Automatic generic Region-Of-Interest selection for video surveillance applications

Citation for published version (APA):

Document status and date:
Published: 01/01/2014

Document Version:
Accepted manuscript including changes made at the peer-review stage

Please check the document version of this publication:
• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher’s website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.
Automatic generic Region-Of-Interest selection for video surveillance applications

S. Javanbakhti, X. Bao, S. Zinger, P. H. N. de With
Video Coding and Architectures Research group,
Electrical Engineering Dept., Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, the Netherlands
{S.Javanbakhti, X.Bao, S.Zinger, P.H.N.de.With}@tue.nl

Abstract
The value of using cameras in surveillance is further augmented when the surveillance system can take decisions autonomously by means of video analysis. For dynamic scene analysis, automatic detection of informative regions such as Regions-Of-Interest (ROI) is a challenging task for surveillance applications due to the large variations of the scene material. Our hypothesis is that if such an ROI is detected, a further advanced video analysis can be applied later exclusively to that ROI. In this paper, we employ a DCT for an ROI detection, since it provides a compact representation of the signal energy and the computation can be implemented at low cost [1]. We verify the usefulness of our hypothesis by cascading our ROI detection techniques with a typical object detector [3] as used in surveillance cases, to evaluate the attractively of the concept. We validate this approach on two different datasets and also compare our algorithm with a number of simple, fast ROI detection techniques. The experimental results show that our proposed approach outperforms the other methods in recall, precision, as well as in computational time.

1 Introduction
Video surveillance has become a key component in ensuring security at airports, banks, casinos etc. and is increasingly used elsewhere in public places. The objective of visual surveillance is to alert police or security officers, where the system applies multiple cameras to look at different locations at the same time. The value of using cameras in surveillance is increasing when the surveillance system can take decisions autonomously by means of video analysis. Studies in psychology and cognition have found that, when looking at an image, our visual system would first quickly focus on one or several “interesting” regions of the image prior to further exploring the image contents. These regions are often called salient regions or Regions-Of-Interest (ROI) [4]. If such an ROI is detected, a further advanced video analysis can be applied later exclusively to that ROI. Our hypothesis is that this provides a significant gain in efficiency in terms of the number of pixels needed to be explored, as compared to analyzing the complete scene. By providing an ROI to a security officer, without the further in-depth analysis, the time for the first-stage visual analysis of the scene is reduced significantly, so that the officer can immediately respond to alarming situations. It can also reduce both the required bandwidth for sending the information and the memory needed for storage.

However, the methods currently used for ROI detection are quite complex and time-consuming. These methods involve training in order to learn the appearances of a specific ROI, before the system can search for it. This learning is time-consuming, ROI detection is also called salient region detection in the literature [4]. ROI detection techniques can be divided into three broad families of approaches: bottom-up, top-down or a combination of both. Bottom-up cues are mainly based on pixel and feature
characteristics of a visual scene, whereas top-down cues are determined by cognitive phenomena, like knowledge, expectations, reward, and current goals [4]. Bottom-up approaches are usually fast in execution, while top-down techniques are slower and task-driven [5]. In the literature, three features are often used for bottom-up ROI detection: intensity (or intensity contrast, or luminance contrast), color, orientation and motion. Intensity is usually implemented as the average of three color channels. Orientation is obtained by a convolution with oriented Gabor filters, or by the application of oriented masks. Motion is based on eye/head movement which can be derived from recordings in freely behaving Rhesus monkey [6]. In this paper, our approach is based on bottom-up techniques using frequency features. For the complete system, we aim at simple and fast methods suitable for real-time operation. The real-time aspect also explains why we have adapted the DCT, as this transform can be efficiently implemented.

To measure the potential efficiency gain, we cascade our ROI detection techniques with a typical object detector [3]. Object detection is challenging, because the appearances and size are highly variable even for the same type of object. Appearance-based techniques have been widely exploited for object detection. Haar-like and Histogram of Oriented Gradients (HOG) features are frequently applied together with classifiers such as Adaboost or SVM, to train appearance-based detectors. For these appearance-based methods, exhaustive search is performed in an image with e.g. a sliding window with various scaling factors, which is rather computationally expensive. However, if we have prior knowledge of ROIs in the image and apply the detector in those regions, the complexity can be largely reduced to enable real-time applications. In turn, the ROI detection should not miss important objects, when it is applied as a initializing stage of processing. The paper is organized as follows. Section 2 describes different approaches for ROI detection. Section 3 presents our results. Conclusions and discussion are provided in Section 4.

2 Region-Of-Interest detection approach

For detecting an ROI, we compare a number of simple, fast ROI detection techniques based on various features, such as sum of edge pixels using a Sobel edge detector, the number of connected components based on the intensity values of neighboring pixels, the number of straight lines found by the Hough transform and the entropy of the frequency-domain features of a DCT.

2.1 Sum of edge pixels

In this approach, the sum of edge pixels is used to detect an ROI. First, a Sobel edge detector is applied to an input frame. Then we combine resulting gradients in both horizontal and vertical directions to calculate the gradient magnitude of each pixels. Subsequently, we impose a threshold to the gradient magnitude of each pixel, in order to eliminate weak edges. After removing weak and small edges, we divide each resulting frame into a number of blocks. For deciding whether each block is an ROI or not, the algorithm compares the number of edge pixels in each block with a threshold, and if this number is above the threshold, it is considered as an interesting region.

2.2 Connected components

In this approach, we compute the number of connected components to find ROIs in an image. First, we apply a Sobel edge detector to the input frame. Then we compute the connected components of the resulting frame to label edge pixels, based on the intensity values of neighboring pixels. In this work, an 8-connected neighborhood is employed.
The result of the connected component algorithm is a labeled map, where each label contains a group of connected edge pixels. We explore the number of edge pixels in each label. When the number of edge pixels is lower than a predetermined threshold, then edge pixels from this label are discarded. This criterion avoids the occurrence of small edges that cannot be removed in the previous approach (see Section 2.1). For deciding whether each block is ROI or not after removing the small edges, the resulting map is divided into a number of blocks. Then, for each block the algorithm checks whether it contains any labels or not. The block is considered as an ROI if it contains any labels. This approach is more reliable compared to using only an edge detector.

### 2.3 Hough lines

In this approach we calculate the number of straight lines found by the Hough transform. The Hough transform is known as an algorithm which can extract lines effectively [7]. Straight lines can be parameterized in the polar Hough parameter space by their length, $\rho$, and orientation, $\theta$, of the normal vector to the line from the image origin [8]. This length parameter $\rho$ is specified by

$$
\rho = x\cos(\theta) + y\cos(\theta),
$$

where $(x, y)$ denotes a pair from a set of image coordinates, which are lying on a straight line. The transform is quantizing the Hough parameter space into accumulator cells. As the algorithm runs, each pixel $(x, y)$ is transformed into a discretized $(\rho, \theta)$ curve and the accumulator cells which lie along this curve are incremented. Resulting peaks in the accumulator array represent strong evidence that a corresponding straight line exists in the image.

To detect straight lines in an image, we first apply the Sobel edge detector to the image. Then the best scoring results after applying a Hough transform provide the straight lines in the binary image containing the edges. The number of straight lines is limited by a number of local maxima values which is selected to choose the best scoring results. Therefore, to avoid false rejections, first we divide a frame into a number of blocks. Later, we apply the Hough transform to each block. For deciding whether each block is an ROI or not, the number of straight lines associated with each block is compared to a predetermined threshold. If this number exceeds the threshold, the block is considered as an ROI.

### 2.4 Discrete Cosine Transform (DCT)

Here, the ROI detection algorithm is based on analyzing the information content of the image in the pseudo-frequency domain by computing the entropy of the involved DCT coefficients. The DCT provides a compact representation of the signal energy. To this end, we apply 2D DCT to each pixel block. The 2D DCT transforms the block of the input frame to a coefficient matrix, where each coefficient represents the degree of which a certain cosine function is present in the block of the input frame. If each block has a high peak only in low frequency and no other significant peaks, it is considered as not an informative block. Since the DCT value on location $(u, v) = (0, 0)$ in the coefficient matrix is rather large and represents the average or DC value, we remove this high value to evaluate the rest of the coefficients. To avoid small DCT coefficients at higher frequencies due to minor intensity changes and reduce influence of noise, we again define a threshold. To this end, we compute the entropy of the DCT coefficients for each block [7] to measure the information content. The entropy is defined as [9]:

$$
H = - \sum_{i=1}^{N} p_i \log_2 p_i,
$$
where $N$ is the size of image and $p_i$ contains the probability of the intensity value $i$ averaged over all pixel locations. The number of bins in the histogram is specified by the image type and its definition. In our case, the input frame is converted to a gray-scale image and we employ 256 bins which corresponds to the number of gray levels.

For defining each block is considered an ROI or not, we compare the magnitude of the DCT coefficients of the current block with the the entropy of this block. If at least one DCT coefficient magnitude of each block is above the entropy of this block, then this block is considered an ROI.

3 Experimental results

3.1 Initial test with still images

For evaluating the above four different ROI detection techniques, we have applied them to 30 images from the Caltech Pedestrian Detection Benchmark [2], and created ground truth for these images by manually annotating them. We have defined empirical thresholds for each technique after extensive experimentation to optimize their performance. We have used 10-frame intervals for temporal filtering of the frame-based ROI detections. This prevents flickering in the output due to small temporal changes and noise.

Figure 2 shows the results of applying the proposed ROI detection techniques to a sample frame. In this experiment, the block size for all four techniques is constant and equal to $32 \times 32$ pixels.

Table 1 presents the average precision and recall rates of the ROI detection approaches on 30 images of the Caltech Pedestrian Detection Benchmark [2]. It shows that the results of the ROI detection obtained from the DCT-based approach are similar to the results of the Hough-based technique. Table 2 shows that the edge-based approach is the fastest and the Hough-based technique is the slowest algorithm. By analyzing Table 1 and Table 2, we conclude that the DCT-based approach is the most attractive compared to the other approaches. Therefore, for further analysis of the DCT-based approach, we have also evaluated it on our highway video sequence. This sequence consists of 40 frames with a resolution of $1280 \times 960$ pixels and a frame rate of 25 fps. We have obtained a precision and recall of 39% and 75%, respectively. The average precision and recall rates of the DCT-based approach for 70 images of both Caltech Pedestrian Detection Benchmark [2] and our dataset are 50% and 77%, respectively.

For our application, we consider the recall rate to be more important than precision because we aim at extracting as many ROI regions as possible from the image. For surveillance application, it is important to present to the security officer all ROI regions which are present in a given frame so that no important information is missed.

3.2 Video surveillance use case

To measure the potential efficiency gain when applying ROI detection in video surveillance applications, we cascade our DCT-based ROI detection technique with an object detector [10], as depicted in Figure 1. Our object detector is trained using Haar-like features combined with Adaboost algorithms, while the cascaded detection framework is evaluated for a car detection task. To train the car detector, we have selected 100 images from the “cars 2001(Rear), Caltech” dataset [11] as training set. To test our detection framework, we have chosen 50 frames from our highway video sequence as the test set.

We consider the size of the car in the test set to be in the range of $20 \times 20$ pixels.
Table 1: Precision and recall rates for the ROI detection algorithms on Caltech Pedestrian Detection Benchmark [2].

<table>
<thead>
<tr>
<th>Approach</th>
<th>Precision (%)</th>
<th>Recall (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edge-based</td>
<td>62</td>
<td>50</td>
</tr>
<tr>
<td>Connected component-based</td>
<td>60</td>
<td>57</td>
</tr>
<tr>
<td>Hough-based</td>
<td>61</td>
<td>80</td>
</tr>
<tr>
<td>DCT-based</td>
<td>61</td>
<td>79</td>
</tr>
</tbody>
</table>

Table 2: Average computational time of the different methods per frame, calculated over 5,454 frames of Caltech Pedestrian Detection Benchmark [2].

<table>
<thead>
<tr>
<th>Approach</th>
<th>Time (milliseconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edge-based</td>
<td>63</td>
</tr>
<tr>
<td>Connected component-based</td>
<td>105</td>
</tr>
<tr>
<td>DCT-based</td>
<td>150</td>
</tr>
<tr>
<td>Hough-based</td>
<td>225</td>
</tr>
</tbody>
</table>

and $120 \times 120$ pixels. We use a sliding window with the size of $120 \times 120$ pixels to scan the image. The scanning step is 10 pixels in both vertical and horizontal directions. If the number of the ROI pixels in the image patch within the sliding window is below 50%, we expect no car inside the image patch, so that we do not apply the car detector. Otherwise, we apply the car detector for the image patch. We have compared the detection results of the new cascaded framework with the results obtained using only the Haar-like object detector.

On the average, for 50 frames of our test set and without using ROI, the car detector is applied to 9,744 image patches. When applying the detector only to ROI regions, only 943 image patches are checked by the detector. The proposed cascaded detection framework using DCT-based technique leads to applying the car detector, which is a computationally expensive algorithm, to only 9.7% of the total amount of image pixels. The average detection rates of car detection with and without using ROI are 80% and 90%, respectively. Figure 3(b) shows the result of the DCT-based approach with $8 \times 8$ block size on our highway video sequence. Figures 3(c) and (d) present the detected cars in that frame without and with the DCT-based ROI detector in cascade, respectively.

Figure 1: Framework of the cascaded object detection approach.

4 Conclusions and discussion

In this paper, we have presented our ongoing research on scene analysis, namely on employing generic ROIs for outdoor video surveillance as a preceding stage for further precise processing such as an object detector. Our hypothesis is that this will provide a significant gain in efficiency in terms of the number of pixels needed to be explored, compared to the conventional analysis of the complete scene. By already providing a generic informative ROI to a security officer without further in-depth analysis, the first-stage visual analysis of the scene is optimized significantly, so that the officer can...
Figure 2: (a) sample frame of Caltech Pedestrian Detection Benchmark [2] video sequence, (b) created ground truth for this frame (black blocks are not ROI), (c) edge-based method, (d) connected component-based method, (e) Hough-based method and (f) DCT-based method.
immediately respond to alarming situations. It can also reduce both bandwidth needed for sending the information and memory needed for storage.

To this end, we have considered a number of simple, fast ROI detection techniques based on various features, such as the entropy of DCT coefficients, sum of the edge pixels, number of connected components and the number of straight lines found by the Hough transform. This means that the criterion for each technique is matched to the nature of that technique. We define empirical thresholds for each of technique after extensive experimentation to optimize the performance. We have used 10-frame intervals for temporal filtering of the frame-based ROI detections in order to prevent the flickering in the output due to small temporal changes and noise.

Our experiments show that the detection results of the ROI detection obtained with the DCT-based approach are similar to the results of the Hough-based technique. Both DCT- and Hough-based approaches provide better results compared to the other ROI detection techniques. Additionally, the DCT-based approach is well-suited for real-time image analysis because it allows parallel processing as the DCT can be computed for each image block independently and thus in parallel. Accordingly, we adopt the DCT-based ROI detection approach for validating its efficiency in a video surveillance use case. The major contribution of this study is developing an accurate ROI detection approach, while maintaining low computational complexity, so that it is suitable for real-time implementation in embedded video surveillance. Furthermore, high adaptivity to new scene types is an additional advantage of the proposed approach.

We have validated the efficiency of our hypothesis by cascading the proposed DCT-based ROI detection technique with an object detector to measure the potential efficiency gain. The experimental results show that such an ROI provides a significant gain in efficiency with respect to the explored number of pixels, as compared to analyzing the complete scene. Even though combining the usage of ROI extraction and the Haar-based object detector slightly decreases the precision, the efficiency is considerably increased, as only 9.7% of the image pixels is analyzed by the object detector. Concerning the future work, we are planning to research a new criteria for threshold selection for the DCT-based ROI detector. Currently the threshold is adaptively chosen.

Figure 3: (a) Sample frame of our video sequence, (b) DCT-based ROI selection, (c) Haar-based car detector without using ROI, (d) Haar-based car detector using ROI (Red rectangles represent the results of the car detector).
for each block in which the DCT is performed. In order to better exploit the spatial context of the scene, we will add the information from the neighboring blocks to the algorithm of threshold selection. We expect that this will increase the precision of the ROI detection results.

Acknowledgement

We are grateful to Patrick Langenhuizen for his enthusiasm and hard work while developing the ROI detectors during his internship research.

References


