CONTOUR DETECTION BY IMAGE ANALOGIES

Slimane Larabi[1], Neil M. Robertson[2]

(1) University of Sciences and Technology
Houari Boumediene,
Computer Science Department,
BP 32 El Alia, Algiers, Algeria
(2) School of Engineering and Physical Sciences,
Earl Mountbatten Building Gait 2,
Heriot-Watt University,
Edinburgh, EH14 4AS, UK

Abstract. In this paper we deal only contour detection based on image analogy principle which has been used in super resolution images, texture, curves synthesis and interactive editing. Human is able to hand drawn best outlines that may considered as benchmarks for contour detection and image segmentation algorithms. Our goal is to model this expertise and to pass on it at the computer for contour detection. Giving a reference image where outlines are drawn by human, we propose a method based on the learning of this expertise to locate outlines of a query image in the same way that is done for the reference. Experiments are conducted on different data sets and the obtained results are presented and discussed.

Key words: Image Analogies, Contour Detection, Outline Shape

1 Introduction

Contour detection is an important task in many computer vision applications such as object recognition, motion, medical image analysis, image enhancement and image compression. There is an huge number of methods in literature devoted to contour detection and many states of the art have been published giving a complete review of proposed techniques [21], [11], [18], [19].

Image analogies constitutes a natural means of specifying filters and image transformations [13] and we can simply supply an appropriate exemplar and say, in effect: "Make it look like this".

Assuming that the transformation between two images A and A' is "learned", image analogies is defined as a method of creating an image filter which allows to recover by analogy from any given different image B the image B' in the same way as A' is related to A [6] [13].

An advantage of image analogies is that they provide a very natural means of specifying image transformations. Rather than selecting from among myriad different filters and their settings, a user can simply supply an appropriate exemplar (along with a corresponding unfiltered source image) and say, in effect: "Make it look like this". Ideally, image analogies should make it possible to learn very complex and non-linear image filters [13].

Few works have been devoted for the use of image analogies in image processing. A method for supervised segmentation of medical images is proposed by Lackey and Colagrosso [15] applying directly the algorithm of Hertzmann [13]. The method is applied only to find by analogies the same colored regions in medical images as those processed by the expert.

Contrary to image processing, image analogies has been largely used in different applications such as super resolution [10], texture [12], [7], [9], [4], [2], [3], and curves synthesis [14], image colorization, texture transfer, image enhancement and artistic filters [17], [20].

However, human can locate easily the contours and results are mainly identical from one person. Our goal is to to model human expertise and to pass on it at the computer for contour detection.

Giving a reference image where outlines are drawn by human, we propose in this paper a method based on the learning of this expertise to locate outlines of a query image in the same way that is done for the reference. In section 2 we present a theoretical foundation as proof that contour may be detected using this technique.

Different data sets have been used to validate our approach. The obtained results are presented in section 3.

2 Contour Detection Using Image Analogies

2.1 Position of the problem

Human is able to draw on natural image good contour and therefore to produce best dataset that will serve as a benchmark for comparing different segmentation and boundary detection algorithms [1]. Our aim is then to locate contours such as human does it including low resolution images where objects have small sizes. For such images, the detection in image of outlines becomes a hard task and human must zoom in to hand drawn them accurately. Let L_A be the image of considered scene (see figure 1). We assume that outline shapes are manually located on L_A and highlighted with a specific color. We obtain then a new image H_A pertinent information (colored contours) that are clearly visible for users and useful for processing.

Given a query image L_B, the problem is then how it can be computed the image H_B which will contains outline shapes located and highlighted in the same way as those located in H_A.

2.2 Basic Principle of the Method

Before the describing of the proposed method, we discuss about the human expertise for contour detection. Given a low resolution image, we believe that

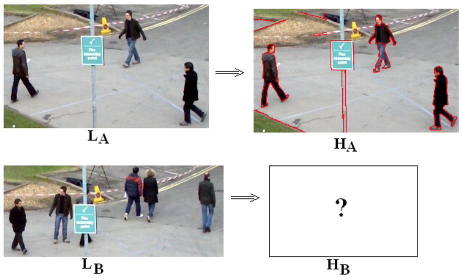

Fig. 1. Contour detection by analogy: Basic principle

human takes into account in this process two criteria for outline shapes locating:
- The first one is the neatness of the difference of gray level intensity (or color) between two neighboring sets of pixels.
- The second one is the knowledge of outline shape geometry inferred from context or some features such as outlines of dominant parts [8]. Indeed, during the process of outline drawing, human can't localize some parts of the outline due to the high similarity between pixels of background and object part but can avoid this difficulty using the prior knowledge. In this paper we deal only with the first criterion and we present our approach in order that computer locates pixels of contour in similar way to what is done by human.

In hand drawing contours task, we assume that a pixel p is considered belonging to the outline if the two following conditions are verified:
(1)- There is a brightness discontinuity at the considered pixel p for at least one directional line d_i $(i = 0..n-1)$ passing by p (see figure 2).
(2)- If the condition (1) above is verified for p, it is also verified for some p-neighboring along other directional lines (d_j).

Let (L_B) be a query image of low resolution. The key idea is to classify each pixel q of L_B using the knowledge that may be inferred from (L_A, H_A): Each pixel q will be classified as its best match p^* in L_A, this means that brightness of p^* and its neighboring are the best similar to those of q and its neighboring.

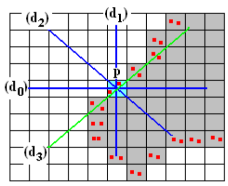

Fig. 2. Properties of contour pixel

Matching process between q and p pixels

Let $N(p)$, $N(q)$ be the $(m\times m)$ neighborhood of p and q in L_A, L_B. Our aim is to search $N(p^*)$ from all $N(p)$ in L_A having the same or equivalent circumstances of $N(q)$ pixels where p^* is classified as contour or shape pixel.

To selected the best match $N(p^*)$, we will take into account:
- The structure similarity between $N(p^*)$ and $N(q)$. This means that if p is a contour pixel, q will be also a contour pixel and the direction of the outline pixels in $N(p^*)$ and $N(p^*)$ must be very similar.
- The brightness similarity between $N(p^*)$ and $N(q)$ pixels.

For both $N(p)$ and $N(q)$, we will note the line of interest as the line which have the direction $d_i, i = 1..n$ and passes by the central pixel p or q. Depending on the chosen direction, each one of these lines may be an outline shape, background line or constituted by a set of shape and background pixels. In the comparison process, euclidian distance is used for similarity measure computation given by equation 1 where more importance is given to pixels of interest. To give more importance to pixels of the boundaries, the kernels $K^i(m \times m)$ characterized by high weight associated to pixels of the lines of interest are used.

$$S_m(q,p) = \sum_{u=-m/2}^{u=m/2} \sum_{v=-m/2}^{v=m/2} K^i(u,v) \times (Diff_{u,v})^2 \tag{1}$$

Where:
- $Diff_{u,v} = N(q)_{(i+u),(j+v)} - N(p)_{(k+u),(l+v)}$
- (i,j), (k,l) are the coordinates of the pixels q, p in the images L_B, L_A
- $N(p)_{i,j}, N(q)_{k,l}$ are intensities of pixels (i,j) and (k,l).

For a given q of L_B, the best match p^* of L_A will be chosen so as $S_m(q,p)$ is the minimum from all computed values. If q is a contour pixel, the minimum of $S_m(q,p)$ values is obtained when $N(p)$ has p as a contour pixel and all neighboring contour pixels must have the same direction as those of q. In addition, the difference of luminosity between corresponding pixels must be minimal.

2.3 Study of the validity of similarity measure

The proposed similarity measure must guaranty that any background or shape pixel q can't be classified as contour pixel and each outline pixel q will be correctly classified using the knowledge inferred from (L_A, H_A).

A theoretical study has be done and we present here only the main results. We investigated the cases where pixels are classified correctly using the two conditions cited in subsection 2.2 and the cases where this method fails.

The main result is that if the training pair of images (L_A, H_A) and the query one L_B are taken from the same scene, the location of contour pixels of L_B is done with success. However when L_B is from a different scene, the location of contour pixels can't be done without the loss of many candidates. This fact is due to following constraints that must be verified.

Let:
- G_A, S_A be the intensity average of $N(p)$ background and shape pixels (see

figure 3),
- G_B, S_B be the intensity average of $N(q)$ background and shape pixels,
- $d_B = S_B - G_B$, $d_A = S_A - G_A$, $d_G = G_B - G_A$.

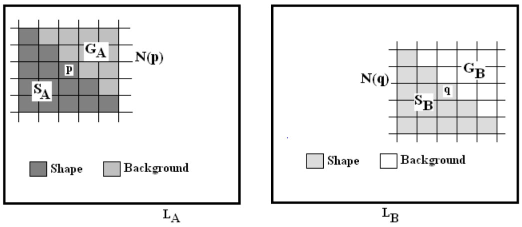

Fig. 3. Basic principle of p, q matching

We demonstrated that:
- if q is a shape or background pixel, it can't be classified as a contour pixel.
- if q is a contour pixel, it will be classified correctly if the constraint (2) is verified (see figure 4).
($d_A > 0$ and $2d_G < d_A < 2d_B + 2d_G$) or ($2d_B + 2d_G < d_A < 0$) (2) The details of this proof aren't included in this paper because the limit of the number of pages.

This means that the contour pixel q of a query image will be located if there are in the training images (L_A, H_A) hand drawn contour pixels so as $N(p)$ and $N(q)$ verify the constraint (2). Otherwise, it will be classified as shape or background pixel.

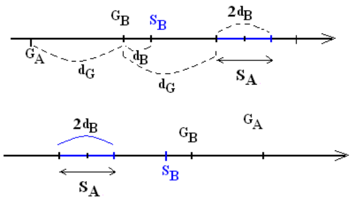

Fig. 4. Constraints on the values of S_A depending on G_A, G_B, S_B

To avoid this constraints, we propose the use a set of artificial patterns (L_A, H_A) instead of real images with hand drawn contours (see figure 5). The values of (G_A, S_A) are chosen so as for any query pixel q, the values of (G_B, S_B) of associated $N(q)$ verifies the constraint (2) for at less one pattern.

This idea is due to the fact that is we have a pattern (L_A, H_A) as illustrated by figure 6, the values of any $S_B > G_B$ where $G_B = (S_A - \varepsilon - G_A)/2$ will verify the constraint (2), because $S_A \in]2d_G 2d_G + 2d_B]$ and then the pixel q of

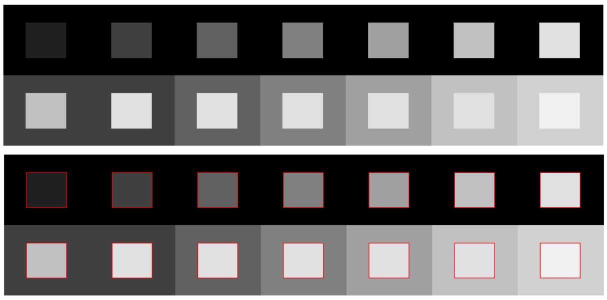

Fig. 5. Artificial patterns (L_A, H_A)

considered $N(q)$ will be classified correctly. The translation of S_A with the value of $2\delta l$, where δl is the minimal difference intensity between two regions, then the value of G_B will be translated by δl. In other side, for a fixed values of G_A, S_A, the value of G_B may decrease from $(S_A - \varepsilon - G_A)/2$ to $(S_A - \varepsilon - G_A)/2) - \delta l$ because the constraint (2) will be verified. For $\delta l = 16$, 14 patterns are generated and allow to located all contour pixels of any query image.

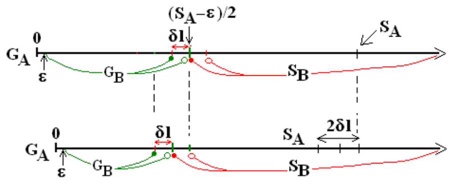

Fig. 6. Basic principle for satisfying all (G_B, S_B) values

3 Results

Experiments have been conducted on different datasets of real images: Caviar [5], PETS [16].

To detect the outline of a query image of a given scene, it is necessary to have a pair of training images (L_A, H_A) where L_A is the image of the scene and H_A is the same image which contain hand drawn contours.

Table 1 illustrates images L_A, H_A from Caviar data set where hand drawn outline shapes are highlighted with red color. The outlines located for some images of the same video are illustrated by the same table. We can see that in the query images only some outlines are located verifying the given constraint.

Despite this limit, hand drawn contours as reference may be used for example for the tracking of a moving object. In this case, it is sufficient to locate the outline of the subject in different conditions of lighting and the outlines of the

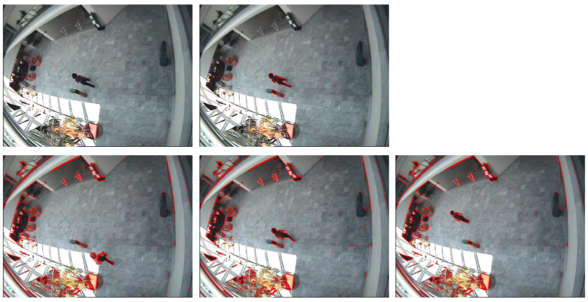

Table 1. The L_A, H_A images and Contours located for some frames of Caviar video

background aren't needed. Table 2 illustrates some frames of Caviar video where human is detected in each one. The hand drawn image is taken from the same video.

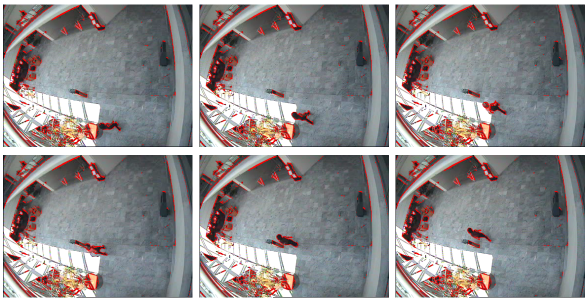

Table 2. Contours located for some frames of Caviar video

Each one of the pairs artificial patterns enables us to detect a specific level of contour depending on the intensities of the neighboring regions to the border. There are then 14 levels which are defined by the pairs (L_A, H_A). Table 3 illustrates contours located on the same frame of CAVIAR video using some patterns of the set of 14 patterns. We can see that the outline is moving through one region.

For the BSD500 dataset, we computed the Precision and Recall and we obtained similar results as Arbelaez et al [1]. Table 4 illustrates some obtained results. For high Recall values, our Precision is better and the difference reaches 20%. This implies that the number of false candidates are less than of Arbelaez et al. However for low Recall values, our Precision values are near from the values of Arbelaez et al [1], the difference is around 3%. This means that the number of missing good candidates are almost the same for both methods.

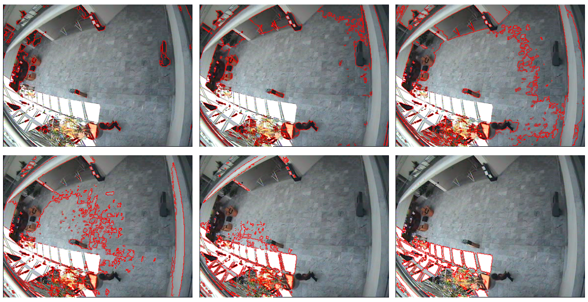

Table 3. Contours computed using respectively the patterns P_3^1, P_5^1, P_7^1, P_9^1, P_{11}^1, P_{14}^1

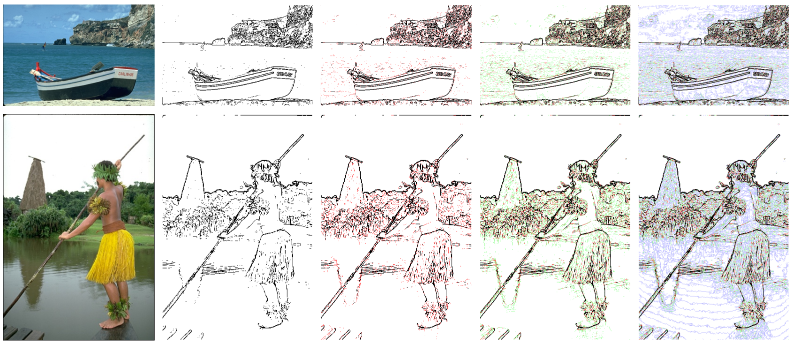

Table 4. Some BSD dataset results. From left to right: Original image, contours with different values of δl

We repeated the same experiments for the dataset of Caviar and PETS. High values of Precision and Recall values have been obtained and exceeds those of Arbelaez et al [1] using BSD dataset and the difference reaches respectively 27% and 10%.

4 Conclusion

We proposed in this paper a new method for contour detection based on image analogies. In the first part, we presented a theoretical foundation for the use of hand-drawn contours as reference images to be used subsequently for the detection of new contour pixels by analogy in the query image. We found that only pixels that have the same conditions as those of the reference image may be located which is perhaps to be expected in such a data driven technique. This implies that numerous reference images are needed to locate all possible new contour pixels which implies in the hard and time-consuming task of hand drawing reference contours, and thus increasing of the algorithm complexity.

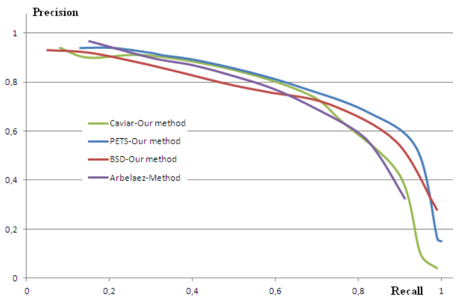

Fig. 7. Obtained ratios (Recall, Precision) for BSD dataset

To avoid this constraint and to locate all contour pixels whatever the image query happens to be, we proposed a set of 14 or 28 artificial pairs of patterns as reference images of low size and containing the required information to locate contours of different levels of resolution where levels are related to the difference of intensity between neighboring regions.

The proposed method has been applied to different types of images: the "natural" BSD dataset, indoor scenes (CAVIAR) and outdoor scenes (PETS2009). Compared to the reference images, our method demonstrates a very good recall, precision and finds all visible contours of gray level images. In addition, outline shapes of low resolution images such as those in CAVIAR and PETS are well located where the width of human as example is around 10 pixels.

5 REFERENCES

References

1. P. Arbelaez, M. Maire, C. Fowlkes and J. Malik, Contour Detection and Hierarchical Image Segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 33(5), pp. 898-916, May 2011
2. M. Ashikhmin, Synthesizing Natural Textures, Proceedings of 2001 ACM Symposium on Interactive 3D Graphics, I3D'2001, ACM Press, pp. 217-226, 2001
3. M. Ashikhmin, Fast texture transfer, IEEE Computer Graphics and Applications, Vol. 23(4), pp. 38-43, 2003
4. P. Bhat, S. Ingram and G. Turk, Geometric texture synthesis by example, Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing, SGP '04, pp. 41-44, 2004, Nice, France
5. Caviar, EC Funded CAVIAR project/IST 2001 37540, Benchmark Data, http://homepages.inf.ed.ac.uk/rbf/CAVIAR/, 2001
6. L. Cheng, S. Vishwanathan and X. Zhang, Consistent image analogies using semi-supervised learning, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Anchorage, AK, 2008
7. J. S. De Bonet, Multiresolution sampling procedure for analysis and synthesis of texture images, Proceedings of the 24th annual conference on Computer graphics and interactive techniques, SIGGRAPH '97, pp. 361–368, 1997

8. J. DeWinter and J. Wagemans, Segmentation of object outlines into parts: A large-scale integrative study, Cognition, Vol. 99(3), pp. 275 - 325, 2006
9. A. A. Efros and T. K. Leung, Texture Synthesis by Non-Parametric Sampling, IEEE International Conference on Computer Vision, Vol. 2, 1999
10. W. T. Freeman, E. C. Pasztor, O. T. Carmichael, International Journal of Computer Vision 40(1), 2000
11. J. Freixenet, X. Munoz, D. Raba, J. Marti and X. Cufi, Yet Another Survey on Image Segmentation: Region and Boundary, Information Integration, ECCV (3), pp. 408-422, 2002
12. D. J. Heeger and J. R. Bergen, Pyramid-based texture analysis/synthesis, Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, SIGGRAPH '95, pp. 229-238, New York, NY, USA, 1995
13. A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless and D. H. Salesin, Image analogies, I n Proceedings of the 28th annual ACM conference on Computer graphics and interactive techniques, SIGGRAPH '01, pp. 327-340,New York, NY, USA
14. A. Hertzmann, N. Oliver, B. Curless, S. M. Seitz, Curve Analogies, EGRW '02 Proceedings of the 13th Eurographics workshop on Rendering Switzerland, Switzerland, 2002
15. J. B. Lackey and M. D. Colagrosso, Supervised Segmentation of Visible Human Data with Image Analogies, IC-AI, pp 843-847, 2004
16. Pets 2009, PETS 2009 Benchmark Data, University of Reading, UK, http://www.cvg.rdg.ac.uk/PETS2009/a.html, 2009
17. D. Sykora, J. Burianek and J. Zara, Unsupervised colorization of black-and-white cartoons, Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering, NPAR '04, pp. 121-127, 2004, Annecy, France
18. J. S. Suri, K. Liu, S. Singh, S. N. Laxminarayan, X. Zeng and L. Reden, Shape recovery algorithms using level sets in 2-D/3-D medical imagery: a state-of-the-art review, IEEE Transactions on Information Technology in Biomedicine, Vol. 6(1), pp. 8-28, 2002
19. G. Papari and N. Petkov, Edge and line oriented contour detection: State of the art, Image Vision Computing Journal, Vol. 29(2-3), pp. 79-103, 2011
20. G. Wang, T. Wong and P. Heng, Deringing cartoons by image analogies, ACM Trans. Graph, Vol. 25(20), pp 1360-1379, October 2006
21. D. Ziou and S. Tabbone, Edge Detection Techniques - An Overview, International Journal of Pattern Recognition and Artificial Intelligence, 1998