AGI engineering

Subscribe

AGI: VISUAL ATTENTION

AN ENGINEERING APPROACH TO BUILDING A SCENE DESCRIPTION

Mykola Rabchevskiy
Jul 2

Attention management is one of the critical aspects of decision making: NUTS AND BOLTS OF THE DECISION MAKING

The most informative fragments of a scene are ***contours*** as noted in
ARTIFICIAL GENERAL VISION
Accordingly, the described implementation of visual attention management classifies fragments of the visual scene using two criteria, the ***degree of "saturation" of the fragment with contours*** and ***dynamism***.

The scene fragments (for each of these criteria are calculated) are ***hexagonal cells***, close in shape to regular hexagons, covering the entire visual scene. In the simplest case, a visual scene is represented by a sequence of frames from a single video camera. In more complex systems, it can be a combination of frames from several cameras or one camera that changes the direction of the sight to cover the whole scene.

A hexagonal grid, in comparison to a rectangular one, provides the maximum closeness to ***isotropy of properties*** and has a single type of cell neighborhood (all six cell neighbors are equal, while in a square grid, half of eight neighboring cells have a common border, and the rest have a common vertex). In the proposed realization, the cell has an area of 62 square pixels. Both criteria are calculated for each cell.

The data describing the analysis result is divided into a ***static*** component (coordinates of the centers of cells and lists of neighbors) and a ***dynamic*** (time-dependent) component (values of two criteria). The data volume of the dynamic component is, respectively, two orders of magnitude less than the volume of the original pixel array representing the video frame.

The ***edge detection*** and ***motion detection*** on video frames are well known; our approach, however, has significant differences.

The classical approach is based on applying '***filters***' to the ***pixel field*** and produces a ***new pixel field*** of the same dimension as the original one. That is, the resulting field is as big as the original one and is structured as weakly as the original one. ***Further analysis is required*** to obtain the desired estimates of the informativeness and dynamism of scene fragments, while the amount of required computational resources is enormous. In our case, the volume of the results is much less, and they are pretty structured (cells, for example, can be sorted according to the degree of importance according to the values of the criterion - which does not make sense for a pixel field). No less important is the fact that to assess the information content (detection of contours) and dynamics, algorithms are used that are ***different from the classical ones***, providing ***insensitivity to noise*** in the video stream, consumes less computational resources, and allow a high level of parallelism of computations.

The result of our experiments is the choice of the ***standard deviation of the current brightness level from the linear approximation of the brightness within the cell*** as a measure of the informativeness and the ***standard deviation of the brightness difference of two successive frames within the cell*** a measure of dynamism.

A relatively low-power computer (AMD Fx-4100, four cores, 2GHz) provides processing of a video stream from a USB-3 camera with a resolution of 1280x720 at a rate of 20 frames per second without involving GPU and 30 frames per second for a 640 x 480 frames. A more powerful desktop (AMD Rizen7 2700, 8 cores, 16 threads, 3.7GHz) provides 30fps at 1920 x 1080.

For a visual assessment of the results, the test application draws a honeycomb mesh, coding two criteria of each cell in color: the brightness of the ***blue-green*** component encodes the level of ***informativeness*** (the presence of a contour fragment in the cell), the brightness of the ***red-green*** component encodes the level of ***dynamism***. The above picture shows in orange a highly dynamic element, a swinging pendulum.

The repository https://github.com/mrabchevskiy/visual-attention contains C++ sources, shell scripts for compilation and execution, and a ready-to-use executable for 64 bit Linux. For visualization, the open-source library **SFML** is used, available for installation by package manager in all major Linux distros (package '***libsfml-dev***'). The video stream source requires a USB camera compatible with the Linux video driver ('***v4l2***'), so the application, unfortunately, cannot run in the Windows Subsystem for Linux (***WSL/WSL2***) environment.

The subsequent stages of the visual scene analysis will be discussed separately.

Subscribe

← Previous Next →

Write a comment…