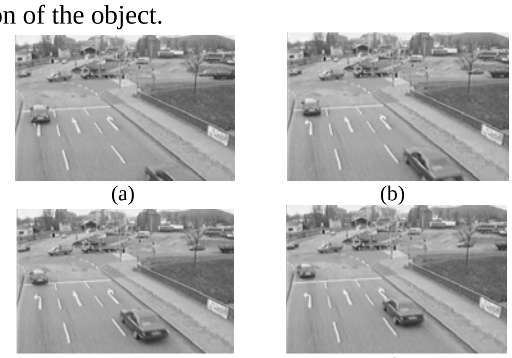
It is always a problem of tracking objects which move too fast or too slowly. For example, consider the tracking of cars in a street. It is possible that a car can move fast enough for its relative displacement in consecutive frames to be large. Fast motion of objects causes poor motion continuity and can be overcome by using larger search space and feature based tracking proposed in [7]. In case of objects which move too slowly, some algorithms which rely upon object motion tend to classify the slow objects as background and thus eliminate them as prospective tracking targets. If the object is small and its motion is slow, then it may be identified as noise. In case of objects which move too fast, if ΔD indicates the displacement of the object in consecutive frames measured in pixels and $\sqrt{W^2+H^2}$ the diagonal length of the frame in pixels, the $Q_{\text{fast moving}}$ can be measured by

$$Q_{\text{fast moving}} = \frac{\Delta D}{\sqrt{W^2+H^2}} . \tag{7}$$

The displacement is divided by the diagonal length of the frame in order to normalize the degree of motion of the object. For objects that move slowly, the $Q_{\text{slow moving}}$ will increase with a decrease in the displacement of the object. The quality of slow moving object outlier can be measured by

$$Q_{\text{slow moving}} = 1 - \frac{\Delta D}{\sqrt{W^2+H^2}} . \tag{8}$$

This ratio increases with a decrease in the rate of motion of the object.



(a) (b) (c) (d)

Fig. 7. Change in velocity of object: (a) t=0.666666 s, (b) t=0.866666 s, (c) t=1.066666 s, and (d) t=1.266665 s

3.2 Variation in Velocity of Object

In Fig. 7, it can be seen that the car is decelerating as it covers lesser distances in the time period of 0.2 s.

Algorithms which use motion history depend on the constancy of velocity or motion of the object. The measure accounts only for changes in object velocity not direction. Thus, the video to test the metric should contain velocity changes. The metric should account for the rate of change of velocity. Consider ΔD is the displacement of the centre of gravity (CG) of the object indicated in pixels and N_{frames} the number of frames over which the the velocity change is to be calculated. ΔD_i–ΔD_{i-1} gives the rate at which the CG of the object changes its velocity for frame 'i'. The average of the rate of change of velocity is calculated by summing up the value for the entire video sequence and dividing it by the number of frames in the video sequence. Then the quality of the variation in object velocity is given by

$$Q_{\text{velocity}} = \frac{\sum_{i=1}^{N_{\text{frames}}} \left(\Delta D_i - \Delta D_{i-1}\right)}{N_{\text{frames}}\sqrt{W^2+H^2}} \tag{9}$$

where ΔD_i is the displacement of the object between frame i and i–1, W is the width of frame, H is the height of frame, and N_{frames} is the number of frames of video sequence over which summation is performed.

The normalization factor in the denominator of (9) is the diagonal length of the frame.

3.3 Directionality

Consider an application of tracking the players in a soccer match as shown in Fig. 8. The objects tend to change their direction of motion suddenly. Motion in any direction in the plane of the image should be traceable irrespective of sudden changes in the direction of motion. Algorithms which use motion history, like the Kalman filter, can not work in this case.

The test conditions for this parameter should have the video contain object varying its direction of motion abruptly.

The metric for sudden directionality changes for the object(s) to be tracked measures the average number and degree of the changes in direction occurring per frame over the entire video sequence and can be express by

$$Q_{\text{directionality}} = \frac{\sum_{i=1}^{N_{\text{frames}}} \theta_i}{2\pi N_{\text{frames}}} \tag{10}$$

where $\theta_i = \frac{\mathbf{GV}_i \cdot \mathbf{GV}_{i-1}}{|\mathbf{GV}_i||\mathbf{GV}_{i-1}|}$, θ_i is the angle between ground truth velocity vectors of object in consecutive frames calculated in radians by computing the dot product, $\mathbf{GV}_i$ is ground truth velocity vector in frame i, $\mathbf{GV}_{i-1}$ is the ground truth vector in the previous frame, and N_{frames} is the number of frames over which the summation is carried out.

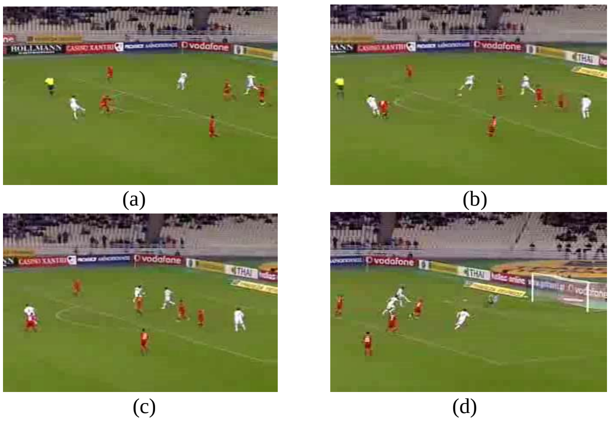


Fig. 8. Abrupt changes in direction of motion of player: (a) frame 1, (b) frame 30, (c) frame 60, and (d) frame 90.

This metric measures the frequency and degree of changes of direction for the motion of object. The summation is carried out to find out the total number and amount of abrupt changes in direction of motion of the object being tracked over the entire sequence. This is divided by the number of frames in the video sequence to give the average number of direction changes occurring per frame and 2π to normalize the amount of rotation.

4. Object Disappearance

If the object is not visible in the scene, there are two possibilities. Either it has left the field of view, or it has been hidden behind an occluding object in the field of view. Based on this consideration, following parameters are proposed.

4.1 Disappearance of Object from Scene

If the object leaves the frame, the tracking algorithm should then identify the absence of the object and indicate the same based on timeout conditions set. The algorithm should be able to decide if the object is merely occluded or has left the scene. The video sequence to test this parameter should contain a scenario in which the object of interest leaves the field of view or is occluded completely. The quality is calculated based on the number of frames for which the object is not visible in the scene.

$$Q_{\text{disapperance}} = \begin{cases} 1 \text{ if } \Delta F > \lambda \\ 0 \text{ if } \Delta F \le \lambda \end{cases} \quad (11)$$

where ΔF is the number of frames during which object to be traced disappears an then reappears and λ is the threshold number of frames during which object absence is to be identified.

The threshold λ has a value λ_1 if the object is occluded and has value λ_2 if it has completely disappeared from the scene. The value of the threshold is based on the complexity of the video sequence. If the algorithm can detect absence of the object, it has the ability to identify the disappearance of the object from the scene.

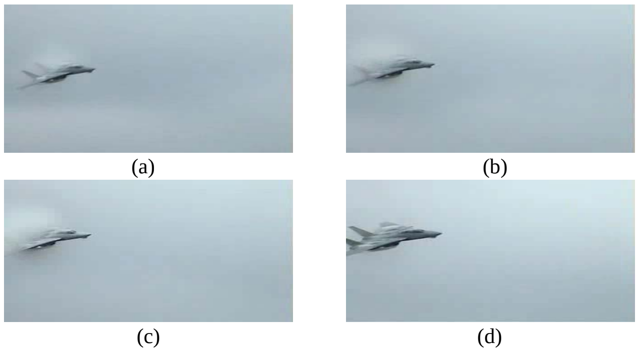


Fig. 9. Partial occlusion: (a) frame 1, (b) frame 5, (c) frame 15, and (d) frame 35.

4.2 Occlusion

Consider the airplane as the tracked object as shown in Fig. 9. The aircraft is occluded partially by the clouds.

The tracker should be able to recover from partial occlusion or even complete occlusions under certain circumstances. Thus, objects which either partially or completely obscure the objects of interest from view should not result in the losing the track of the object. Occlusions are handled using appearance models[8] the temporal information[9], or using multiple camera feeds[10].

The strength of the occlusion can be measured in terms of the percentage of the tracked object occluded. This is calculated as

$$Q_{\text{occlusion}} = 1 - \frac{N_{\text{visible}}}{N_{\text{object}}} \quad (12)$$

where N_{visible} is the number of visible pixels of the object and N_{object} is the actual number of pixels occupied by object if not occluded.

The value number of visible pixels can be easily estimated. To find out the total number of object pixels, the numbers of visible object pixels under no occlusion need to be found.

5. Lighting Conditions

In order to find out the effect of the lighting conditions on the design or for comparison of tracking algorithms, following parameters are proposed.

5.1 Outdoor Day/Night Operation

Consider the case of tracking people in an outdoor environment both during the night and the day. Typically, tracking systems using infrared cameras can operate under such conditions. The quality factor in this case is determined by the camera specifications. If the camera operates in infrared spectrum, Q=1. The algorithm should be evaluated based whether it can track effectively during night as well. As this measure computes the qualitative ability of the algorithm to track objects during the day or the night, the test sequence should contain the same scenario shot during the day as well as the night.

(a) (b)

Fig. 10. Variation in light intensity: (a) ambient during day and (b) ambient in evening.

5.2 Immunity to Variation of Intensity of Light

Changes in the ambient or local light intensity could cause a change in the appearance of the object or the background. The light source intensity could cause change in pixel values as shown in Fig. 10.

The strength of light intensity outlier can be measured in terms of the rate of change of average light intensity across the entire frame. Let L_1 be the average gray level in frame 1 and L_2 be the average gray level in the next frame 2. L_1 can be computed as

$$L_1 = \frac{\sum_{i=1}^{N} P_i}{N} \tag{13}$$

where P_i is the gray level value of pixel i and N is the total number of pixels in the frame.

Similarly calculating the value of L_2, the value of this outlier is computed as

$$Q_{\text{lighting}} = \frac{L_1 - L_2}{L_1} \tag{14}$$

where L_1 is the average gray level in frame 1 and L_2 is the average gray level in frame 2. This gives us the rate at which light intensity changes.

5.3 Change Due to Reflectance

The reflectance of the object can change if its orientation with respect to the camera changes. Also, if the light source is directional and not diffused, object appearance can change due to the relative motion between light source and object. Additionally, the object may have its own light source as shown in Fig. 11. Thus, the variability of surface reflectance over time can cause a change in the perception of the color of the object[11]. This can cause the loss of association of the object being tracked. The tracker should be able to follow objects irrespective of color changes.

The video sequence to test this outlier consists of changes in the object reflectance considering all the scenarios mentioned above. The strength of this outlier is measured in terms of the change in the local color or gray level value. Consider the image sequence to be a color sequence having 3 color channels corresponding to the colors red, green, and blue or a grayscale image. The strength of the change can be given in (15) and (16):

$$Q_{\text{color}} = \frac{\sum_{N_{\text{local pixels}}} \left(\frac{\Delta R}{R}\right) + \left(\frac{\Delta G}{G}\right) + \left(\frac{\Delta B}{B}\right)}{3N_{\text{local pixels}}\left(2^k - 1\right)} \tag{15}$$

$$Q_{\text{grayscale}} = \sum_{i=1}^{N_{\text{local pixels}}} \frac{\Delta P_i}{N_{\text{local pixels}}\left(2^k - 1\right)} \tag{16}$$

where ΔR, ΔG, and ΔB are the changes in value of color component at the pixel, R, G, and B are the values of color component at the pixel, $N_{\text{local pixels}}$ are the number of local pixels undergoing change in the value of the color components for the tracked object, ΔP_i is the change in value of pixel i, and k is the number of bits to represent grayscale value.

The values of ΔR, ΔG, and ΔB are calculated in terms of the change in the respective color components at the local pixel locations where the change has occurred over the last frame. The value of ΔP_i is similarly calculated for a grayscale image sequence. The summation is carried out to account for all the pixel locations where the change has occurred.

6. Camera Positioning

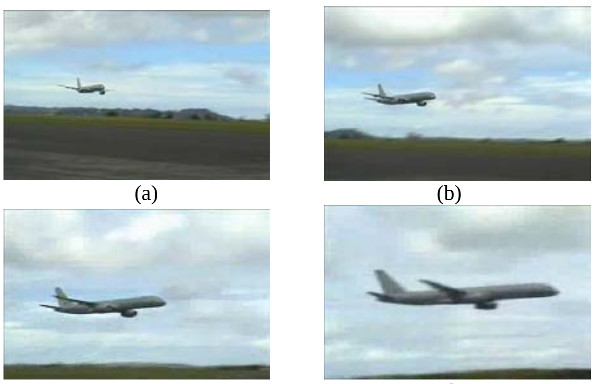
The position of the camera with respect to the object dictates the area of coverage of the scene. Following is a description of how these parameters can affect the tracking algorithm.

6.1 Area of Coverage of Tracking-Scale Immunity

As shown in Fig. 12, the size of the airplane approaching the camera keeps increasing. The tracking algorithm must be able to track objects undergoing any scale change.

The motion of the object toward or away from the fixed camera changes the appearance of the size of the object without causing any motion in the plane of the image. Algorithms like optical flow face the “aperture” or

Fig. 11. Pixel density change-object with own light source.



(a) (b) (c) (d)

Fig. 12. Change of scale of object: (a) frame 1, (b) frame 30, (c) frame 90, and (d) frame 150.

problem due to this. Algorithms should be able to adapt the size of their search windows to accommodate this change in scale. This is especially important for automatic zoom calculation for the camera. The zoom of the camera is calculated based upon the perspective size and adjusted to preserve it[12]-[14].

The strength of the change in scale is measured by the apparent change in size of the object as follows

$$Q_{\text{scale}} = \begin{cases} \dfrac{S_{\text{2pixels}}}{S_{\text{1pixels}}} & \text{if } S_1 > S_2 \\ 1 - \dfrac{S_{\text{1pixels}}}{S_{\text{2pixels}}} & \text{if } S_1 \le S_2 \end{cases} \tag{17}$$

where S_{1pixels} is the size of object in frame 1 in pixels, S_{2pixels} is the size of object in frame 2 in pixels.

The change in size is given by the magnification/shrinkage computed by finding out the ratio of the number of pixels occupied by the object in the one frame to the number of pixels occupied by the object in the earlier frame.

6.2 Objects Too Small in the Field of View

The camera position determines the field of view, which in turn determines the size of the object. If the field of view is too large, the objects may appear too small. Objects that are too small may not be detected by the algorithm. The tracking algorithm to be evaluated should be able to detect the object of interest irrespective of the size of the object in the field of view. If the selected tracking algorithm can not track distant objects less than specific size, a camera with automatic zoom should be selected. So the quality of size, Q_{size} can be expressed as

$$Q_{\text{size}} = 1 - \frac{S_{\text{bounding box}}}{S_{\text{frame}}} \tag{18}$$

where $S_{\text{bounding box}}$ is the size of bounding box and S_{frame} is the size of frame.

The strength of this outlier is measured in terms of size of the square bounding box in pixels which exactly encompasses the tracked object.

7. Change in Object Appearance

Objects in the scene can change their appearance in the course of tracking because the object is non-rigid or the pose of object changes.

7.1 Rate of Change of Deformable Bodies

A perfect example of a deformable non-rigid object is the human body or a bird as shown in Fig. 13. The motion of the wings from the body gives an impression that the entire shape of the bird has changed. Algorithms that use templates rely on matching shape of the object or optical flow algorithms that compute velocity vectors according to

(a) (b)

Fig. 13. Non-rigid bodies: (a) frame 1 and (b) frame 15.

feature points on the tracked object will have large deviation in the motion velocity vector.

To measure the quality of image sequence to determine the degree of deformation, the quality of the rate of change of deformation, $Q_{\text{deformation}}$ is given as

$$Q_{\text{deformation}} = \frac{\sum_{i=1}^{N_{\text{frames}}} \left| S_i - S_{i-1} \right|}{\sum_{i=1}^{N_{\text{frames}}} S_i} \tag{19}$$

where S_i is the size of the object in pixels in frame i and N_{frames} is the number of frames of video.

Assuming the change in deformation size is not more than the size of the tracked object, the value of the quality factor will be between 0 and 1.

7.2 Change in Shape of Body Due to Orientation

The shape of a non-rigid body may change due to its change in orientation or pose. Feature based algorithms are not suitable in such scenarios. The quality of the orientation changes $Q_{\text{orientation}}$ can be defined as

$$Q_{\text{orientation}} = \frac{\sum_{i=1}^{N_{\text{frames}}} \left(\left|\Delta\theta_x\right|_i + \left|\Delta\theta_y\right|_i \right)}{2 \times 2\pi N_{\text{frames}}} \tag{20}$$

where $\Delta\theta_x$ is the degree of rotation of object along the x axis, $\Delta\theta_y$ is the degree of rotation of object along the y axis, and N_{frames} is the number of frames over which orientation change is calculated.

The change in orientation of the object in radians over consecutive frames around the x and y axis is averaged across the entire length of the video. This is divided by 2π to give the value of the quality of the orientation changes varying between 0 and 1. A value of 0 indicates no change in the shape of the body, while a value of 1 indicates a complete change due to a change in orientation.

8. Conclusions

Heuristic measures for evaluation of tracking systems have been proposed in this paper. This paper attempts to identify in detail the different constraints imposed on a video tracking algorithm. Having identified the important parameters, a metric for each of the parameter proposed is defined. Based on these measures, a cumulative metric could be derived by weighting and adding the quality metrics of the video sequences for each outlier and use for benchmarking. These parameters will also help in the design process of a tracking algorithm and comparing existing algorithms. This cumulative metric will also lead to algorithms which would allow the tracking system to detect change in these parameters and automatically select and switch to appropriate tracking algorithms. Thus, a general purpose tracker can be constructed.

References

[1] E. Trucco and K. Plakas, "Video tracking: a concise survey," *IEEE Journal of Oceanic Engineering*, vol. 31, no. 2, pp. 520-529, Apr. 2006.

[2] D. P. Young and J. M. Ferryman, "PETS metrics: on-line performance evaluation service," in *Proc. of the 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Beijing, China, 2005, pp. 317-324.

[3] M. Krinidis, G. Stamou, H. Teutsch, S. Spors, N. Nikolaidis, R. Rabenstein, and L. Pitas, "An audio-visual database for evaluating person tracking algorithms," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Philadelphia, USA, 2005, pp. 237-240.

[4] B. Tordoff and D. Murray, "Reactive control of zoom while fixating using perspective and affine cameras," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 98-112, 2004.

[5] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang, "Framework for performance evaluation of face, text, and vehicle detection and tracking in video: data, metrics, and protocol," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 319-336, 2009.

[6] A. G. A. Perera, A. Hoogs, C. Srinivas, G. Brooksby, and W.-S. Hu, "Evaluation of algorithms for tracking multiple objects in video," in *Proc. of the 35th IEEE Applied Imagery and Pattern Recognition Workshop*, Washington DC, USA, 2006, p. 35.

[7] Y. Li, H.-Z. Ai, T. Yamashita, S.-H Lao, and M. Kawade, "Tracking in low frame rate video: a cascade particle filter with discriminative observers of different lifespans," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, USA, 2007, doi: 10.1109/CVPR.2007.383199.

[8] F. Campbell-West, H.-B. Wang, and P. Miller, "Where is it? object reacquisition in surveillance video," in *Proc. of International Machine Vision and Image Processing Conference*, Portrush, UK, 2008, pp. 182-187.

[9] J.-Y. Pan and B. Hu, "Robust occlusion handling in object tracking," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, USA, 2007, doi: 10.1109/CVPR.2007.383453.

[10] S. Velipasalar and W. Wolf, "Multiple object tracking and occlusion handling by information exchange between uncalibrated cameras," in *Proc. of IEEE International Conference on Image Processing*, Genoa, Italy, 2005, pp. II-418-21.

[11] J. P. Renno, D. Makris, T. Ellis, and G. A. Jones, "Application and evaluation of colour constancy in visual surveillance," in *Proc. of the 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, Beijing, China, 2005, pp. 301-308.

[12] C. J. Poelman and T. Kanade, "A paraperspective factorization method for shape and motion recovery," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 3, pp. 206-218, Mar. 1997.

[13] B. Tordoff, D. Murray, "Reactive control of zoom while fixating using perspective and affine cameras," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 98-112, 2004.

[14] Y. Wei and W. Badawy, "A novel zoom invariant video object tracking algorithm (ZIVOTA)," in *Proc. of IEEE Conference on Electrical and Computer Engineering*, Canada, 2003, pp. 1191-1194.

Mukesh Motwani received his B.E. degree in electronics engineering from the University of Pune, India, in 1999, and an M.S. degree in computer science from the University of Nevada, Reno (UNR), in 2002. He is presently pursuing his Ph.D. degree in computer science and engineering at UNR and works as a solutions architect to provide consulting services to the IT industry. His research interests lie in service oriented architecture, digital rights management systems, watermarking, and applied computational intelligence.

Rakhi Motwani received her B.E. degree from the University of Pune, India, in 2000, and an M.S. degree from UNR, in 2002, both in computer science. She received her Ph.D. degree in computer science and engineering from UNR, in 2010. She is currently a post-doctoral research associate with the Department of Computer Science and Engineering, UNR. She is an IEEE member. Her research interests include information hiding techniques and applied artificial intelligence.

Fredrick Harris, Jr. is currently a professor with the Department of Computer Science and Engineering and the Director of the High Performance Computation and Visualization Lab at the University of Nevada, Reno, USA. He received his B.S. and M.S. degrees in mathematics and educational administration from Bob Jones University in 1986 and 1988 respectively. He got his second M.S. and Ph.D. degrees in computer science from Clemson University in 1991 and 1994 respectively. He is a member of ACM, IEEE, and ISCA. His research interests are in parallel computation, graphics and virtual reality, and bioinformatics.