Moving Object Detection and Extraction for Video Editing

Mayur Ghogale, Prasad Marla, Nandan Daga, Ankush Kawanpure

Abstract— A novel approach for extraction of foreground object and using the extracted object in applications like object cloning in a video, insertion of object in an another video, increasing speed of objects in a video. Gaussian mixture model is used for foreground extraction as it deals effectively with lighting changes, repetitive motions from clutter etc. Further a fully automated way of extracting the object from video along with its relative motion is developed. Such extracted object can be inserted in videos of same scene or different scenes to produce pleasant visual effects.

Index Terms— Moving Object Detection, Moving Object Insertion, Object Extraction, Object Cloning, Video Editing, Gaussian Mixture Model.

1 Introduction

Editing video for producing various special effects is a critical task in many computer graphics applications. Consider a scene in a movie which requires object insertion in a particular frame. Fast and efficient algorithms for moving object removal and insertion are proposed in this paper which could be used for the above problem. C. Kim et al. [5] discuss a segmentation approach for moving objects and extracting video object planes for these objects. It uses a double-edge map which is constructed from difference between two successive frames. When all the edge points which belong to previous frame are removed remaining edge map is used to extract video object plane. Limitations of this approach are that it works only for mpeg videos. Various change based detection and extraction approaches are discussed [6], [7], [8], [9] it involves discriminating background and moving objects by means of the higher-order statistics (HOS) performed on the inter frame differences of DC image. These approaches are restricted because they are specific to particular scenes and video types. A video type independent object extraction algorithm is proposed which facilitates automatic object extraction.

The paper is organized as follows. In next section different techniques for foreground detection are discussed. In section 3 and 4 we discuss algorithms for object extraction and insertion. In section 5 and 6 we discuss results and applications, followed by conclusion and references respectively.

2 Foreground Detection

Identifying mobile objects from a video sequence is a fundamental task in many computer graphics applications. A common approach is to perform background subtraction, which identifies moving objects from the portion of a video frame that differs significantly from a background model. This approach is also known as foreground detection. Any efficient background subtraction algorithm must be 1) robust against changes in illumination. 2) It should avoid detecting non-stationary background objects and shadows cast by moving objects. There are many algorithms discussed so far and we would like to list few before narrowing onto approach we chose for foreground detection.

2.1 Frame Differencing

Frame differencing makes use of the pixel-wise differences between consecutive frames in an image sequence to extract moving regions or moving object. Foreground pixels are determined by:

$$B(x,y,t)=I(x,y,t-1) \qquad (1)$$
$$|\ I(x,y,t) - B(x,y,t)\ | > th \qquad (2)$$

I(x,y,t) = Current image at time t.
B(x,y,t) = Background image
th = threshold.

Accordingly, if condition in Eq. (2) is satisfied then the pixel is foreground else background.

2.2 Mean Method

This approach works similarly to frame differencing, but the background image is modelled as the mean of n frames given by the formula,

$$B(x,y,t) = 1/n \sum_{i=0}^{n-1} I(x,y,t-1) \qquad (3)$$
$$|\ I(x,y,t)- B(x,y,t)\ | > t_h \qquad (4)$$

Accordingly, if condition in Eq. (4) is satisfied then the pixel is foreground else background.

2.3 Median Method

This approach works similarly to frame differencing, but the background image is the median of n frames given by the formula,

$$B(x,y,t) = Median(I(x,y,t-1)) \qquad (5)$$
$$|\ I(x,y,t) - B(x,y,t)\ | > t_h \qquad (6)$$

Accordingly, if condition in Eq. (6) is satisfied then the pixel is from the foreground otherwise it is background pixel.

2.4 Feature Based Recognition

Feature based recognition calculates a number of properties of the input image and combines them into a feature vector. An object model is a set of feature vectors associated with a set of representative images of the object. A target image is classified by computing the feature vector of the image and comparing it directly to the model vectors. An image is identified as an instance of an object when that object model contains the feature vector that is closest to the image feature vector.

2.5 Viola-Jones Object Detection

The basic steps involved in this algorithm are:

1) Haar Features Selection: Common human facial properties e.g. nose bridge region is brighter than the eyes.
2) Creating Integral Image: Data structure used for generating the sum of values in a rectangular subset of a grid.
3) Adaboost Training algorithm: Learning algorithm
4) Cascaded Classifiers: Series of classifiers to maximize the accuracy of output.

2.6 Gaussian mixture model

Single Gaussian per pixel approach fails when dealing with lighting changes, repetitive motions of scene elements, tracking through cluttered regions, slow-moving objects, and introducing or removing objects from the scene. To avoid drawbacks of single Gaussian, Proposed algorithm uses background model suggested by Stauffer et al. [1] where each pixel is modelled as a mixture of Gaussian and an online approximation to update the model. Here, the value of a particular pixel is modelled as a mixture of Gaussians rather than modelling it as particular single Gaussian. Depending on the persistence and the variance of each Gaussian of the mixture, it is determined which Gaussian may correspond to background colors. Pixel values that do not fit the background distributions are considered foreground. The online update process of this method is described below.

Consider a pixel {x0,y0} at any time t with its history X1,X2,..Xt where Xi {RGB value at {x0,y0} at time i: 1≤ i ≤ t}. A mixture of K Gaussian distribution is used to model the recent history of each pixel, { X1,X2,..Xt}. The probability of observing the current pixel value is :

$$P(X_t) = \sum_{i=1}^{n} \omega_{i,t} * (\eta\, X_t \mu_{i,t} \Sigma_{i,t}) \tag{7}$$

Where,
K – number of gaussians usually 3-5
$\omega_{i,t}$ – weight of the ith gaussian at time t.
$\mu_{i,t}$ – mean of ith gaussian in mixture at time t.
$\Sigma_{i,t}$ – covariance matrix ith gaussian in mixture at time t.
□ - gaussian probability density function.

$$(\eta X_t \mu \Sigma) = (1/2\Pi\Sigma^{0.5}) * exp^{(X_t-\mu_t)^T \Sigma^{-1} (X_t-\mu_t)} \tag{8}$$

The co-variance matrix is a 3X3 matrix and is a diagonal matrix because R, G, B values are assumed to be independent.

$$\Sigma_{k,t=}\sigma^2 i \tag{9}$$

The major components of the mixture model will be represented by one of the new pixel value and is used to update the model. A match is defined as a pixel value within 2.5 standard deviations of a distribution. If none of the K distributions match the current pixel value, the least probable distribution is replaced with a distribution with the current value as its mean value, an initially high variance, and low prior weight. The prior weights of the K distributions at time t, $\omega_{k,t}$, are adjusted as follow

$$\omega_{k,t} = (1 - \alpha)\omega_{k,t-1} + \alpha M_{k,t} \tag{10}$$

α is the lerning rate and $M_{k,t}$ is 1 for match otherwise 0.
The μ and σ paramters for unmatched distributions remain same .For matched paramters

$$\mu_t = (1 - \rho)\mu_{,t-1} + \rho X_t$$
$$\sigma_t^2 = (1-\rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^T (X_t - \mu_t) \tag{11}$$

where learrning factor $\rho = \alpha\eta(X_t|\mu_t\sigma_K)$

3 Object Extraction

Once targeted object from video is detected, it has to be extracted. Object extraction is facilitated by the segmentation information. The part of the video frame which is segmented is the object, that contour when matched with the original video frame will yield the actual extracted object. An intrinsic noise is often introduced during the segmentation process and such noisy parts of the frame will show up in the segmentation results.

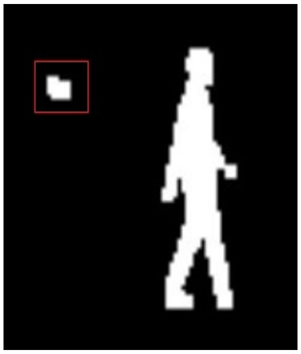


Fig.1(a) Segmented Object

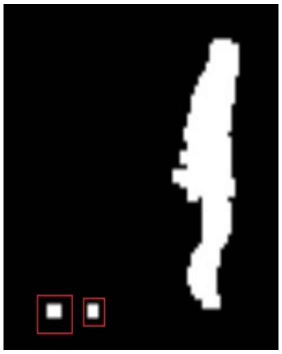


Fig.1(b) Segmented Object

Fig 1(a) and Fig 1(b) are segmented video frames taken from input video. Segmented parts that are bounded by boxes are examples of noise introduced during the detection. When extracting the object from video sequence such noisome parts are to be avoided as they are not the part of intended object. For selecting the object patch for extraction only select the patches that can be reached from the biggest patch in the segmentation result. Let T be the patch with maximum area (representing the intended object) and let U and V be the patched introduced by noise. U and V are isolated patches. As U and V cannot be reached from T these patches are discarded. As the bounding box will always contain the tracked object, the patch with maximum area will always be that of the tracked object. Thus T will be the resultant object for extraction.

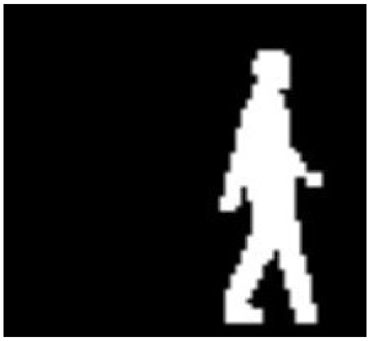
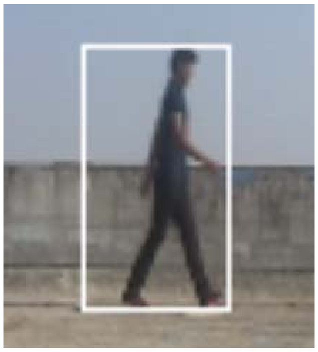


Fig 2: Segmented Object after smoothing out noise

After contour T has been selected, each pixel location from this contour is to be matched with pixel from actual video frame. Data represented by the pixels location from original video will give the actual object.



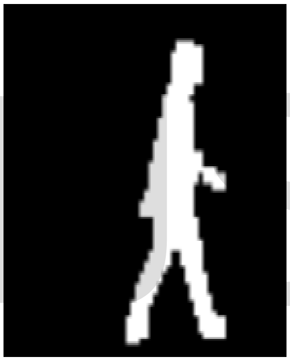


Fig. 3(a) Actual Mapping *Fig. 3(b) Segemented Result*

After object has been detected and extracted, its motion has to be stored. How object's location changes between frames denotes the motion of the object. Let F0 be the first frame in the video where object is detected and (x0,y0) and (x1,y1) be the left top and right bottom coordinates of the bounding box in frame F0. The relative change in these coordinates in the immediate next frame F1 in which object is detected will represent the motion of object across the frame. It can be inferred that relative change from F0 to F1 i.e. Difference between F1 and F0 is inter frame motion. A matrix representing the inter frame motion; called Motion Matrix M indicates motion of object in the entire video. This enables to maintain the object's original motion when object is inserted in some other video.

In Fig (4) two frames of a video are taken. The change in the relative position of the bounding box (white bounding box in frames) will denote the relative motion of the object across frame. Motion matrix will contain this relative motion. Depending upon the movement of object change in position can be called as xDiff (difference along x-direction) and yDiff (difference along y-direction).

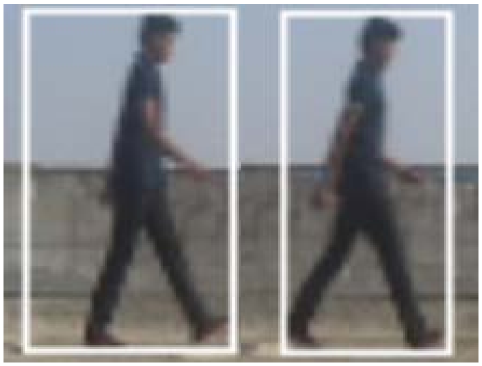


Fig. 4 Relative motion calculation across frames

Thus the algorithm for video object detection and extraction can be summarized below as

```
1. Get input video.
2. Set up a foreground detector with n number of Gaussians.
3. Select k as a no of running frames.
4. Store video details.
5. While (all frames are over)
        5.1 Read next frame.
        5.2 Extract foreground of the frame.
        5.3 Remove redundant noise.
        5.4 Put a bounding box around detected foreground.
        5.5 If (one object detected)
                5.5.1 Fill the motion matrix of object
                5.5.2 for (all the pixels)
                5.5.3 If pixel is part of object
                5.5.4 Save the object data
        5.6 If (more than one object is detected)
                5.6.1 Save individual motion.
                5.6.2 Repeat steps 5.1 for each object
        5.7 Save the extracted object and its motion matrix
```

4 Object Insertion

Before inserting extracted object in other scenes, different parameters have to be taken in consideration. Each video is sequence of video frames and each video frame is characterized by its properties like brightness, contrast. Thus it is necessary to consider the difference of these properties between source object video and target video. As visual quality is governed by changes in brightness and contrasts a more seamless integration of extracted object in target video demands that these properties should be uniform across both the videos. Brightness is the mean of all the pixels from the data. Brightness of extracted object is increased or decreased depending upon the brightness of the target video. Similarly contrast of the target video frame is calculated and accordingly extracted object's contrast is adjusted such that source video and target video exhibit uniform contrast. Depending upon subjective quality requirement different factors like hue, saturation should de made uniform. Thus the algorithm for moving object insertion can be summarized as below

Algorithm for Object Insertion

```
1. Accept destination video.
2. Accept startX, startY for insertion.
3. Load motion matrix.
4. Initialize inter-frame differences xDiff and yDiff.
5. While (all frames are over)
        5.1 Read destination frame.
        5.2 Extract foreground of the frame.
        5.3 Retrieve extracted object.
        5.4 Insert object between
        Y = destY to Y = destY + motion_mat[i].Height
        X = destX to X = destX + motion_mat[i].Width
        5.5 Update the xDiff and yDiff accordingly
        5.6 Readjust the startX and startY values
        5.7 Save the final results.
6. Stop
```

Where,

startX – starting co-ordinate for x axis co-ordinates
startY – starting co-ordinates for y axis co-ordinates
xDiff – inter frame difference between x axis motion of object given by Motion Matrix
yDiff – inter frame difference between y axis motion of object given by Motion Matrix.

5 Results

5.1 Cloning of same object in the video

Fig. 5(a) shows a video frame of a person walking by the wall. From this input video the walking person is tracked and detected. After applying extraction technique on the object it is extracted with its motion matrix. The same object is then inserted in the same video. There is only spatial difference between these two objects. The inserted object is inserted some distance ahead of the source object. This gives a pleasant feeling of cloning the video objects. The similarity in movement of both the objects is striking. The insertion is seamless and indistinguishable.

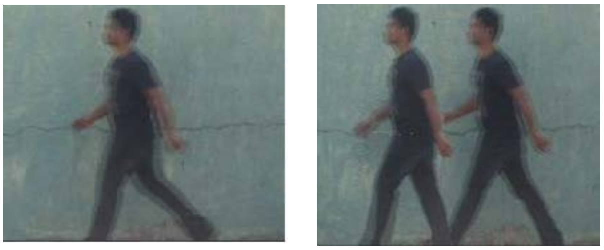


Fig. 5 (a) Input *Fig. 5 (b) Cloned Result*

5.2 Increasing Speed of Object

During the extraction of object using extraction technique, motion matrix is calculated. This matrix contains relative motion of the object across the video frames. Using this motion information, the motion of the object when it is inserted into target video can be controlled. In this result source video contains a person walking by wall. After inserting the object into target video its speed is increased by considering relative motion from motion matrix. Since motion matrix contains relative inter frame positioning of object from source video, this fact can be exploited to adjust the speed of the object. Similarly it is possible to slow down the inserted object by decreasing its speed.

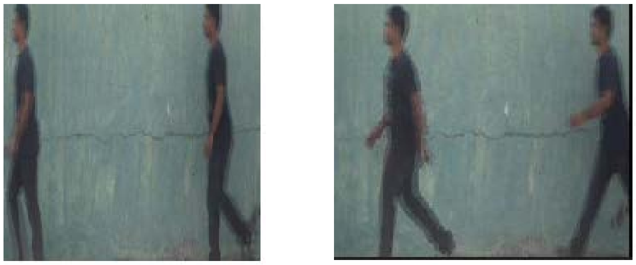


Fig. 6 (a) Input *Fig. 6 (b) Increased Speed*

5.3 Insertion of Object in Empty scene

In this video a walking person is tracked and extracted. This object is inserted into an empty scene from the same background. The insertion of newly extracted object is achieved in correct manner evident by similarity between both the videos. It is also possible to reverse the direction of the object by rotating it by 180 degrees.

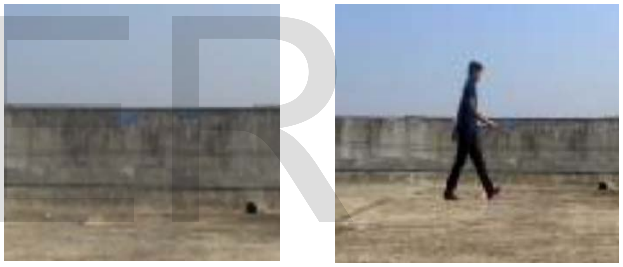


Fig. 7 (a) Input *Fig. 7 (b) Insertion Result*

6 Applications

Object detection and tracking technique explained in this paper can be used for traffic surveillance, unattended object detections in public places. Object extraction can be used in various special effects creation during movie post-production. The results explained in this video can be extended to create cloning effects to generate multiple moving objects from one object. It is possible to increase the speed of moving entities to create more sci-fi movie like effects.

7 Conclusion

Using an extension of Gaussian Mixture model it is possible to track objects in video with stationary cameras. Such tracked objects can be segmented and extracted using the motion based extraction technique. Motion based extraction technique facilitates changing the motion parameters of the extracted object. Adjusting the parameters such as brightness and contrast allows seamless insertion of extracted object in the video. Gaussian based tracking technique does not generate proper object detection and tracking for videos shot with moving camera. As foreground subtraction algorithm fails to generate feasible results in that case. While inserting extracted object in target videos that don’t exhibit any similarity with original source video scene, insertion does not produce visually pleasant result. Upon close observation the difference between the two scenes is evident. This happens because during detection phase some part of the original scene is detected as a part of the object and thus when such object is extracted and inserted, it is possible to detect the object as a foreign entity. Such discrepancy needs manual interaction from user’s part to remove unwanted parts.

In future, work is proposed to improve the detection technique to make it more accurate so that no part from surrounding scene is detected as a part of the object. Also it should be able to handle videos that are shot with moving cameras. Thus this will allow to develop a fully automated tool that can detect, extract and insert moving objects from given video to target video.

REFERENCES

[1] Stauffer C, Grimson W. Adaptive background mixture models for real-time tracking. Proc IEEE Conf on Comp Vision and Patt Recog (CVPR 1999) 1999; 246-252 J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Nir Friedman and Stuart Russell. “Image segmentation in video sequences: A probabilistic approach,” In Proc. of the Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI), Aug. 1-3, 1997.

[3] B. K. P. Horn. Robot Vision, pp. 66-69, 299-333. The MIT Press, 1986R.

[4] Background subtraction techniques: a review. In IEEE International Conference on Systems, Man and Cybernetics, 2004, volume 4, pages 3099-3104, 2005. 740–741.

[5] C. Kim and Hwang, "Fast and Automatic Video Object Segmentation and Tracking for Content-Based Applications" in IEEE Transactions On Circuits And Systems For Video Technology, Vol. 12, No. 2, February 2002, pp 122-129.

[6] T. Aach, A. Kaup, and R. Mester, “Statistical model-based change detection in moving video,” Signal Processing, vol. 31, pp. 165–180, Mar 1993.

[7] R. Mech and M. Wollborn, “A noise robust method for 2D shape estimation of moving objects in video sequences considering a moving camera,” Signal Processing, vol. 66, pp. 203–217, Apr. 1998.

[8] A. Neri, S. Colonnese, G. Russo, and P. Talone, “Automatic moving object and background separation,” Signal Processing, vol. 66, no. 2, pp. 219–232, 1998.

[9] J. Guo, J. W Kim, and C.-C. J. Kuo, “Fast and accurate moving object extraction technique for MPEG-4 object-based video coding,” SPIE, vol. 3653, pp. 1210–1221, Jan. 1999.

[10] T. Sikora, “The MPEG-4 video standard verification model,” IEEE Trans. Circuits Syst. Video Technol., vol. 7, pp. 19–31, Feb. 1997.