Object Detection and Motion Analysis in a Low Resolution 3-D Model

Diego F. Pava
Florida Atlantic University, Boca Raton, Florida, USA, dpava@fau.edu

William T. Rhodes
Florida Atlantic University, Boca Raton, Florida, USA, wrhodes@fau.edu

ABSTRACT

With augmenting security concerns and decreasing costs of surveillance and computing equipment, research on automated systems for object detection has been increasing, but the majority of the studies focus their attention on sequences where high-resolution objects are of interest. The main objective of the work reported here is the detection and extraction of information of low-resolution objects (e.g., objects that are so small or so far away from the camera that they occupy only tens of pixels) in order to provide a base for higher level information operations such as classification and behavioral analysis. The system proposed is composed of four stages (preprocessing, background modeling, information extraction, and post processing) and uses context-based region-of-importance selection, histogram equalization, background subtraction, biological motion analysis, and morphological filtering techniques.

The result is a system capable of detecting and tracking low -resolution objects in a controlled background scene which can be a base for systems with higher complexity.

Keywords: Background Subtraction, Low definition video, Morphological operations, Object detection, Object recognition

1. INTRODUCTION

1.1 MOTIVATION

For years, automatic video recognition of moving objects has been one of the most rapidly developing topics in video image processing due to the great variety of fields that could potentially benefit from such advancement. Over the past decade, numerous algorithms have been proposed for moving-object tracking, but a solution that clearly outperforms the human vision system is still missing, leaving room for new researchers to come up with new ideas on how to improve existing methods or develop new ones.

Video surveillance and security systems have become a topic of great importance, not only to the government, but also to industries and the general public. With the price of video surveillance dropping, it is common today to find security systems where several screens receive video feeds from cameras distributed across an area under observation (Hu et al, 2004).

Security personnel can sometimes be overwhelmed by the amount of information they must process, leading them to make costly mistakes by overlooking important information or losing time and resources on unimportant information. If the objects moving occupy only a few pixels in the screen, either because they are very small or because they are so far away from the camera, the work of security personnel without computerized help would be virtually impossible.

The task of detecting the presence of moving objects (human or vehicles for example) that are so far away as to only occupy a few pixels in the video sequence is not a simple one. Blurring of background into the image of interest can degrade information exploited with conventional techniques such as shape and color.

This work investigates systems able to detect low-resolution moving objects in video sequences and extract information from the imagery allows future algorithms to classify such objects, especially to determine if the object may be human beings.

1.2 GENERAL DESCRIPTION

The system, whose block diagram appears in fig. 1, starts with a fixed camera that sends video information to a computer. In the first stage of the system a training process is implemented where a user is asked to manually select the regions where the presence of low-resolution moving objects could mean the presence of important objects (for example humans at a great distance).

After the regions of importance are selected, the video is transformed to grayscale and enhanced using histogram equalization (Gonzalez and Woods, 2002) to emphasize contrast between moving objects and background. After this, a background subtraction algorithm is implemented. Since only moving objects are of interest, the background is then defined in this work as the common pixel information present across the frames of the video sequence For this work three different algorithmic techniques for the subtraction of background were used: a low-level-of-complexity Frame Difference algorithm (Piccardi, 2004), a mid-level-of-complexity Approximate median method (McFarlnel and Schofield, 1995) and a high-level-of-complexity mixture-of-Gaussians algorithm (Grimson and Stauffer, 1999).

After the background subtraction is performed, the video is converted to a binary image sequence, where 0 represents a background pixel and 1 represents a foreground pixel (and, thus, a part of a moving object). The resulting video has a considerable level of noise due to environmental factors (shadows, wind, rain, etc.), imperfection of the camera, and from the algorithms themselves. The noise is then reduced by using morphological filtering (Eddins et al., 2004).

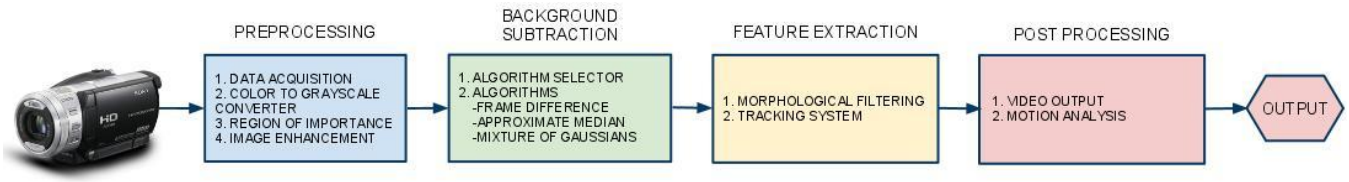

Figure 1: Block diagram

The filtered video serves as input to a tracking system that will detect and then keep information about the object (position, size, velocity, etc.) across frames even if the object is partially occluded by a background object (e.g., a tree or a wall).

The tracked information is stored in a database and motion analysis is performed by using a self-similarity matrix technique (Cutler and Davis, 2000) that tests for periodicity of the object motion. Parallel to the database, a video showing the detected and tracked objects on screen is provided as visual information for the users.

2. PROJECT DEVELOPMENT

All the image processing operations in this project were developed in MATLAB language with the input video in avi format.

2.1 REGION OF IMPORTANCE

If moving objects in low-resolution 2D video imagery are placed in their 3D context, ambiguities concerning the identity of the objects can often be removed. When objects occupy just a few pixels in a scene, there are usually

important portions of the video sequence where the presence of objects of such characteristics is unlikely or unimportant. . In the identification of objects moving in a video sequence, the availability of a 3-D model of the scene can reduce, often greatly, uncertainties in the nature of what is being observed (Pava and Rhodes, 2008).

In a complete solution, we can get better results if we can exploit our knowledge of the 3D models by creating regions of importance (ROI). Because regions of importance depend on so many factors, user-created ROIs are preferred over those automatically determined. The system employed in this study requires that the user draw with the mouse the ROI. A binary mask of the ROI is then created and applied to the video after the image is enhanced using histogram equalization. Through the setting regions of importance, inevitable noise coming from unimportant regions can be ignored with a resulting improvement in overall response of the system and computing resources management.

Finally the system is capable of selecting several unconnected regions of importance on the same video sequence as can be seen in Fig. 2.

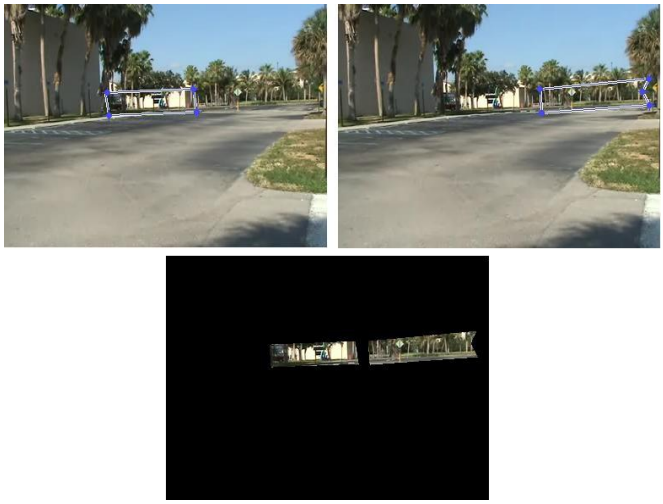

Figure 2: Region-of-importance selection process.

2.2 IMAGE ENHANCEMENT

A small object can be sensed if its contrast is large enough for the human visual system (or the computer vision system) to detect. Contrast depends on multiple factors such as color difference (not only hue difference but saturation and brightness as well), level of illumination of the scene, quality of the camera, etc. Although most of these features are out of the control of the object detection system, some improvement can be achieved through the application of image processing techniques. For our system we implement discrete histogram equalization to the image in order to enhance the contrast between foreground objects and the background.

For the system, contrast enhancement is desired because it facilitates the differentiation between the object and the background. Consider Fig. 3, which depicts a person walking in the distance and occupying just a few tens of pixels.

Figure 3: Low resolution Object (Left), contrast enhanced image (Right).

The shirt of the person has less contrast than the pants as can be appreciated in the color and grayscale versions of the image. Note how after the contrast enhancement, the object and the background tend to be mostly black and mostly white which makes the object easier to recognize. The result is that more information can be extracted as more pixels from the objects are detected as foreground. At the same time, there is more noise in the system due to the contrast enhancement, which enhances changes in the scene as well as the object-background contrast.

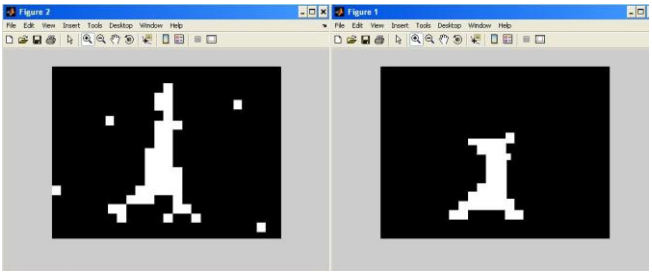

Figure 4: Background subtraction of image enhanced object (Left), and non enhanced object (Right).

In fig. 4, the image on the left has more information but the system has more overall noise, while the image on the .right has better noise handling but some information is lost in the process due to poor contrast in some regions.

2.3 BACKGROUND SUBTRACTION

The program implements one of the three background subtraction algorithms available. The three algorithms were selected because they are quite different in their approach.

The frame difference is perhaps the simplest and fastest background subtraction method available. In this method, each frame is subtracted from the previous frame, and the difference is then compared with a threshold. If the difference is bigger than the threshold then the pixel is foreground, otherwise it is background. This approach has an important advantage in the fact that a constantly changing background makes this algorithm a fast adapting one. It adapts quickly to changes in illumination and shadows as well as to changes in the weather conditions of the video. On the other hand, the system is very susceptible to noise, and all the objects must be moving constantly because the moment they stop they will be identified as background in subsequent frames. Furthermore, the inside of the objects would be recognized as background if the objects are big enough with little internal structure.

Approximate Median is of middle complexity, being as easy to optimize as the frame difference method but with added robustness and less susceptibility to noise. In this method each pixel in the current frame is compared with the one in the background. If the pixel in the current frame is larger, then the intensity of the pixel in the background is incremented by one. If on the other hand the background pixel is larger, then it is decreased by one. The background will then tend to be a good approximation of the median with the time of stabilization being a function of the number, the size, and the velocity of the objects moving. This method will have less memory usage at the expense of some stabilization time

Mixture of Gaussians is complex and elegant but takes a significant amount of time and computer resources, and its optimization is more difficult because it is multivariate. This technique takes into account changing elements in the background such as moving trees or falling snow. In order to create the model of the background, a combination of different Gaussian PDF's is required to model each pixel. In MoG, the background is not modeled as a frame of values. Instead, the model is purely parametric with each pixel location represented by a number (mixture) of Gaussian functions that sum together to form a probability distribution function of the form:

$$F(i_t = \mu) = \sum_{i=1}^{k} \omega_{i,t}.\eta(\mu, \sigma) \qquad (1)$$

The parameter μ corresponds to the mean of each Gaussian component and can be thought of as an educated guess of the pixel value in the next frame assuming that pixels are usually background. The parameter ω, which is the weight, and σ, which is the standard deviation of each component, can be thought of as measures of our

confidence in that guess (higher weight and lower standard deviation equals higher confidence). Because of memory limitations, the program works with only three Gaussian components per pixel.

The three algorithms have as output a binary image for each frame of the video, with zero representing the background and one representing the foreground. The images still contain some noise due to the different conditions of the video sequence.

Figure 5: Grayscale image (bottom), background subtraction algorithm (top-left), and morphological filter output (top-right).

2.4 MORPHOLOGICAL FILTERING

The morphological filtering operation is intended to reduce the noise as much as possible in background subtraction systems, but in the case of low-resolution objects special care has to be taken. Due to the nature of the object (objects comprising just a few pixels), a morphological operation could easily remove important information (even remove the object entirely) or allow noise to pass. The morphological filters were chosen to reduce spatially small noise that is present across the video sequence. The noise comparable to or bigger in size than the object is handled partially in the selection of the Region of Interest and partially by the buffering system in the tracking algorithm. An example of how the morphological filtering removes the noise of the system can be seen in Fig. 5.

2.5 TRACKING SYSTEM

The tracking system implemented is a Mealy finite state machine (FSM) with three definite states: a buffer state, an active state, and an inactive state. The diagram is shown in Fig. 6.

Buffer State: When a new object is detected, the buffer state keeps track of the object in the first three frames; this is done to avoid the appearance of ghost objects. The buffer state saves system resources by allowing the FSM to keep track only of persistent objects in the video. When the object has been in the buffer state for three consecutive frames, its information is compared with that of the Inactive State to check if the new object is in fact an old object that previously disappeared due to an occlusion. If no object in the inactive list is comparable to the new object, then the object is labeled as a new object and its information is transferred to the Active State. Further development in this algorithm will label places where a new object can appear so that no new objects can appear in an unrealistic way in the middle of the scene.

Active State: The active state keeps track of the objects while they are present in the video and after they have passed the buffer state. The active state keeps track of the centroid position, past centroid positions, and the index for each of the pixels that compose the object. If an active object disappears in the middle of the video, the ID of the object is stored in the Inactive State and the Active State stops tracking it until the buffer state finds a match

between a new object that appeared in the middle of the video and the stored inactive object. When that happens, the buffer state transfers the information to the Active State and the tracking is resumed.

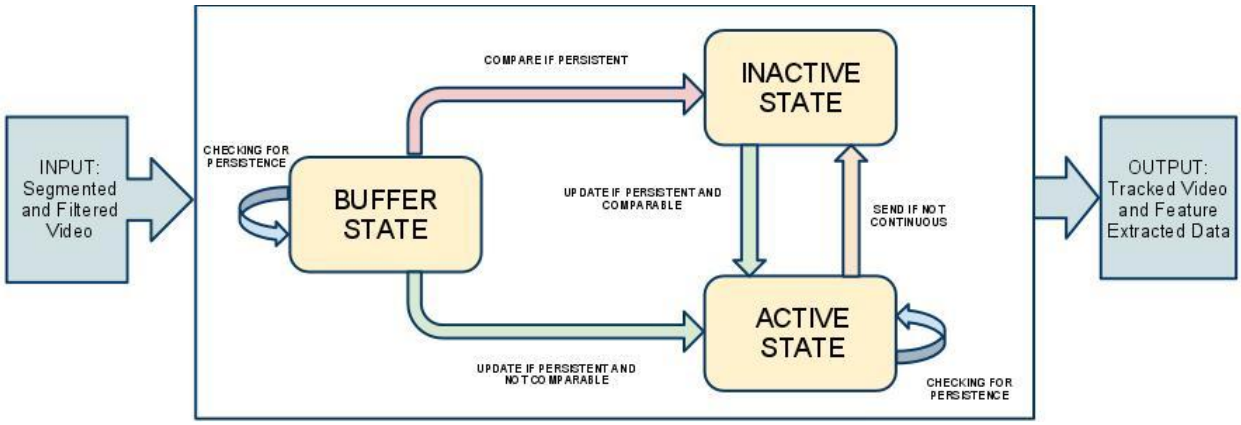

Figure 6: Tracking system state diagram.

Inactive State: The inactive state is the only state of the system where information about the physical properties of the object is not stored or generated. Instead, it keeps a list of IDs or pointers of the objects that were being tracked by the Active State and that disappeared in the middle of the video, probably because of an occlusion. Occlusions in the middle of the video can be due to static background objects that can be marked beforehand, such as trees or walls, or due to moving foreground objects, such as other persons moving. When an object is ready to go out of the buffer state, the inactive state sends the ID to the buffer state where a comparison is made to check whether or not the new object is in fact an inactive object reappearing

To determine if an object in the current frame is the updated version of an object being tracked, the first step is to create an extended bounding box around the object being tracked and check for centroids of objects inside this region in the current frame as seen in Fig. 7. The bounding box is extended to compensate for speed of the moving objects four pixels in each direction. If there is only one object in that region, then it is considered a match and the information for that object is updated accordingly. If, on the other hand, there are more than one object inside the region, the system compares the object sizes of the candidates with that of the previous frame to decide which one is a match. Lastly, if there are no matches, the object is either discarded or transferred to the inactive state if it has been a persistent object. An object is considered persistent when it has been in the active list for at least three frames.

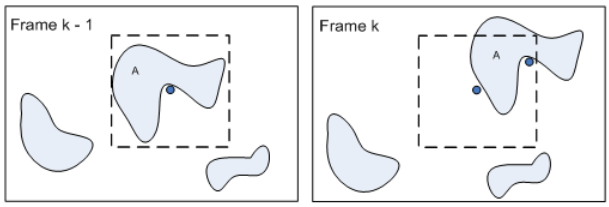

Figure 7: Centroid tracking systems across frames.

2.6 DATA CELL STORING AND VIDEO PRESENTATION

A cell is a matrix where each of its elements is of a different nature (e.g., one of the elements is a vector, another one is a matrix, another is a string of characters, etc.). The cell generated by the program stores information from new objects such as the frame in which it appeared, the history of the position of the centroids, the list of pixels of the object, the bounding box information, and the instantaneous velocity.

The video presentation generates an output video that is like the original video but with the objects detected being circled and the trajectory of the centroids of the objects highlighted. The video presentation does not present exact data but it gives a good idea of how the system is behaving. It is also an early alarm system telling the user where

the activity is in the video so that the user can understand the data from the cell. In Fig. 8, the video presentation shows how the system handles occlusions.

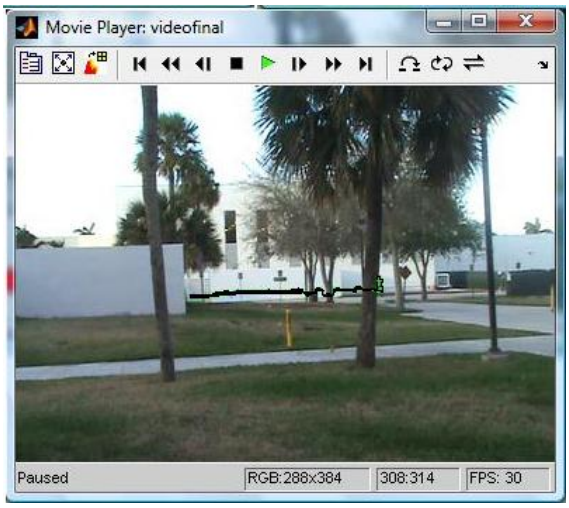

Figure 8: Tracking system showing occlusion. (from video 1)

2.7 MOTION ANALYSIS

There is behavioral evidence that animals and humans can recognize biological motion because it contains periodic characteristics. This has been confirmed by different studies; the most relevant being the one by Johansson (Johansson, 1973), in which moving light displays where attached to subjects' joints in a dark environment, showing that by monitoring only those few points over time it was possible to analyze human and animal motion.

The algorithm implemented in the program computes an object's self-similarity as it evolves in time. For periodic motion, the self-similarity measure is also periodic, and time-frequency analysis is applied to detect and characterize the periodic motion. The periodicity is analyzed using the 2D lattice structures inherent in similarity matrices. From this approach, the system extracts the isolated image of an object across N consecutive frames. Once the information is accumulated, the images are resized according to the median values. Then the system calculates its correlation matrix, according to the following equation:

$$St_1, t_2 = \sum_{(x,y)\in Bt_1} |Ot_1(x,y) - Ot_2(x,y)|, \qquad (2)$$

where *B* is the bounding box of object *O*, and t_i makes reference to the different resized instances of the object ($0 \leq i \leq N$). Figure 9 shows the graphical representation of an object's correlation matrix. In the figure, the main diagonal is the correlation between the frames with themselves (e.g., frame 1 against frame 1 of the video) so the correlation is total and appears as a black line (for this figure, total correlation is represented by black while no correlation is represented by white). If there appear lines parallel to the diagonal with some important correlation values that means the object is moving with periodic motions. Rigid objects do not exhibit this behavior.

The next step is the computation of the discrete Fourier transform (DFT) of the correlation matrix. The object's motion will be considered as periodic if there are values that meet the condition below:

$$P > \mu_p + K\sigma_p \qquad (3)$$

Where *P* is the maximum of the magnitude of the DFT of the correlation matrix, and μ_P and σ_P are the mean and standard deviation of *all values of the DFT with K a* constant defining the periodic threshold.

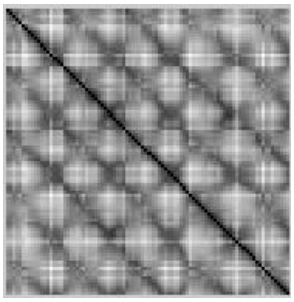

Figure 9: Example of the correlation matrix of a tracked object.

3. EXPERIMENTS AND RESULTS

Three videos were fully analyzed, the first one with a person walking through multiple occlusions (video 1), the same scene with the person running instead of walking (video 2), and a different scenario with two persons present and one occlusion (video 3). The scenes present manmade features such as walls and sidewalks and natural features such as sunlight and trees. Typical object sizes were 18 pixels of height for videos 1 and 2 and 8 pixels height for video 3.

3.1 BACKGOUND SUBTRACTION

One of the traditional methods for comparing background subtraction algorithms is the use of the ground-truth comparison (Abdou and Pratt, 1979). In such a scheme, background subtraction algorithms are compared with images annotated by hand as in Fig. 10 and the result is analyzed using detection theory techniques.

In high-resolution images, ground Truth analysis is useful because the limits between objects and background are well defined. In low-resolution images, however, the object is blended with the background in such a way that for some regions it is not clear if a given pixel is a part of the object or the background. This effect is evident in Fig.11 which is an object from video 3.

Figure 10: Ground Truth (manually annotated).

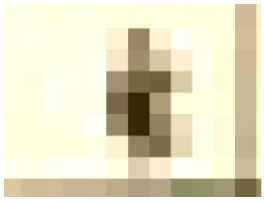

Figure 11: Very low resolution image from video 3.

Rather than through the ground-truth analysis, the algorithms were compared when applied to the same set of videos. Prior to the comparison, a tuning process was applied to the Frame Difference and the Approximate

Median methods. The Mixture of Gaussian Method is a multivariate parametrical method and thus its tuning is rather complex. In this case, values were moved around the ideal set.

For the frame difference method and using a grayscale of 8 bits, a value of 70 in the difference threshold showed the best balance between noise level and preservation of the integrity of the object (the value was changed from 30 to 120).

A similar test was performed with the approximate median method with an optimal value of 40 for the threshold, proving it has a better performance against the noise than the frame difference. A sample of the test realized is shown in Figure 12.

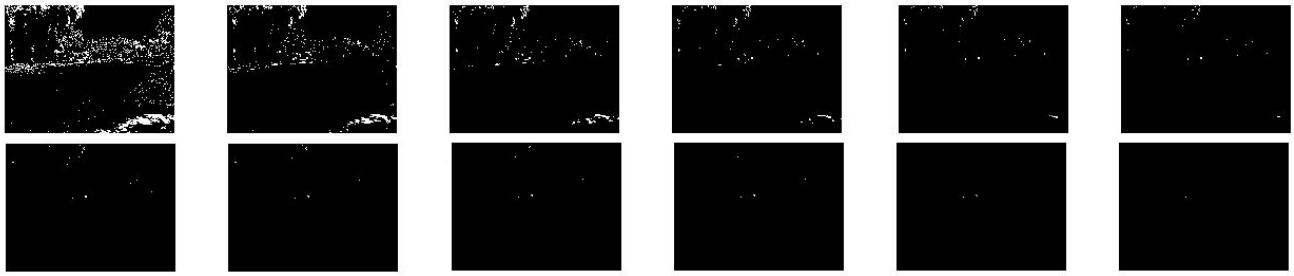

Figure 12: Comparison of the background subtraction algorithm for different threshold (frame difference).

Because the Mixture of Gaussians algorithm has many parameters, the tuning problem was approached one parameter at a time, sweeping the algorithm for a particular parameter and then sweeping another parameter with the previous one fixed at the best response. This approach is far from ideal, since it does not take into account possible interaction between parameters.

The optimal threshold values found were stored as default values but the user can change them if desired.

3.2 TRACKING SYSTEM

Each of the videos that were tested featured different scenarios. The videos were prerecorded as the system in this first stage is not yet intended to work in real time. The system was able to detect and track even the person that barely moved as can be seen in Fig. 13.

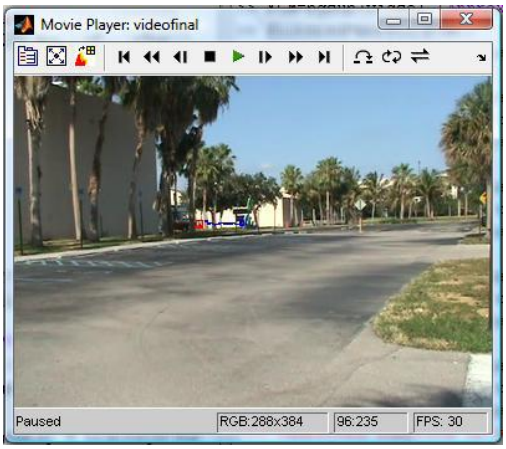

Figure 13:Video 3 with two tracked objects

The cell stores the data obtained from the analysis. The video provides the visual aid but with the real information lies in the cell.

3.3 MOTION ANALYSIS

The three videos were tested for periodicity using the self-similarity matrixes in the occlusion-free regions showing that even in the very low resolution regime there is still detectable periodic biological motion. The self-similarity test of a boat is shown in Fig. 14 for comparison.

Figure 14: Self Similarity tests for videos 1 and 3 (left, center), self similarity test of a boat showing not periodic behavior (Right)

4. CONCLUSION

The proposed system is capable of detecting, tracking and extracting information from objects as small as 8 pixel of height showing that with a combination of well known image processing techniques and carefully selected filtering it is possible to perform motion analysis of low resolution objects present in a given video sequence.

REFERENCES

I. Abdou and W. Pratt, "Quantitative design and evaluation of enhancement/thresholding edge detectors," Proceedings of the IEEE, vol. 67, no. 5, pp. 753–763, 1979.

R. Cutler and L. S. Davis, "Robust real-time periodic motion detection, analysis, and applications," IEEE Trans. Pattern Anal. Machine Intell., vol. 22, Aug. 2000, pp. 781–796.

S. Eddins, R. C. Gonzalez and R. E. Woods: "Digital Image Processing Using Matlab", Prentice Hall, Second edition (2004).

R. C. Gonzalez and R. E. Woods: "Digital Image Processing", Prentice Hall, Second edition (2002).

W.E.L. Grimson and C. Stauffer, "Adaptive background mixture models for real-time tracking," Proc. IEEE CVPR 1999, June 1999, pp. 24&252.

W. Hu; T. Tan; L. Wang; S. Maybank, "A Survey on Visual Surveillance of Object Motion and Behaviors," IEEE Trans. Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 34, no. 3, 2004, pp. 334–352..

G. Johansson, "Visual Perception of Biological Motion and a Model for its Analysis," Perception and Psychophysics, vol. 14, 1973, pp. 210-2 11.

N. J. B. McFarlane1 and C. P. Schofield, "Segmentation and tracking of piglets in images", Machine Vision and Applications, Vol 8-3 pp. 187-193.

D. Pava and W. T. Rhodes, "Removing Ambiguity in 2-D Image Information by Means of 3-D Models," in *Digital Holography and Three-Dimensional Imaging*, OSA Technical Digest (CD) (Optical Society of America, 2008), paper DMA3.

M. Piccardi, "Background subtraction techniques: a review," 2004 IEEE International Conference on Systems, . Man and Cybernetics, vol.4, Oct. 2004, pp. 3099-3104.

Authorization and Disclaimer

Authors authorize LACCEI to publish the paper in the conference proceedings. Neither LACCEI nor the editors are responsible either for the content or for the implications of what is expressed in the paper.