A Framework for Designing Natural Object Detector in Video

Wanjun Jin; Zhiyan Tang; Zeyu Zhao

Department of Computer Science & Engineering
Fudan University, Shanghai, P.R. China
{jwj, zytang, fdzzy}@fudan.edu.cn

Abstract

This paper presents a framework for detecting natural objects in video. We analyze each key frame of video to determine if there is any interesting natural object inside. After key frames are extracted by our Shot Boundary Detection (SBD) system, natural object detection in a key frame is done by following such a general pipeline. First, a novel representation CF*IRF (Color frequency and Inverse Region Frequency) is developed to detect all possible regions of the target object. Then those regions are validated by texture features and other color features. Finally the shape and location information of the regions give the final answer of the present of the target object. In this paper we also discuss the problem of search task’s evaluation. A set of criteria is proposed, by which different search tasks can be classified and analyzed.

Keywords

Object detection, video search, CF*IRF

1. Introduction

Automatic event detection in video becomes more important with increasing volume of digital video. Some projects on event detection focus on tracking foreground, such as car, face, fire etc. The color, structure, and motion information are selectively used for specific foreground object tracking. Some projects focus on background detection, since background has different attributes from foreground such as little active motion, no definite shape and large area.

We have participated TREC Video Retrieval Evaluation (TRECVID) workshop for 3 years. The main goal of the TRECVID is to promote progress in content-based retrieval from digital video via open, metrics-based evaluation. Most search tasks and part of feature extraction tasks in TRECVID have tight relationship with event detections, both foreground detection and background detection.

From our experience in TRECVID 2002, search task is a very difficult problem and it is always unrealistic to devise the generic algorithm which can work well for all search tasks. The main reasons are that visual and motion features of the scene differ greatly from one to another, and limited query samples provide information far less than needed. For example, the method used to search grassland may have poor performance when searching aircraft, and given an image containing the sunny Golden Bridge as the query, it is unlike to retrieval images containing the Gold Bridge in cloudy days or at night. Some special search tasks, such as “finding person”, “finding person X”, can be regarded as the problem of face detection and face recognition, which much effort has been made to solve. As to other tasks, few well-devised methods are available. So usually, we just take it as a content-based image retrieval problem and attempt to use various visual features, as well as Automatic-Speech-Recognition (ASR) results, for retrieval. Though ASR is proven to be quite useful for most search tasks, we still wonder how visual cues can be used for searching objects in video and under which condition, visual cues are helpful to search tasks and under which condition they get things worse.

Since much work has been done in the area of object detection, here we just concern some work which is specially devised for event detection in video and is relevant to our search tasks in TRECVID. Vehicle detection has been long studied for the traffic surveillance [1][2] and the main information used is motion between images. Fire detection [3][4] can be useful in categorizing movie according to the level of violence. Recently the research of face detection, for its importance, draws much attention [5][6][7]. There is also some work for background detection [8], characterized by little active motion, large area and no definite shape.

In this paper, we present a general pipeline for video search tasks, i.e. any specific search task can be followed in the same pipeline, but specified steps have to be adjusted according to the content of real task. Here we focus on the detection of natural objects which appear frequently in outdoor scenes, and often have uniform color and texture and have little global motion, such as vegetation, sun, mountain, fire etc. Knowledge of existence of such objects in video will be helpful for many search tasks in TRECVID. What’s more, although we limit our experiment on those natural objects for their simplicity, the pipeline we proposes still take effect in detecting other objects in video.
2. Key frame extraction

First, key frames of the video are extracted by using our Shot Boundary Detection (SBD) system[9] which is stated in this section. After shot segmentation, we take certain frame of each shot as its key frame according to the type of the shot boundary.

In our SBD system, FFD (Frame to Frame Difference) calculated by Luminance Difference and Color Histogram Similarity are used to detect the Shot Changes. We use two thresholds $\theta_{CC}$ and $\theta_{G}$, which are calculated automatically according to the FFD value histogram in 500 frames, to detect if there is a clear FFD value change caused by Shot Changes. Then Flashlight Detection and Motion Detection are applied for candidate Shot Changes to remove the false alarms of Cut and Gradual. The parameters used in the system are trained and adjusted based on the TREC-10 Video Library. If cut is detected, we take the first frame of the next shot as the key frame. If a gradual shot transition is detected, we take the middle frame of the next shot as the key frame.

3. A general pipeline for natural object detection

We propose a general pipeline for natural object detection, which is shown in Figure 1. The line connecting boxes stands for certain operation and the rightmost parts are features used to judge whether the target object exists.

3.1 Find the candidate region

Color is one of the most frequently used visual cues in image/video retrieval. So in the first step, we consider the color property of the object to detect possible regions which may contain the target object, named Candidate Regions. Most natural objects have limited number of typical colors. So we try to use CF*IRF (Color Frequency and Inverse Region Frequency), a new feature we designed for detecting specific object, to capture those typical colors from the training samples and later to determine the existence of the target object in new video. It is assumed that the more typical colors a new video have, the more possible it has a desired object inside. To our knowledge, our proposed CF*IRF metrics is novel and no previous work introduces the similar notion in object detection scenario.

The concept of TF*IDF (term frequency and inverse document frequency) is widely used in document retrieval to calculate the correlation of two documents. Here is the formulation.

$$W_{Di} = \frac{D}{D_i} \log\left(\frac{T}{T_i}\right)$$

where $W_{Di}$ is the weight of keyword $i$ in target document, $D$ is the number of all keywords in target document, $D_i$ is the number of the keyword $i$ in target document, $T$ is the number of all keywords in the whole corpus and $T_i$ is the number of the keyword $i$ in the whole corpus. The basic idea of TF*IDF is that the more often one keyword shows in certain document, the stronger it is to describe the document (TF), and the less frequently one keyword shows in other documents, the more unique it is (IDF).

To some degree, object search in image can be formulated as the similar problem. If we take the pixel’s color as the “term”, the object region as the “document” and the whole image as the “document corpus”, we can naturally introduce the concept of CF*IRF (Color Frequency and Inverse Region Frequency) for color-based object detection. The CF*IRF is defined as follows,

$$W_{C} = \frac{R}{R_i} \log\left(\frac{C}{C_i}\right)$$

where $W_{C}$ is the weight of color $i$ for the object region, $R$ is the number of points in object region, $R_i$ is the number of the points with color $i$ in object region, $C$ is the number of points in the whole image and $C_i$ is the number of points with color $i$ in the whole image. The basic idea of CF*IRF is that the more often one color shows in certain region, the stronger it is to describe the region (CF), and the less frequently one color shows in other regions, the more unique it is (IRF). With CF*IRF, we can get the weight of specific color $i$ for the target object. If a color has large CF*IRF, it has strong capacity to describe the object.

The CF*IRF color lookup for the specific object is constructed by using the following steps.

a) Create pairs of training images – each pair consists of a color image, and a Boolean mask, which specifies the locations at which the target object occurs. For every pixel in each image which represents a color that is being searched for, there should be a “1” in the corresponding location in the Boolean mask, and a “0” for every background location. From our tests, we found ten training
images from our data set to be sufficient to construct an effective color predicate. In order for this to be sufficient, it is necessary to ensure a variety of scenes and a variety of illumination condition.

b) Divide HSV space to 16*8*8 and then construct an empty color lookup, with the format of HSV bin number and its CF*IRF value. Construct a local color histogram as follows: for every pixel in the image, if the value in the corresponding mask location is “1” then add 1 to the corresponding bin of the local histogram. Otherwise, if the value in the corresponding mask location is “0”, do nothing. Then, construct a global color histogram as usual.

c) For every non-empty bin in local color histogram, calculate the CF*IRF value of it by using the definition mentioned above, with the slight difference that the color i in the original definition is replaced by all the color in the bin, i.e. we calculate the CF*IRF value on the scale of color bin. If the corresponding HSV bin number doesn’t exist in the color lookup, add a new item to it, with the bin number and corresponding CF*IRF value. If the corresponding HSV bin number exists, set its CF*IRF value the larger one between current value to the existing value in that item.

d) Repeat the step b and step c for every training image pairs. And finally threshold the CF*IRF value to the desired level.

Based on the above preparation, we come to the definition of the Color Importance:

\[
CI(C_i) = \frac{W_{C_i}}{\sum_{j=1}^n W_{C_j}}
\]

(3)

Basically, the importance of a color is its CF*IRF normalized over all regions in an image such that the sum of all color importance weights is equal to 1. Thus we can construct a CI lookup from CF*IRF color lookup (on the scale of bin), resulting a function of CI(C_i), which, given an (H,S,V) triple, will transform it to the corresponding HSV bin number and return a value, indicating whether or not an input color is among the desired colors and if yes, how importance the input color is.

With the CI function, it is quite easy to detect candidate regions for the target object in image. For every image, if

\[
\sum_{p \text{image}} CI(\text{color}(p)) \leq \text{Threshold}
\]

(4)

(where p is the pixel of the image and color is a function returning the color of the pixel), we can declare that the image probably don’t have the target object and just discard it. If not, construct a Boolean mask image. For every pixel, set 1 in the mask image if the CI function at this point is non-zero and set 0 otherwise. Thus a Boolean image is constructed where all 1-points compose candidate regions, which we call as Candidate Regions mask.

A refining step is often needed next to diminish the effect of the noise. First we discard all the regions whose size is smaller than a threshold MINAREA (here we choose 50 pixels). Then we use the morphologic operation “open” on the whole Candidate Region mask. The morphologic core can be selected with a small square and operation times should be less than three. (Here we choose the value of 3 and 2 separately)

3.2 Region validation

After we find the candidate regions of the image, we have to validate them by using other features. The first feature used for region validation is texture. We select Gabor wavelet features [10] for its success in many applications such as texture classification and pattern retrieval.

Given an image I(x,y), its Gabor wavelet transform is then defined to be

\[
W_{mn}(x,y) = \int I(x_1,y_1) g_{mn}^*(x-x_1, y-y_1) dx_1 dy_1
\]

(5)

where * indicates the complex conjugate. Since it is assumed that local texture regions are spatially homogeneous, we use the mean \(\mu_{mn}\) and the standard deviation \(\sigma_{mn}\) of the magnitude of the transform coefficients as the feature components. The feature vector is then constructed using \(\mu_{mn}\) and \(\sigma_{mn}\) corresponding to different m and n.

\[
\mu_{mn} = \frac{1}{N} \int \int |W_{mn}(x,y)| dx dy
\]

(6)

\[
\sigma_{mn} = \frac{1}{N} \sqrt{\int \int (|W_{mn}(x,y)| - \mu_{mn})^2 dx dy}
\]

(7)

\[
\int = [\mu_{00}, \mu_{00}, \mu_{01}, \cdots, \mu_{nn}, \sigma_{nn}]
\]

(8)

Where N is the number of pixels in the neighbor region. We construct a feature vector for some sample points in training images, put them as training samples to a SVM (Support Vector Machines) classifier and get the final
4. Experiment

As the time being, we try to test our approach on the task of detecting grass, part of the search task in TRECVID 2003 which is required to find segment contains living vegetation in its natural environment. Experiments show that our CF*IRF approach is efficient. Most grass pixels are detected correctly, as long as grass images under different illumination condition have been chosen as training samples. We use our image database containing 562 scenic images, in which there are altogether 74 images containing grass. Ten of them are selected and manually segmented for training. Then we retrieve the first 64 images in the rest corpus and get 58 hits. The precision is 90.6%. The detection process of one hit is shown below, with the original image, candidate region mask, refined candidate region mask and validation mask. Through experiments on image set, we find that except for those grass images under extremely unusual condition, our algorithm performs well.

Table 1 Experimental results on video

<table>
<thead>
<tr>
<th>Video ID</th>
<th>Fs</th>
<th>Ks</th>
<th>Gs</th>
<th>KGs</th>
<th>M</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseball</td>
<td>2274</td>
<td>19</td>
<td>14</td>
<td>14</td>
<td>13</td>
<td>5</td>
</tr>
<tr>
<td>Outdoor 1</td>
<td>1865</td>
<td>23</td>
<td>9</td>
<td>9</td>
<td>7</td>
<td>12</td>
</tr>
<tr>
<td>Outdoor 2</td>
<td>1651</td>
<td>14</td>
<td>10</td>
<td>9</td>
<td>9</td>
<td>4</td>
</tr>
</tbody>
</table>

Fs: number of frames
Ks: number of key frames (shots)
Gs: number of shots with grass in fact
KGs: number of key frames with grass
M: number of correctly detected shots with grass
N: number of correctly detected shots without grass

5. Summary and discussion

In this paper, we propose a framework to detect interesting natural object in video. We propose a novel representation CF*IRF to reflect the importance of certain color to target object, and later use it to detect candidate regions. We introduce a general pipeline for designing natural object detector. Briefly speaking, candidate regions are first detected by using our proposed CF*IRF, then are validated by more complex texture and color cues. And at last, regions’ shape and temporal property are used for final conclusion.

When a new search task is available, the problem we have to consider is how to evaluate its difficulty and how to get a start to solve it. Generally, we think search tasks can be classified and analyzed according to following criteria.
**Number of color patterns**: The object may have only one color pattern (1), a limited number of color patterns (n) or countless color patterns (∞). Object with various color patterns only caused by different illumination condition is considered as another situation (n').

**Number of colors in one color pattern**: For each color pattern, the object may have only one color (1), or many colors (n).

**Texture**: The object may have one kind of texture (1) or various kinds (n). The object without obvious texture is marked “N/A”.

**Shape**: The shape of the object may change with time (changeable), or be amorphous (amorphous).

**Motion**: The global motion the object can have is little (static), unknown (unknown), or can be formulated several typical motion mode (formulated).

According to those criteria, several typical search tasks can be evaluated as follows. Since some subjective factors are inevitably included when evaluating such tasks by people and semantic meaning of the task may have some inherent ambiguity, the evaluation result given by different people may differ slightly.

We can see that detecting sky, grass is comparatively easy task because its color and texture are uniform and the shape is often static. The problem is that different illumination condition has to be concerned. One approach to solve this is to model several light models for different illumination conditions [8]. For fire and smoke, temporal variance is a good cue to distinguish them. As to mountain and car, things turn to be a little bit difficult since each case has one dominating color but the dominant color is not fixed. It is often hard to develop several typical color models. Our CF*IRF can perform well in this situation. And considering other shape features of the object, such as the symmetry of car’s shape and triangle-like shape of mountain, will help to solve the task. Carrot and zebra often have some dominating colors for one case, which makes the problem harder. A possible approach for this task is that at the training stage, the correlation of colors in each case is recorded for further use. Objects with general concept, such as animal, may be too complex to handle.

We think the only way to do such tasks is to narrow down the concept and find some subset of it.

**References**


| **Table 2**: Evaluation of several typical search tasks |
|---|---|---|---|---|
| **Color pattern No.** | **Color No. in one pattern** | **Texture** | **Shape** | **Global motion of object** |
| Fire | 1 | 1 | N/A | changeable | static |
| Sky(no cloud) | n' | 1 | 1 | steady | static |
| Grass | n' | 1 | 1 | steady | static |
| Smoke | n | 1 | N/A | changeable | unknown |
| Car | n | 1 | N/A | steady | formulated |
| mountain | n | 1 | 1 | steady | static |
| Animal | ∞ | n | n | amorphous | unknown |