International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161
©2014 INPRESSCO®, All Rights Reserved
Available at http://inpressco.com/category/ijcet

Research Article

Video Object Detection using Variable Threshold

Kumudini Borkute[A*] and Ashwini Shende[A]

[A]Department of Electronics Engineering,Rajiv Gandhi College of Engineering, nagpur

Accepted 10 May 2014, Available online 01 June2014, **Vol.4, No.3 (June 2014)**

Abstract

Recognition-by-components is a theory of object recognition that accounts for the successful identification of objects despite changes in the size or orientation of the image. RBC explains how moderately degraded images, as well as novel examples of objects, are successfully recognized by the visual system This paper describes a general method for building cascade classifiers from part-based deformable models such as pictorial structures. This paper focuses primarily on the case of star-structured models and show how a simple algorithm based on partial hypothesis pruning can speed up object detection by more than one order of magnitude without sacrificing detection accuracy. It based on two algorithms; the cascade variant dynamic programming algorithm fills values in DP tables and training algorithm for the thresholds used in the cascade.

Keywords: *Cascade, object detection, star model, deformable part model, thresholding*

1. Introduction

Videos are actually sequences of images, each of which called a frame, displayed in fast enough frequency so that human eyes can percept the continuity of its content. It is obvious that all image processing techniques can be applied to individual frames. Besides, the contents of two consecutive frames are usually closely related. Visual content can be modeled as a hierarchy of abstractions. At the first level are the raw pixels with color or brightness information. Further processing yields features such as edges, corners, lines, curves, and color regions. A higher abstraction layer may combine and interpret these features as objects and their attributes.

Object detection from a video stream is one of the essential tasks in video processing, understanding, and object-based video encoding (e.g., MPEG4).An commonly used approach to extract foreground objects from the image sequence is through background suppression, or background subtraction and its variants (M. Everingham, L. Van Gool, C. K. I.Williams, J.Winn, and A. Zisserman *et al*,2009) when the video is grabbed from a stationary camera. These techniques have been widely used in real-time video processing. However, the task becomes difficult when the background contains shadows and moving objects, e.g., wavering tree branches and moving escalators, and undergoes various changes, such as illumination changes and moved objects.

Video object detection is the process of locating a moving object (or multiple objects) over time using a camera. It has a variety of uses, some of which are human-computer interaction, security and surveillance, video communication and compression, augmented reality, traffic control, medical imaging and video editing tracking can be a time consuming process due to the amount of data that is contained in video. dding further to the complexity is the possible need to use object recognition techniques for tracking.

Many methods have been proposed for real-time foreground object detection from video sequences. However, most of them were developed under the assumption that the background consists of stationary objects whose colour or intensity may change gradually over time. The simplest way is to smooth the colour of a background pixel with an Infinite Impulse Response (IIR) or a Kalman filter through real-time video. A better way to tolerate the background variation in the video is to employ a Gaussian function that describes the color distribution of each pixel belonging to a stable background object

In this paper we describe an object detection system for video that represents highly variable objects using mixtures of multiscale deformable part models i.e. we describe a method for building part-based deformable models in a cascades for such as pictorial structures. We focus primarily on the case of star-structured models due to their recent strong performance on difficult benchmarks such as the PASCAL datasets (P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan *et al*,2008; P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan *et al*,2009).This algorithm leads to a detection method over 20 times faster than the standard detection algorithm, which is based on dynamic programming and generalized distance transforms, without sacrificing detection accuracy. As described in (P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan *et al*,2008; P. Felzenszwalb, R. Girshick, D. McAllester, and D.

*Corresponding author: **Kumudini Borkute**

Ramanan *et al*,2009), detection with a deformable part model can be done by considering all possible locations of a distinguished "root" part and, for each of those, finding the best configuration of the remaining parts. (N. Dalal and B. Triggs *et al*,2005)

Deformable part models such as pictorial structures provide a well-designed framework for object detection. Pictorial structures represent objects by a collection of parts arranged in a deformable configuration. Each part captures local appearance properties of an object while the deformable configuration is characterized by spring-like connections between certain pairs of parts. (H.Schneiderman *et al* ,2004; N. Dalal and B. Triggs *et al*,2005)

2. Overview of proposed Scheme

In our proposed approach we first take the video as an input we are converting this input in the form of image. i.e. considering frame. Fig.1 shows the proposed system architecture. Next, we follow the work in to build the star model of an object. In which one of the parts of the object is chosen as the landmark part, and the location of any other part is only dependent on that of the landmark. Object detection with a deformable part model can be done by considering all possible locations of a distinguished "root" part and, for each of those, finding the best configuration of the remaining parts i.e. determination of threshold for each part.

We then apply these thresholds over an input image which passes the values over a specific threshold, and then the object detection algorithm is applied to get a set of detected objects which can be filtered out to get output with output parameters such as speedup factor, PSNR, precision and recall rate. and then perform the same steps for object detection. Steps will be described in detail in the following sections.

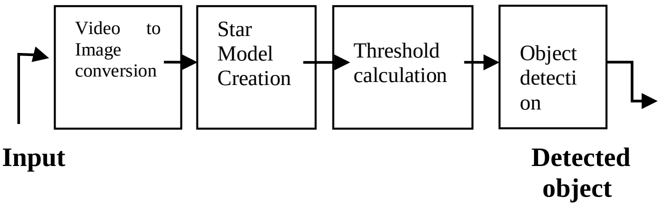


Fig.1 Overview of the proposed scheme.

3. Creation of Star Model

We follow the work in (P. Felzenszwalb and D. McAllester *et al*,2010) to represent an input image as a star model. In the star model, shown in Fig. 3, one of the parts of the object is chosen as the landmark part, and the location of any other part is only dependent on that of the landmark

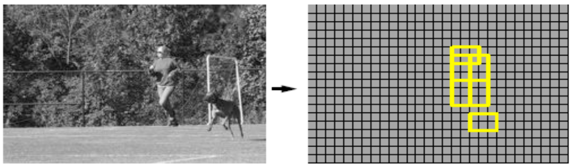


Fig 2 Example of Star model for input image.

Let M be a model with a v_0 as a root and n additional parts i. e. $v_{1,\ldots\ldots\ldots,} v_{n.}$

Let Ω be a space of locations for each part e.g. ω ε Ω could specify a position and scale. Let $m_i(\omega)$ be the score for placing v_i in location ω and it depends on the image data. let $a_i(\omega)$ specify the ideal location for v_i as a function of the root location. Let Δ be a space of displacements, and let ●: Ω X Δ→ Ω be a binary operation taking a location and a displacement to a "displaced location." Let d_i (δ) specifies a deformation cost for a displacement of v_i from its ideal location relative to the root.The score of a configuration is the sum of the scores of the parts at their locations minus deformation costs associated with each displacement and an overall score for a root location based on the maximum score of a configuration rooted at that location.

$$\text{SCORE}(\omega,\delta_{1,\ldots,}\ \delta_n) = m_0(\omega)+\sum_{i=1}^{n} m_i(a_i(\omega)\bullet\ \delta_i)-\ d_i\,(\delta_i) \qquad (1)$$

We can define an overall score for a root location based on the maximum score of a configuration rooted at that location. In a star model each part is only attached to the root, so the score can be factored as follows.

$d_i\,(\delta)$

$$\text{SCORE}(\omega) = m_0(\omega)+\sum_{i=1}^{n} \text{SCORE}_i(a_i(\omega)) \qquad (2)$$

$$\text{SCORE}_i(\eta)=\max_{\delta_i\ \varepsilon\Delta}(m_i(\eta\bullet\delta_i\ )\ )-\ d_i\,(\delta_i) \qquad (3)$$

Here $\text{SCORE}_i(\eta\ )$is the maximum, over displacements of the part from its ideal location, of the part score minus the deformation cost associated with the displacement.

4. Threshold Calculation

Here we describe an algorithm for calculation of threshold i.e. T which can be used to find the cascaded threshold and probable objects in an image. To select pruning thresholds, we introduce the notion of probably approximately admissible (PAA) thresholds have a low error with high probability. This leads to a simple method for picking safe and effective thresholds. The thresholds are safe because they have low error with high probability and lead to a fast cascade with significant pruning. After threshold calculation we describe a cascade algorithm for star models that uses a sequence of thresholds used for detection in terms of subsets of parts as

$$\text{SCORE}(\omega) \geq \text{T} \qquad (4)$$

By evaluating parts in a sequential order we can avoid evaluating the appearance model for most parts almost everywhere.

5. Object Detection

A popular approach for object detection involves reducing the problem to binary classification. However, testing all points in the search space with anon trivial classifier can be very slow. An effective method for addressing this problem involves applying a cascade of simple tests to each hypothesized object location to eliminate most of them very quickly (Everingham, L. Van Gool, C. K. I.Williams, J.Winn, and A. Zisserman *et al*,2009; P. Felzenszwalb and D. Huttenlocher *et al*,2005) Another line of research, separate from cascade classifiers, uses part-based deformable models for detection. There has been significant success in algorithmic methods for searching over these large hypothesis spaces, including methods that are "asymptotically optimal" for tree-structured models (N. Dalal and B. Triggs *et al*,2005).

6. Implementation

In this, we describe the algorithm for object detection.

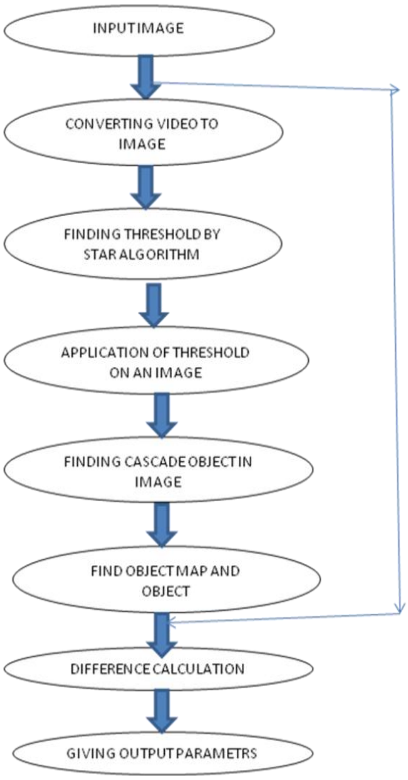

INPUTIMAGE
CONVERTING VIDEO TO IMAGE
FINDING THRESHOLD BY STAR ALGORITHM
APPLICATION OF THRESHOLD ON AN IMAGE
FINDING CASCADE OBJECT IN IMAGE
FIND OBJECT MAP AND OBJECT
DIFFERENCE CALCULATION
GIVING OUTPUT PARAMETRS

Fig.3 Flowchart

6.1 *Development environment*

In this paper we are considering two algorithms; such as star algorithm and training algorithm for the thresholds used in the cascade.

6.2 *Frame difference Calculation:*

Frame difference calculates the difference between 2 frames at every pixel position and store the absolute difference. It is used to visualize the moving objects in a sequence of frames. It takes very less memory for performing the calculation. Let us consider an example, if we take a sequence of frames, the present frame and the next frame are taken into consideration at every calculation and the frames are shifted (after calculation the next frame becomes present frame and the frame that comes in sequence becomes next frame). Figure 2 shows the frame difference between 2 frames.

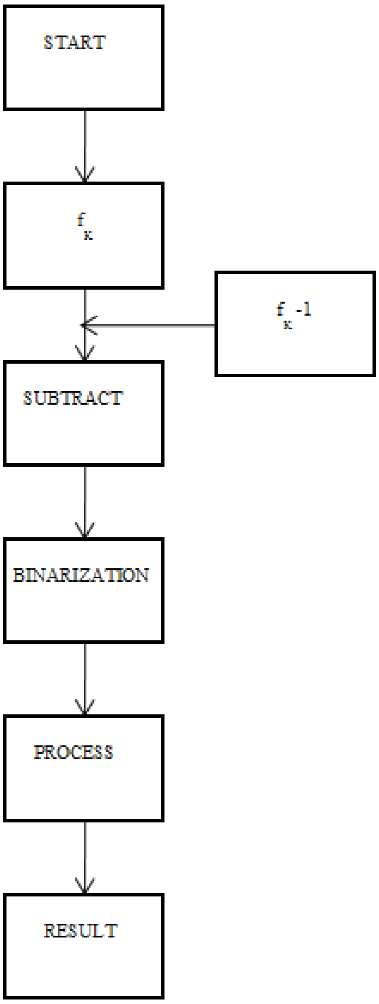


Fig.4 Flowchart of Frame Difference

6.3 *Determination of threshold*

Thresholding of images is a very important idea in image processing. It has applications in wide variety of fields such as segmentation, motion detection, blob detection etc. Thresholds are typically applied to grayscale images. Threshold selection involves choosing a gray value 't' such that all gray levels greater than 't' are mapped into the "object" label while all other gray levels are mapped into the "background" label

Steps:

(i) Find the gray levels in the image

(ii) Divide the image into NxN blocks, and then find the mean values for the block

(iii) From these mean values apply the star algorithm and then find the thresholds

(iv) Store these thresholds in a file for every image, and then apply these thresholds to an input image to get the objects in the image.

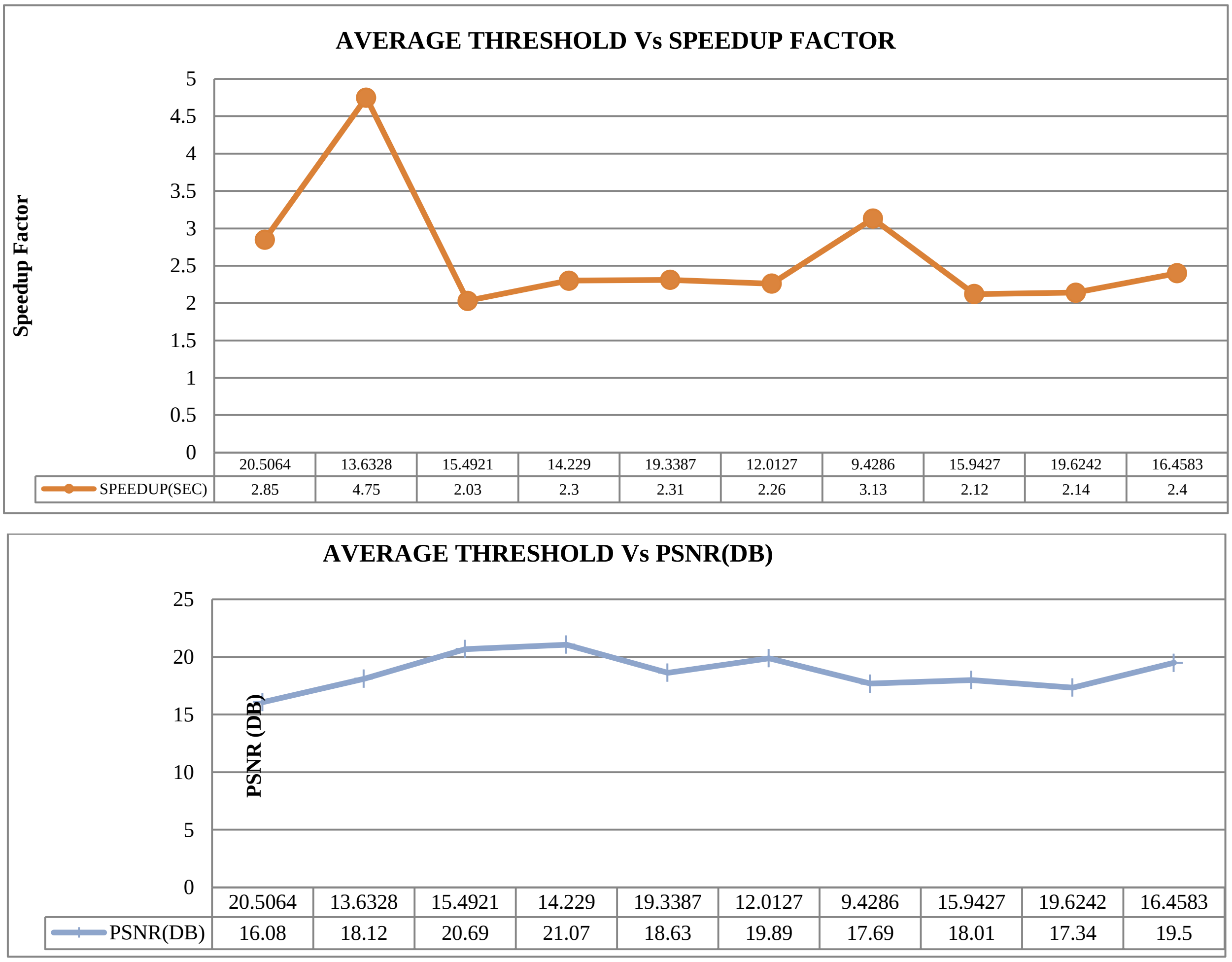


Fig 5 Graph showing output parameters: (a)Average threshold Vs Speedup factor (b)Average threshold Vs PSNR.

6.4 Application of threshold

After the calculation of threshold for each part as we are considering the deformable part model i.e. pictorial structure we have to apply threshold over an input image which gives cascaded threshold image**.**

Steps:

1) Decide starting and end point of threshold and generate threshold file,
2) Read the image file
3) Convert the image from RGB to GRAYSCALE image
4) Resize the image.
5) Apply threshold on file .Check image values i.e. between threshold or not i.e. score (w) ≥ T. a) If yes-then pass the image
   b) Else -make it zero.
   (ix) Resize the image.
6) Display Cascaded threshold image

The thresholds‘T’ automatically given an input image to classify 2 different labels in the image – an object and a background. Hence the output image can be represented as a binary image of 0 and 1 (black and white) to differentiate between the 2 labels.

6.5 Object Map Creation

After the determination of cascaded threshold image next step includes creation of object map which represents boundary and location of an object in an image .For that we have to initialize certain parameters such as: open Border Cut which decides to cut object boundary, Border Cut Value can be used to cut the object boundary but for sharp and overlap cutting smoothing value is used. These can be used to represents Hyper complex representation which gives object in terms of part.

For each part we have to find input feature map which gives components in horizontal. Vertical and in diagonal which can be used to gives hyper complex representation which can be filtered out for combining each part by using Gaussian filter. We choose the Gaussian filter due to its desirable characteristics i.e. its step response contains absolutely no overshoot, uses a different kernel that represents the shape of a Gaussian distribution, separable Gaussian functions.

6.6 Detected object

After the determination of object map we have to represent this map in RGB which represents detected objects in an image in colored form.

6.7 Output parameters

We are calculating speedup factor, PSNR.

6.7.1 Speedup factor

Speedup is defined by the following formula:

$$S_p=\text{Time2}/\text{Time1} \tag{5}$$

Where, Time1 is starting time for object detection.
Time2 is ending time for object detection

6.7.2 PSNR

Comparing restoration results requires a measure of image quality.. Given a noise-free $m\times n$ monochrome image I and its noisy approximation K, MSE is defined as:

$$\text{MSE} = 1/mn \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [\, I\,(i,j) - K\,(i\,,\,j)]^2 \tag{6}$$

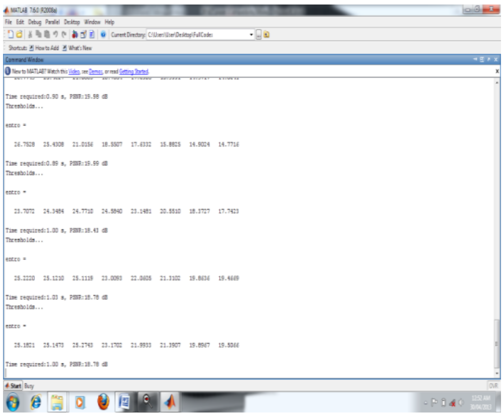
The PSNR is defined as:

$$\begin{aligned} \text{PSNR} &= 10 \log_{10}(\text{MAX}^2{}_1/\text{MSE}) \\ &= 20 \log_{10} (\text{MAX}_1/\surd\text{MSE}) \\ &= 20 \log_{10} (\text{MAX}_1) - 10 \log_{10}(\text{MSE}) \end{aligned} \tag{7}$$

Here, MAX_I is the maximum possible pixel value of the image.

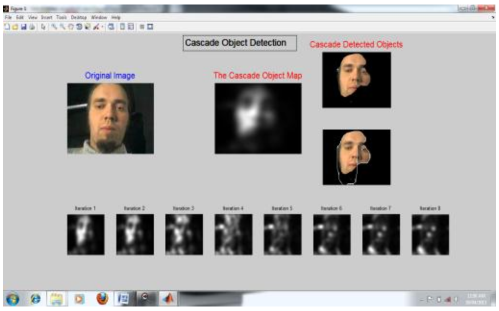
7. Results

7.1 Values of Threshold for Video


MATLAB 7.6.0 (R2008a)
File Edit Debug Parallel Desktop Window Help
Current Directory: C:\Users\User\Desktop\FullCode
Shortcuts How to Add What's New
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Start Busy
OVR

```
Time required:0.90 s, PSNR:19.98 dB
Thresholds...

entro =

   26.7528   25.4308   21.0156   18.5507   17.6332   15.8825   14.9024   14.7716

Time required:0.89 s, PSNR:19.99 dB
Thresholds...

entro =

   23.7072   24.3484   24.7718   24.5840   23.1481   20.5510   18.3727   17.7423

Time required:1.00 s, PSNR:18.43 dB
Thresholds...

entro =

   25.2220   25.1210   25.1119   23.0093   22.0605   21.3102   19.8636   19.4669

Time required:1.03 s, PSNR:18.78 dB
Thresholds...

entro =

   25.1821   25.1473   25.2743   23.1702   21.9933   21.3907   19.8967   19.5066

Time required:1.00 s, PSNR:18.78 dB
```

7.2 Object map and Detected Object for Video



References

M. Kearns and U. Vazirani(1994),An Introduction to Computational Learning Theory,MIT Press.

P. Felzenszwalb and D. McAllester (2010), Object detection grammars. Univerity of Chicago, CS Dept., Tech. Rep.

P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. (2009),Object detection with discriminatively trained part based models, PAMI.

P.Felzenszwalb,D.McAllester,and D. Ramanan (2008),A discriminatively trained, multiscale, deformable part model, In CVPR.

J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid (2007), "Local features and kernels for classification of texture and object categories, A comprehensive study," International Journal of Computer Vision, vol. 73, no. 2, pp. 213–238.

S. Zhu and D. Mumford(2007), A stochastic grammar of images, Foundations and Trends in Computer Graphics and Vision, vol. 2, no. 4, pp. 259–362.

P.Felzenszwalb and D. McAllester(2007),The generalized architecture, Journal of Artificial Intelligence Research, vol. 29, pp. 153–190.

Y. Jin and S. Geman (2006),Context and hierarchy in a probabilistic image model, in IEEE Conference on Computer Vision and Pattern Recognition.

L. Bourdev and J. Brandt. (2005),Robust object detection via soft cascade, In CVPR.

P. Felzenszwalb and D. Huttenlocher (2005), Pictorial structures for object recognition, IJCV, 61(1):55–79.

N. Dalal and B. Triggs(2005),Histograms of oriented gradients for human detection, in IEEE Conference on Computer Vision and Pattern Recognition.

H. Schneiderman (2004),Feature-Centric Evaluation for Efficient Cascaded Object Detection,In: Proc IEEE Computer Society Conference on Computer Vision and Pattern Recognition

M. Elad, Y. Hel-Or, and R. Keshet (2002), Pattern detection using a maximal rejection classifier,PRL, 23(12):1459–1471, 2002.

M. Everingham, L. Van Gool, C. K. I.Williams, J.Winn, and A. Zisserman (2008),The PASCAL VOC Results.

M. Everingham, L. Van Gool, C. K. I.Williams, J.Winn, and A. Zisserman (2009),The PASCAL VOC Results.