Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 2, p. 47–48.
http://urlib.net/dpi.inpe.br/ismm@80/2007/06.15.18.57

Attribute sub-tree matching algorithm

André Körbes[*,1],
Alexandre Gonçalves Silva[1,2] and
Roberto de Alencar Lotufo[3]

[1] *Universidade do Estado de Santa Catarina, Brazil*

[2] *Universidade Estadual de Campinas, Brazil*
`alexandre@joinville.udesc.br`

[3] *Universidade Estadual de Campinas, Brazil*
`lotufo@unicamp.br`

1. Introduction

Tracking objects on image sequences is a problem addressed by various methods, through different approaches, using specialized operators to isolate movement [2], combined morphological operators for background removal [1] and foreground region detection [5], and graph-based algorithms, with particular detail to tree-based ones, where the trees are built by hierarquical segmentation tecniques [3]. This paper studies methods for approached matching of trees from grayscale images (automatic segmentation of different frames) and sub-tree obtained from objects (manual segmentation of the first frame) based on their structures and attributes that must be flexible and generated efficiently. A new algorithm is proposed using the max-tree [4] consisting in simple comparisons between vertices and ranking of the best candidate solutions.

The paper is organized as follows: Section 2 presents the algorithm in details, Section 3 shows results from a prototype applied on a testing video from project CAVIAR[1], and Section 4 discusses some of the future work to be done on the algorithm, such as the inclusion of more attributes, besides temporal and spatial coherence.

2. Algorithm project

The proposed algorithm works by matching trees, and some notation must be introduced first: T is the tree where the sub-tree S is searched, and as non-oriented graphs are denoted $T(V_T, E_T)$ and $S(V_S, E_S)$ where V and E are the set of vertices and of edges, respectively. E is a set of pairs of elements from V, for example: $V = \{v_1, v_2, v_3, v_4, \dots\}$ and $E = \{(v_1, v_2), (v_1, v_3), (v_2, v_4), \dots\}$. The definition of degree in this paper is related to the number of child of a vertex. The algorithm is divided into three steps, presented in the following sections.

2.1 Selection

The first step of the algorithm is focused on selecting matches for vertices of S on T. This is performed as an attempt to map the trees on a vertex level through basic attributes available, like area and degree. In order to have a useful mapping, all vertices from S must be compared against all vertices from T, and the following condition is required:

$$|A(v_t) - A(v_s)| \leq T_A \quad \text{and} \quad |G(v_t) - G(v_s)| \leq T_G$$

A is the area and G is the degree of the vertices, whose absolute difference must be less than a tolerance T_A and T_G respectively. This step creates a list for each vertex on V_T with its mapping on V_S, named as $M(v_t) = \{m_1^t, m_2^t, m_3^t, \dots\}$. After all vertices are compared, each one can have multiple elements on its list due to the tolerance applied, and multiple candidate sub-trees for S might exist. Therefore, a classification method is necessary to tell what is the best match for each vertex (second step).

2.2 Classification

The classification step generates values for each vertex on $M(v_t)$ list to identify the best match using a value $F(v_t)$ to indicate how it fits as a candidate solution. This is done only for vertices of V_T that have any matches located.

The method of classification uses a sum of the value of an attribute $P_i(m^t)$ multiplied by its weigth W_i that gives the importance of each one. Equation 1 shows the general formulation for i attributes resulting on value Q for each vertex. If Q is high, the match is good. Equation 2 selects the greatest value amongst all Q generated upon M.

$$Q(m^t) = \sum_i P_i(m^t) \times W_i \qquad (1)$$

$$F(v_t) = max(\{Q(m_1^t), Q(m_2^t), \dots\}) \qquad (2)$$

Values used for P_i are under development to choose the better ones, but the absolute difference between areas and degrees are very likely, as well as the inverse of the distance to corresponding parent of m_i^t on S mapped on T, as shown in Figure 1(a) by the arrow from vertex T_5 to T_1. With a classification rate on all mapped vertices, another step to identify the candidate sub-trees is necessary.

[*] `andre.korbes@gmail.com`
[1] http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

Proceedings of the 8th International Symposium on Mathematical Morphology,
Rio de Janeiro, Brazil, Oct. 10–13, 2007, MCT/INPE, v. 2, p. 47–48.
http://urlib.net/dpi.inpe.br/ismm@80/2007/06.15.18.57

2.3 Identification

The identification is based on grouping vertices mapped on first step and then evaluated on the second into sub-trees, in a way to find candidate solutions for the output of the algorithm, and then choosing one to be the final result. This step is necessary to deal with situations where it is not obvious which sub-tree is the solution, as in Figure 1.

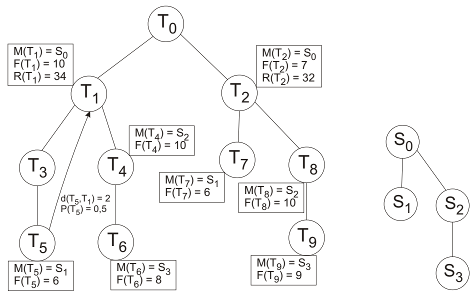

Figure 1. Tree matching. (a) Tree T with values of M, F and R. (b) Sub-tree S.

This step works with the values produced previously, through a sum of all its children values of F for every vertex of T that is mapped and does not have any direct or indirect parent mapped. This value might be interpreted as an indicator of the quality of the sub-tree as a solution, and is denoted R. On Figure 1(a) it is calculated only for T_1 and T_2. Before summing R it is interesting to threshold the values of F on a low value to eliminate those vertices that have a weak relation and are not good for the final result.

On trees where more than one vertex is attributed with R there is a need to rank those approximate sub-trees through its properties, as height - which is desirable to be minimal, the rate of vertices not mapped by total, the rate of mapped vertices of T by the vertices of S, among other possibilities. As in step 2, each attribute is weighted to indicate its relevance. Figure 1(a) shows that the sub-tree rooted at T_2 has a lower value of R than the one rooted at T_1, but its rates are higher, what indicates a better solution.

3. Results

Preliminary results are shown on Figure 2, tracking the manually segmented object in first frame. The sub-tree identified by the algorithm is enhanced with a contour, and better results are noticeable on areas with higher contrast. Since the algorithm does not have any training steps, results are obtained from the first frame.

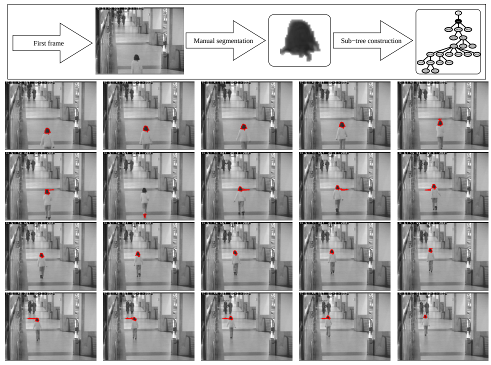

Figure 2. Template and hair segmentation by frame.

4. Conclusions

This paper presented a method for approache matching of trees with attributes based on simple comparisons between vertices. The results presented on Section 3 are preliminary, as well as parameters and weights must be more deeply studied and maybe automatized for some images. Considering that no aspect of subtraction, optical flow or motion estimation is implemented, current results are interesting and fast (around 0.5 seconds by frame of 384×288 pixels). The use of shape descriptors has been studied in order to improve segmentation and selection of vertices, for instance, by differentiating lines and circles that have the same area.

References

[1] Arnaldo Câmara Lara and Roberto Hirata Jr., *Motion Segmentation using Mathematical Morphology*, XIX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'06) (2006).

[2] Philippe Salembier, Albert Oliveras, and Luis Garrido, *Motion Connected Operators for Image Sequences*, European Signal Processing Conference (EUSIPCO) (1996).

[3] Julian Mattes, Mathieu Richard, and Jacques Demongeot, *Tree Representation for Image Matching and Object Recognition*, Proceedings of the 8th International Conference on Discrete Geometry for Computer Imagery (1999).

[4] Luis Garrido, *Hierarchical Region Based Processing of Images and Video Sequences: Application to Filtering, Segmentation and Information Retrieval*, Universitat Politècnica de Catalunya, Barcelona, 2002.

[5] Kairoek Choeychuen, Pinit Kumhom, and Kosin Chamnongthai, *An Efficient Implementation of the Nearest Neighbor Based Visual Objects Tracking*, Proceedings of the International Symposium on Intelligent Signal Processing and Communications (ISPACS '06) (2006).