Autonomy Oriented Computing Applied in Color-based Image Segmentation

Slobodan Ribarić and Branko Samaržija

Faculty of Electrical Engineering and Computing, University of Zagreb
Unska 3, 10000 Zagreb, Croatia
E-mail:{slobodan.ribaric, branko.samarzija}@fer.hr

Abstract – This paper presents an Autonomy Oriented Computing (AOC) approach to color image segmentation. Basic elements in AOC systems are an environment and autonomous entities. Environment, in our case, consists of the three layer 2D lattice which contains color images in RGB and HSI color spaces as well as storage cells through which entities interchange information. Environment serves as the place where autonomous entities roam and operate. Entities exhibit a number of reactive behaviors such as diffusing, self-reproduction, waiting or dying. Goal of each entity is to locate and label image pixels whose color satisfies skin-like criteria. Entities which are successful in locating such pixels have an opportunity to produce offspring entities and place them within their neighborhood, where they are most likely to find other skin color-like pixels. The entities that don't succeed in locating any skin color like pixel vanish from the system when their ages exceed their lifespan. Results of all entity families are combined into regions. Experiments, based on a program simulator, were run over set of frontal face images from the XM2VTS database.

1 INTRODUCTION

Autonomy Oriented Computing (AOC) is a relatively new bottom-up paradigm for problem solving and complex system modeling where special attention is paid to the role of self-organization and interaction among the autonomous entities [1]. An AOC system is formally described with the 3-tuple: AOC = $(\{e_1, e_2, ..., e_N\}, \mathbf{E}, \mathbf{\Phi})$, where $\{e_1, e_2, ..., e_N\}$ is a group of N autonomous entities which are the smallest and simplest building blocks of the system, **E** is an environment in which entities reside, and **Φ** is a system objective function, which is usually a nonlinear function of entity states. The main characteristics of the entities in the AOC system are [1]:

1) **Autonomy**: System entities are (rational and/or reactive) individuals that act independently.
2) **Emergency**: They exhibit complex behaviors that are not present or predefined in the behavior of the autonomous entities within complex adaptive systems.
3) **Adaptive**: Entities often change their behaviors in response to changes in the environment in which they are situated.
4) **Self-organization**: Entities are able to organize themselves to achieve the above behaviors.

All entities in an AOC system are characterized with their internal states, evaluation functions, goals, primitive behaviors and behavioral rules. Entities can also exhibit three types of complex behavior: emergent behavior, purposeful behavior and emergent purposeful behavior. Different types of complex behavior can emerge from interaction among entities or groups of entities directly, or through the environment.

AOC systems applied to image processing is a newly explored area of research that studies the emergent behaviors in a lattice where entities react to the digital image environment according to a set of behavioral rules [2]. The AOC theory is mainly inspired by multi-agent system (MAS) theory and distributed behavior-based agent concepts. Here follows related works based on the MAS and behavior-based agent concepts for image processing and feature extraction: edge detection and image feature tracking [2]; gray-scale image region segmentation [3]; face detection and localization in color spaces [4, 5].

In this paper we have described a model of AOC-based system for color image segmentation. To test the proposed model we defined system objective function **Φ** in such a way that it is able to segment skin-color like image areas.

2 SYSTEM ORGANIZATION

The proposed system is two-level AOC-based and dedicated for low level image processing. The first, lower level called pixel-level, is modeled according to AOC paradigm. It is consisting of autonomous entities with primitive behavior and ability to self–reproduce, diffuse and interact with each other in 2-D lattice representation of image environment. At the second level, called algorithmic level, information obtained from entity or groups of entities are gathered and processed in order to generate final results i.e. image regions. Pixel-level is problem independent and can be used for wide range of image processing tasks, while the algorithmic level is problem specific.

2.1 Pixel level

The pixel-level of our AOC = $(\{e_1, e_2, ..., e_N\}, \mathbf{E}, \mathbf{\Phi})$, model is, in general, characterized by entities' states and goals, and provided with their evaluation function, primitive behavior and behavioral rules, environment and system objective function as follows:

a) The state of an entity is defined by the following attributes: internal state (*active*, *sleep* or *dead*), age, position, lifespan, family descriptor. First three attributes are dynamically changed, while the last two are static (predefined constant).
b) Goals of an entity are to explore an image and find the image features of interest.
c) Evaluation function is task-oriented function and it is used to judge if a pixel at which entity is positioned belongs to image feature of interest.
d) Entity behavior consists of: self-reproduction, pixel marking, sleeping, diffusion and dying.
e) Behavioral rules determine entity state transitions and select group of behaviors based on evaluation function and state of environment.

Environment **E** plays three main roles [1]:

1) It is domain in which autonomous entities roam,
2) Environment acts as a notice board where entities post and read their shareable information,
3) Environment keeps a central clock that helps synchronize the behaviors of all autonomous entities, if necessary.

The pixel-level domain, in which entities roam, is two-dimensional multi-layer lattice, with finite number of cells. Each cell corresponds to a place that entity can visit and/or stay at.

The notice board is represented as one layer of two-dimensional lattice of storage cells. Entities leave the certain patterns called markers in the storage cells. Each entity has ability to sense markers within its 8-neighborhood of the notice board.

Environment that keeps a central clock and synchronizes the behaviors of entities is required in parallel implementation of the system. In simulation this part of environment is implicitly incorporated in the system supervisor programs.

Φ is a system objective function state- or process-oriented [1], which is usually a nonlinear function which maps a subset of the state of entities into the set of real numbers or integers.

2.2 Algorithmic level

The algorithmic level, which is problem specific, gathers information about entities or groups of entities and process markers from the notice board. Based on information such as current position of all entities, their states and values of markers, the algorithmic level has a global overview of current problem space and it determines the system's task-oriented procedures.

3 AOC BASED IMAGE SEGMENTATION

Image segmentation is process of partitioning an image into regions. All pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity or texture. The proposed system is specialized for image segmentation where pixels are grouped into regions according to their color property.

3.1 Image segmentation

At pixel-level of the AOC-based system, color-based image segmentation can be obtained by modeling entity's evaluation function to return logical *true* if evaluated pixel color is in predetermined range of colors or logical *false* if not. Entities leave indexes of their families as markers at the notice board for every pixel which have received *true* form the evaluation function. At the algorithmic level of the system, family index markers are grouped into regions. The system objective function serves as a global measurement for the performance of the system and it leads system towards state in which whole image is segmented. It maps states of all entities into the number of currently active entities. System objective function returns numerical value zero only if all entities become inactive. This state is considered as the end of process of image segmentation.

Since our AOC-based system at pixel-level is specialized for image segmentation task based on fusion of intensity of primary spectral components of red, green and blue (RGB) with information obtained form HSI (Hue, Saturation and Intensity) color model, each entity is placed in the environment **E** which is characterized as follows:

The environment **E** is 3-layer 2D lattice with $W \times H$ cells**,** where W and H correspond to the dimensions of an image (Fig. 1).

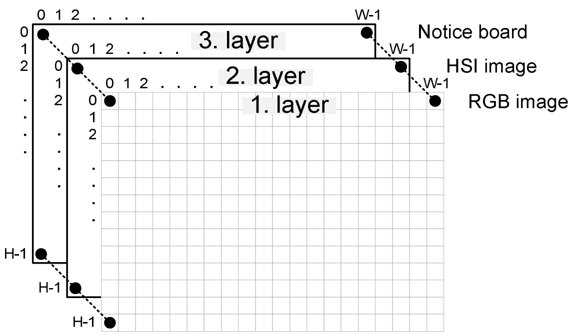


Figure 1: Pixel level environment E

First layer contains original color image in RGB color space. Second layer is problem specific and contains corresponding color image in HSI - color model. Use of both layers enables fusion when target range of colors (i.e. color subspace) is specified. Third layer is a notice board and it has same dimensions as first two layers. It is important to notice that if entity is positioned at cell[i, j] of the first layer, it is also positioned at the cell[i, j] of the second layer and has ability to leave marker at cell[i, j] of the third layer, where i = 0, 1, ..., W-1 and j = 0, 1, ... H-1. Each entity is described by 3-tuple (S, F, B), where S is the current state of entity, F is its evaluation function, and B is its primitive behavior. The current state of entity is defined as 5-tuple as follows:

$$S = (internal\_state, a, p, Ls, Fi), \quad (1)$$

where:

internal_state – define entity's internal state, which can be*: active* it is initial state in which entity stays during its lifecycle while searching for pixels with predefined range of colors; *sleep* in which entity goes when it found unmarked pixel which value is in predefined range of colors; *dead* in which entity goes when its age exceeds its lifespan or entity isn't utilizable for further image processing.

a – entity's age. Initial ages of all entities are set to zero. At the beginning of every central clock cycle an entity selects one type of behavior. Entity's age is incremented during performing each type of behavior (i.e. *Diffusing, Waiting, Self-reproduction, Dying*). That means, for example, if entity diffuses from cell to the neighboring cell its age is incremented. When an entity enters into *sleep* state its age is frozen.

p – entity's position in the environment **E**. Entity's position is represented by cell coordinates [i, j] in **E**.

Ls – entity's lifespan. When entity's age reach *Ls* as a threshold value, an entity changes its state to *dead*.

Fi – entity's family index. It is an integer value and it is unique for each entity initially placed in **E**. Every entity has ability to produce offspring by applying *Self-reproduction* behavior. All offspring belong to same family will share the same family index. A family index is used as a marker and it is written to the notice board when target value of the pixel is found.

Evaluation function F performed by each entity is used for assessment of conditions in environment **E**. Entity performs evaluation of pixel values in the first end second layer of the environment **E**. Evaluation function F can be described as:

$$\mathrm{F}: \mathrm{D}_{\mathrm{ES}} \rightarrow \{0,1\}, \tag{2}$$

where

$\mathrm{D}_{\mathrm{ES}} = Red \times Green \times Blue \times Hue \times Saturation \times Intensity$

is a space of primary spectral components red, green and blue stored at first layer and values of hue, saturation and intensity stored at second layer of E.

Function F at a cell [i, j] is modeled as:

$$\begin{aligned}&\mathrm{F}(Red, Green, Blue, Hue, Saturation, Intensity)_{[\mathrm{i},\mathrm{j}]} = \\ &\mathrm{C}_{[\mathrm{i},\mathrm{j}]}(Red) \wedge \mathrm{C}_{[\mathrm{i},\mathrm{j}]}(Green) \wedge \mathrm{C}_{[\mathrm{i},\mathrm{j}]}(Blue) \wedge \\ &\mathrm{C}_{[\mathrm{i},\mathrm{j}]}(Hue) \wedge \mathrm{C}_{[\mathrm{i},\mathrm{j}]}(Saturation) \wedge \mathrm{C}_{[\mathrm{i},\mathrm{j}]}(Intensity)\end{aligned} \tag{3}$$

where $\wedge$ denotes logical AND, and

$$\mathrm{C}(argument)_{[\mathrm{i},\mathrm{j}]} = \begin{cases} 1 & \text{if } min\_argument \le argument_{[i,j]} \le max\_argument \\ 0 & otherwise \end{cases} \tag{4}$$

min_argument and *max_argument* are predefined minimum and maximum color values of a pixel at cell[i, j] and $argument \in$ *{Red, Green, Blue, Hue, Saturation, Intensity}*.

During the course of evolution, each of the entities in the environment **E** will exhibit several types of behavior: *Diffusing*, *Waiting*, *Self-reproduction* and *Dying*.

Each entity can perform one of the following behaviors depending on its stimuli present in the environment and internal state:

Diffusing – entity makes step to the next cell if an evaluation function at the cell[i, j] returns false. Each entity has ability to sense its 8-neighborhood of cells and randomly move to any free location. The free location is the cell[i, j] with no entity.

Waiting – if all neighboring cells are occupied by other entities, the entity will stay at the current cell until one of neighboring cells becomes free. Entity's age will be increased by 1 for each cycle of the central clock.

Self-reproduction – if entity receives true from the evaluation function it will leave a mark (entity family index) at the current cell of the notice board and reproduce from zero to eight offspring entities within its 8-neighborhood depending on a number of free cells. If there are no free cells then no offspring will be created. After self-reproduction entity changes its state to sleep and becomes a sleeping entity.

Dying – entities whose ages exceed given maximal age will become inactive and vanish from the system.

The system objective function $\Phi^{(cycle)}$ for image segmentation is defined as a state-oriented function and it maps set of states of all entities at the given cycle of the central clock into set of integer values Z. It is given as follows:

$$\Phi^{(\mathrm{cycle})}: internal\_states^{\infty} \rightarrow \mathrm{Z} \ , \tag{5}$$

where $internal\_states^{\infty}$ denotes a multiset of $internal\_states = \{active, sleep, awake, dead\}$, and Z is a set of integers. Mapping $\Phi^{(cycle)}$ returns number of entities in an *active* state at the given central clock cycle (*cycle=1, 2, 3, ... ,*); i.e. $z \in Z$ is #(*active*, $internal\_states^{\infty}$). The #(*x*, A) function defines the number of occurrences of an element *x* in a multiset A. Goal value of the function $\Phi^{(cycle)}$ is $z = 0$, i.e. #(*active*, $internal\_states^{\infty}$) = 0. This value defines end of any activity at the pixel-level of the AOC-based system.

The next phase of color image segmentation is performed at the algorithmic level of the AOC-based system, as follows: According to information contained in the notice board of the environment **E** at the pixel - level, cells marked with the same family index form one part of the region. Through the iterative merging process, parts of the regions are merged according to the neighborhood criterion. The neighborhood criterion for merging two parts of the regions is based on the size of the border between them. If border between region parts is larger than some predefined threshold, parts are merged in the one bigger part. The threshold value is relative and it is determined as a percentage of the smaller perimeter of two neighboring region parts. Region merging is completed when there are no two region parts that satisfy merging criterion.

As result we have segmented color image where all image regions are enumerated and can be extracted or analyzed separately. Regions which are smaller than predefined threshold (for example 3% of the size of the biggest region) are deleted; i.e. corresponding entities change their state from *sleep* to *dead*. It should be noticed that pixel-level of AOC-based system still contains sleeping entities. Sleeping entities can be awaken and employed for additional image processing.

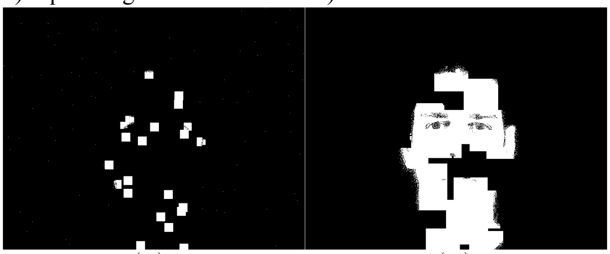
4 EXPERIMENTS

The AOC-based system has been implemented as program simulator and it was tested on XM2VTS database [6] containing frontal face images, where color-skin like areas were segmentation target. To detect such regions, the evaluation function was modeled using the following parameters [4]:

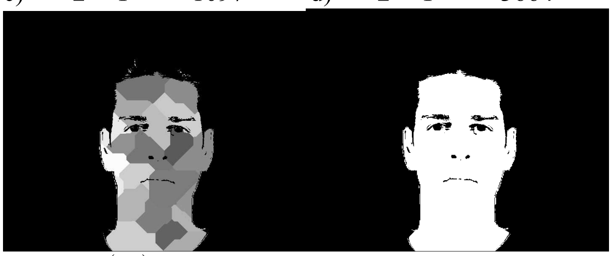
$$\text{Hue} \in [50, 340], \text{Saturation} \in [0.2, 0.7], \text{Red} \in [50, 255] \quad (6)$$

Parameters for Intensity, Blue and Green were not used. Fig.2. illustrates different phases of image segmentation.

a) Input image: 720 x 576 b) $z = \Phi^{(0)} = 100 \quad Ls = 50$



c) $z = \Phi^{(10)} = 1097$ d) $z = \Phi^{(40)} = 3664$



e) $z = \Phi^{(147)} = 0$ f) Result
#entity families = 35 #regions = 1
#sleeping entities = 85066 #sleeping entities = 84732

Figure 2: An example of image region segmentation

Fig. 2.a) represents input image (RGB; resolution 720x576) from XM2VTS database. In Fig. 2.b) the initial state of environment **E** is shown: one hundred entities are distributed randomly in the E. The entity lifespan was set to 50 clock cycles. Fig. 2.c) shows state of the notice board at the central clock cycle = 10. The white regions represent entity families consisting of entities which evaluation function has returned 1 (cells with the skin color – like pixels are found). Fig. 2.d) represents state of the notice board at cycle = 40 with 3664 active entities. Fig. 2.e) depicts the final result of image segmentation at the pixel-level obtained in cycle = 147; the number of active entities is zero and number of entity families is 35. The final result of image segmentation after merging process and after elimination of small regions at the algorithmic level is shown in Fig. 2.f).

Experiments were run over 600 images, containing 150 persons, 4 images for each one. A facial region was successfully detected and segmented on each image. Given fusion of color models has shown to be very robust for different race skin colors detection. During the experiments an initial number of entities was set to be between 100 and 1000 (0.024% to 0.24% of the number of total image pixels), and the entity life span was varied between 30 and 100 clock cycles. In all experiments entities were initially distributed in the random manner. Average simulator processing time for image segmentation was 98 ms; for the PC based on dual core processor running at 2.8 GHz. Pixel level of AOC- based system took about 90 ms and algorithmic level 8 ms. It is important to notice that performance measure of the pixel-level is number of the environment **E** central clock cycles. Average number of central clock cycles for image segmentation was 150.

5 CONCLUSION

This paper describes an approach to build and utilize an AOC-based system for the image segmentation problem. Modeling AOC environment and placing an image into it, enables construction of various image processing algorithms by defining entities' behavior. For the color image segmentation primitive behaviors such as diffuse and self-reproduce accompanied with the ability to evaluate and label environment cells have shown to be sufficient for detection of skin color like region parts. In future, based on results of simulation, we can expect that AOC based hardware architecture which enables entities in the system to act parallel will provide at least the same quality of segmentation as "classical" approaches to segmentation, but with significant speedup. For the comparison purposes we have used the algorithm for the skin color image segmentation based on the connected components labeling [7] which needs about 20 ms for the color image segmentation. Hardware AOC-based architecture can produce same results for about 8 ms according to our simulations.

REFERENCES

[1] J. Liu, X. Jin, K. C. Tsui, *Autonomy Oriented Computing*, 1st edition, Springer, New York, 2004.

[2] J. Liu, Y.Y.Tang, Y.C.Cao, "An evolutionary autonomous agents approach to image feature extraction", *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 2, pp. 141 - 158, Jul 1997.

[3] J. Liu and Y.Y. Tang, "Adaptive Image Segmentation With Distributed Behavior-Based Agents", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 21, no. 6, pp. 544 - 551, June 1999.

[4] D. Kipcic and S. Ribaric, "A multi-agent-based approach to face detection and localization", *27th International Conference on Information Technology Interfaces*, pp. 377 - 382, June 20-23, 2005.

[5] Y. Wang and B. Yuan, "Fast method for face location and tracking by distributed behavior-based agents", *IEE Proc. on Vision, Image and Signal Processing*, vol. 149, no. 3, pp.173 - 178, June 2002.

[6] The extended M2VTS database: http://www.ee.surrey.ac.uk/ CVSSP/ xm2vtsdb/ (1. July 2008)

[7] R. C. Jain , R. Kasturi, B. G. Schunck, *Machine Vision*, 1st edition, McGraw-Hill, New York, 1995.