A HOUGH TRANSFORM BASED CITYSCAPE CLASSIFIER

Alain PUJOL, Liming CHEN
alain.pujol@ ec-lyon.fr, liming.chen@ec-lyon.fr

Ecole Centrale de Lyon - 36 avenue Guy de Collongue - 69134 Ecully Cedex – France

ABSTRACT

The task of image indexing within semantically meaningful categories remains a very challenging task in the domain of content based image retrieval. The source of this problem lies within the amount of information we draw from each image and the way we extract it, therefore the choice of the descriptors to use is a very important step in the design of a classifier.

In this paper we present a city/non-city classifier based on a specifically designed descriptor which is based on a variant of Hough transform called “Fast connective Hough transform” which allows us to deal with line detection efficiently. Here the purpose is to use as few good descriptors as possible to avoid high dimension features vectors which increase computing time dramatically.

1. INTRODUCTION

The well known “semantic gap” presents us with a problem that calls for more and more workarounds in order to simulate the human ability to interpret images. If we want to classify data into a broad category such as ‘cityscapes’ we have to use information deduced from appropriate low level descriptors themselves drawn from empiric observations. For instance, the task of identifying cityscapes has already been tackled by Vailaya et al. basing on the presence of strong vertical and horizontal edges reported in an edge direction histogram and a edge coherence histogram [1,2]. While we can assume this is an interesting property, this criterion lacks accuracy as on one side forest pictures can show similar edge properties and, on the other side, city pictures can be distorted by perspective or include some trees which will induce much noise in these histograms. Actually these detection criteria seem to work well on professional city pictures with very little noise.

Although the criterion seems adapted to our problem, the descriptor used here lacks discriminating information and robustness. Moreover, because pictures vary in scale and buildings have different architectures and colors, the usual color/texture descriptor does not provide enough discriminating power in this case.

Hough transform is a well known tool in image processing, and has been used for straight edge detection for a very long time [3]. In this paper we present a city/non-city classifier using segments detected by Hough transform as a base low level descriptor and we will discuss the various choices made in the elaboration of this classifier: because it also allows extracting position information from segments, spatial localization should noticeably increase the robustness of the detector. We introduce this descriptor as our goal here will be to use as little features as possible in order both to demonstrate its effectiveness and decrease processing time.

We will first describe in detail the variant of the Hough transform we used there and the way we chose to interpret the data it provided. Then we will discuss some other aspects of the classifier. Indeed, while the information extracted from line segments should be the main discriminating feature, we also decided to perform coarse color segmentation prior to line segment extraction because it allows dealing more efficiently with heterogenic images (such as a city picture showing some trees in the foreground). We then also added a simple color feature to our classifier in order to handle vegetation more efficiently. Data drawn from these features is processed by a feed forward two layer neural network which provides a classification decision.

Finally we will provide an evaluation of the devised classifier and talk about its performances regarding computing time and accuracy as well as future work planned on this subject.

2. LINE SEGMENT DETECTION USING FAST CONNECTIVE HOUGH TRANSFORM

2.1. From Hough transform to Fast Connective Hough Transform

Basic Hough transform has many flaws, influence of isolated pixels and computing time requirements being among the biggest noticeable problems.

Ben-Tzvi and Sandler [4] proposed a combinatory approach that lessens the influence of isolated pixels by first processing image blocks. Regarding processing speed, Xu [5] then Kälviäinen[6][7] have proposed new variants of the Hough transform in order to address these issues: computing time was reduced by using a randomly chosen subset of edge pixels.

Basing on these techniques, more variants (as referenced in surveys by Leavers [8] and Illingworth [9]) were proposed in order to use Hough transform to perform line segment detection. However, in terms of speed and robustness to noise, these methods still fall short of expectations when we need processing videos or a large number of consumer photographs for example. This led to developing our new method called “Fast connective Hough Transform” in order to match these needs [10]. Our method uses Hough space to determine local connectivity rather than using it as an accumulator space. We will now explain the mechanisms of this method.

2.2. Fast Connective Hough Transform

We aimed to address the previously mentioned flaws using both a local approach and diminishing the need to use costly primitives. First in order to process line segments, we choose to search sequentially edge points which are potential segment starting points; in that purpose we start from the origin point of the edge image and we scan it by incrementing x and then y. For each edge point we will detect the longest corresponding line segments and clear it afterwards in order to make sure we do not process the same segment twice.

When an edge pixel is detected, we need to scan all possible directions $0 < \theta \leq \pi$, in each direction we will search for edge points contributing to our line segment: this translates the connectivity of the points of the segment.

The principle is as follows: we explore each direction θ until no more contributing edge pixels are found, exploration of neighbourhood is an important part of the detection process as it can prove very time consuming. Here we will search for contributing neighbour edge pixels by incrementing either x or y coordinate (depending on θ, see below) and then calculating the other coordinate by using trigonometric formulae. For instance the coordinates of the neighbour to check at step n are as follows (E designates the integer part function):

$$0 < \theta \leq \frac{\pi}{4}; y_n = E((x_n - x_0)\tan(\theta)) + y_0; x_n = x_{n-1} + 1$$

$$\frac{\pi}{4} < \theta < \frac{\pi}{2}; x_n = E\left(\frac{y_n - y_0}{\tan(\theta)}\right) + x_0; y_n = y_{n-1} + 1$$

$$\theta = \frac{\pi}{2}; x_n = x_0; y_n = y_{n-1} + 1$$

$$\frac{\pi}{2} < \theta \leq \frac{3\pi}{4}; x_n = E\left(\frac{y_n - y_0}{\tan(\theta)}\right) + x_0; y_n = y_{n-1} + 1$$

$$\frac{3\pi}{4} < \theta < \pi; y_n = E((x_n - x_0)\tan(\theta)) + y_0; x_n = x_{n-1} - 1$$

These relative coordinates will be pre-computed for discrete values of θ and thus allow having only array reference operations for local edge pixel search. We then clear the longest segment to avoid processing it twice and store it if it is longer than a specified threshold. These processes make this variant of Hough Transform particularly fast. A study of its complexity indeed shows the number of iterations is almost of linear complexity regarding the number of edge pixels (it is not really linear though, as more edge pixels indirectly increase the iterations necessary to perform the local detection process as segments become longer). Testing made in [10] show that fast connective Hough transform performs 30 times faster than Randomized Hough transform.

2.3. Fast Hough Transform as a base descriptor

By using the Fast Connective Hough transform on an edge image obtained with a Sobel filter, we get a list of detected line segments each identified by its length, origin point and direction. As cityscapes show very important variations both in scale and in building size, we decide to take segment length into account only in order to filter “noise” segments generated, for example, by edge thickness. Each segment was therefore characterized by its orientation and the position of its origin which we translate into the distance to the upper left corner of the image in order to work in a lower dimension space.

By using a very basic clustering algorithm, we determine the main segment orientations within each region. These orientations are used to determine a reward coefficient which is determined by a set of Gaussian curves centered on the directions we are seeking. These Gaussian filters will thus produce an output of 1 for the directions we expect to find in a cityscape; namely 90°, 0° and 180°. We define the variance of each Gaussian filter empirically as a compromise between discriminating power and tolerance to deformations by perspective.

Each group of segments is determined not only according to its orientation but also by using a spatial localization feature. Its purpose is to handle separately groups of segments that were drawn from different spatial locations as well as to discard segments which would be well oriented but come all from the same place (typically produced by thick edges and/or poor quality pictures). The idea is to consider that a cityscape will have small patches of segments oriented in the same horizontal or vertical directions. By processing each group of segments, we produce a value between 0 and 1 which estimates whether the edge segments group should be considered as "city edges" or not.

The various values produced in the same region are classified into three categories which produce three indicators for the region: one for the evaluated vertical lines, one for the horizontal lines and one for the lines identified as "noise". This set of values represents the output of the descriptor for this region.

3. CLASSIFIER DESIGN

3.1. Coarse color segmentation

After a first testing phase we see that our classifier identifies quite well the edges properties we wished to extract. Yet some images still pose some problems: for example the presence of trees or other occlusion phenomena (a pedestrian walking in front the building) generate much unwanted noise. If we chose to ignore the problem, we will have to tolerate more and more "parasite" edges and thus weaken our classifier. To avoid this we decided to perform a color segmentation phase first: its purpose was to generate regions large enough to contain whole buildings so that we could extract lines segments in each of these regions and discard as much unwanted data as possible before applying Hough transform. This also meant we had to avoid over-segmentation as much as possible.

Our segmentation algorithm had to be a good compromise between speed and efficiency as well as ensure the creation of groups consisting of connected pixels only. We thus chose color structure code algorithm based on a merge and split region building process [11]. Along with defining thresholds in order to get larger regions, we wanted to have a color space that would allow us both to use Euclidian metrics and to ignore the lightness component (to avoid, as much as possible, splitting a region because of a shadow). These necessities led us to choose CIE-Lab color space and take into account only "a" and "b" components. This space being close to perceptual homogeneity, using Euclidian distance for color merging leads to visually satisfying results. By setting large threshold values in the CSC algorithm, we obtained the coarse segmentation we were looking for.

3.2. Basic color feature

By adding color segmentation we could handle heterogeneous images much better. However, we noticed that a specific category of images still had the classifier fail systematically with some pictures containing vegetation. While segmentation and noise filtering avoid considering isolated artificial structures as cities, it does not prevent false positives from some specific pictures. Indeed, sceneries such as artificial forests or close shots on grass fields present strong vertical lines, some horizontal lines and very little noise. In this case the best discriminating criterion appeared to be color; we thus decided to add a very simple color feature to our region feature vectors.

As we said earlier our purpose is to integrate as few features as possible and keep them as simple as possible as well. Therefore we used a sigmoid fuzzy membership function to map a coefficient between 0 and 1 to the distance between the region color and a reference green taken within lab space. The parameters of the sigmoid were determined through experimental measures taken on pictures representative of various seasons and lighting conditions.

3.3. Classifier structure and training

To characterize a region, we thus use a four dimension feature vector composed of values between 0 and 1. As indicated above, three of these are drawn from the Hough transform: evaluation of vertical lines, horizontal lines and noise; the fourth value being given by the color feature. These four values are used as input for a feed forward two layer neural network trained by back propagation. Two outputs respectively produce an evaluation for cityscape and non-cityscape.

Training of the network is made through manual annotation of each region of each image of the training set: the system shows each region with the detected segments and asks the user whether this data should induce classification into cityscape/non-cityscape or if this information is irrelevant. This way we integrate negative feedback in the learning process which allows drawing more information from smaller training sets. Off course, training sets should be as large as possible.

Decision for each region is then taken by simply comparing the two outputs, their closeness giving an idea of the confidence in the result. We arbitrarily defined that a cityscape was an image whose surface was covered by at least 25% of city regions.

4. PERFORMANCE EVALUATION, DISCUSSION AND FUTURE WORK

4.1. Performance evaluation

Not surprisingly, the classifier produced very good detection results for high quality professional city images; however our purpose here was to address the problems posed by consumer pictures which included most of the time "parasite" elements such as trees or pedestrians. We thus constituted a corpus of 250 web images which, for the most part, were specifically chosen to include those possible sources for either false detection or false rejections. The training set used a separate set of 50 city

images and 50 non-city images. We are planning further testing with bigger databases but for now we are lacking material for a more comprehensive benchmark.
Over the 250 images, our classifier identifies 81% of cityscapes correctly and indexes 88% of non-cityscapes correctly. Using C# code and without intensive optimizations, it takes about 1s on a 256 mb 1.7 GHz Pentium IV to process a 300x200 picture. Typical false rejections include cityscapes containing oddly shaped buildings and pictures with ill-defined edges. Noticeably pictures with "burnt" (i.e. completely white) skies show almost no edges between sky and buildings. This lack of horizontal edge sometimes caused the image to be discarded. False detections mostly include man-made structures such as isolated houses and walls.

4.2. Discussion on results

Our evaluation showed that segments detected using Hough transform are a viable low level descriptor and an improvement over previous ones such as the edge histogram. Considering the classifier itself, there are now many areas were results can be improved.

Poor performance has been reported on snowy landscapes for example, this is mainly due to the metrics used in the color segmentation process: indeed ignoring the "L" component in CIE-Lab color space poses a problem when there is a large amount of white and gray or black in the image as "a" and "b" components are undefined for black gray and white which means those very different "colors" will merge in a single region. We must also notice that currently (and not surprisingly) the system does not tell cityscapes from indoor images.

We also noticed that smaller, poor quality images may have some confuse edges which generate some parasite segments. This would require some prior filtering while processing edge.

4.3. Future work

The purpose of this work was to validate the use of Hough transform as a low level descriptor. We now aim to improve efficiency by the addition of other features such as a more complex color feature and also a texture feature. Although the usefulness of texture in cityscape characterization is doubtful (because of the amount variations in scale, and architecture), it would definitely prove useful for the purpose of detecting sky, grass or roads in order to help with the aforementioned indoor/outdoor detection problem. Yet we will try to keep dimensionality as low as possible.
The color segmentation algorithm also did not give full satisfaction regarding both computing time and the necessity to define thresholds for each picture. We therefore plan on designing a faster adaptive algorithm.
As far as Hough transform is concerned, poor quality of images sometimes generate thicker edges and thus parasite segment detections, the use of Canny Edge detector and further image filtering should thus be done.
At last the decision and learning process is also subject to improvement: we plan on introducing active learning procedures in order to allow for bigger training sets without increasing the data annotation time too dramatically. The increased size of the training set should allow getting rid of empirical formulae used to select and evaluate segment orientations.

5. REFERENCES

[1] A. Vailaya, M. Figueiredo, A. K. Jain, and H.-J. Zhang, "Image Classification for Content-Based Indexing" , *IEEE Transactions on Image Processing*, Palo Alto (CA), vol. 10, no. 1, pp 117-130, January, 2001.

[2] A. Vailaya, A. K. Jain and H.-J. Zhang, "On Image Classification: City Images vs. Landscapes", *Pattern Recognition*, vol. 31, pp 1921-1936, December 1998

[3] P.V.C Hough, "Methods and means for recognizing complex patterns", U.S. patent 3, 069,654 ; 1962

[4] D. Ben-Tzvi and M.B. Sandler, "A combinatorial Hough transform", *Pattern recognition letters*, Volume 11, n°3, pp 167-174; 1990

[5] L. Xu and E. Oja, "Randomized Hough transform (RHT): basic mechanisms, algorithms and computational complexities", *CVGIP: image understanding*, volume 57, n°2, pp 131-154, march 1993

[6] H. Kälviäinen, P. Hirvonen, L. Xu and E. Oja, "Probabilistic and non-probabilistic Hough transforms: Overview and comparisons", *Image and vision computing*, Volume 13, n°4, pp 239-252; 1995

[7] H. Kälviäinen, P. Hirvonen and E.Oja, "Houghtool – a software package for the use of Hough transform", *Pattern recognition letters*, Volume 17, n°8, pp 889-897; 1996

[8] V.F Leavers, "Which Hough transform", *CVGIP: image understanding*, volume 58, N°2, pp 250-264; 1993

[9] J.Illingworth and J. Kittlet, "A survey of Hough transform", *Computer Vision Graphics and Image Processing*, Volume 44, pp 87-116; 1988

[10] Mohsen Ardebilian Fard, "A contribution to content based video segmentation", PhD thesis directed by Pr. Chen L., Université de Technologie de Compiègne, december 2001

[11] Lutz Priese and Volker Rehrmann, "A Fast Hybrid Color Segmentation Method", *DAGM – Symposium*, vol 15, pp 297-304, 1993