Salience Detection for Vision-Based Robot Navigation

Enric Celaya

Institut de Robòtica i Informàtica Industrial (IRI)
UPC-CSIC
Llorens i Artigas 4-6, 08028 Barcelona (Spain)
Celaya@iri.upc.es

Abstract

This paper presents a new approach to detect salient regions in an image. A number of units is used to respond to the most salient regions, adapting their response to the size and range of colors present in each region. The system can be directly used on a sequence of images, continuously adapting its output with time. A quality estimation of each unit allows to select the most relevant regions present in the image at any time. Experiments performed on test images showing a robust behavior of the system are presented.

The output of this process can be used in a visual landmark-based navigation system to determine what parts of the image should be explored to find the most useful landmarks.

1 Introduction

Robot navigation in unknown outdoor environments is still an unsolved problem. The utilization of global positioning systems can help in many situations, but they are not available everywhere, and are of limited utility when there is no previous knowledge of the navigation environment. The use of vision is the natural choice in most navigation tasks, but the problems still to be solved in this area constitute a serious obstacle for its ready application. Many vision-based methods developed for structured indoor settings are not useful in natural environments because of the lack of regular structures to be identified, not to mention the problems derived from the variations in the illumination conditions that take place naturally. Further developments are needed in the field of robot vision until visual-based navigation becomes a commonplace.

To drive a robot towards a visually defined goal, it is necessary to build some internal representation of the environment. For medium to long navigation tasks, a complete reconstruction of the environment is too costly, and often unnecessary. A more suitable approach is to use visual landmarks to determine the position of the robot relative to them and to the goal [5]. An important problem in this case is the selection of reliable landmarks, which must be easy to find and identify from different points of view, a reason for which using color images is much preferable. Existing image processing techniques may be applied to select and characterize the most promising objects in the image, but the processing time they require makes their real-time application difficult [3, 4, 6]. This problem is especially relevant in mobile robot applications where the available computational resources are often very limited because they must be all carried on-board, whereas real-time operation is mandatory, since failing to react in time to an event may cause the robot to collide, or missing the right way to the goal.

One important problem of image processing is the large amount of data coming from the camera that must be dealt with. The volume of incoming data could be reduced on different ways, such as lowering the spatial or the color resolution of the images, or lowering the frame rate, but these are in detriment of image quality and would impoverish the possibly achievable results. A more promising technique consists in selecting small parts of the image where relevant objects are expected to be found and limit the more detailed examination to only these regions. This is sometimes accomplished in active vision systems by concentrating the most expensive processes to the foveal area, which often is obtained with increased resolution as compared to the peripheral zone [2]. This approach relies on the capability to detect salient regions in a fast way, so that attention can be shifted to them when necessary, either by performing fast saccadic movements in order to place them in the fovea, or just by selecting a specific image subarea for further processing.

This paper presents a method to detect salient regions of the scene that in the case of robot navigation are expected to be good candidates to constitute useful landmarks. A main feature of this method is that the visual input is not treated as a sequence of individual image captures that must be fully processed one after another, but as a continuously evolving image stream that is processed at the same time as it changes, in a metaphor of how biological visual systems work. A general overview of the method is given in Section 2, and a more detailed description is provided in Section 3. In Section 4, some preliminary experimental results are presented, and Section 5 suggests several improvements or modifications of the method that we expect to introduce in the near future.

2 Overview of the system

The visual input coming from a camera is highly redundant: nearby pixels tend to have similar color values, and, often, they experience only little changes from frame to frame. Examining every pixel at each frame acquisition is too costly and not very effective. Furthermore, from the whole information contained in the images, only a small part happens to be meaningful for the task to be carried out. Despite this, determining which part of the information must be retained and which part can be dropped away seems impossible without previous examination of all of it. By this reason, it would not be clever to arbitrarily crop part of the incoming data to reduce processing times.

The method proposed here does not exclude any pixel a priori, but it takes pixels at random from the whole image and processes them individually. In this way, the amount of pixels processed in each frame depends on the processing resources available, and not on the design of the algorithm, so that resource utilization is optimized as much as possible. By its side, the output of the algorithm can be read at any time and is always meaningful, varying continuously as more pixels are evaluated. When a new frame is acquired, the updating process does not start from scratch for the new image, but continues updating the already obtained results with the new data. This makes the method well suited to deal with the smoothly varying visual streams obtained by the camera of a moving robot. The output, following the input, evolves smoothly with time. Hardware-dependent aspects of the image acquisition process, such as frame rate or image resolution, can be safely ignored by the algorithm, thus making it completely independent of them. Since the algorithm does not take into account if one pixel comes from one frame or the next one, it can be implemented by directly accessing the memory area that is being updated by the input device, avoiding the need for double buffering techniques.

The output of the system consists in the values taken by a number of units that compete to respond to interesting parts of the image. Each unit defines a domain in the visual space specifying the location, extension, and range of color values of the pixels to which it responds. Each time a new pixel is processed, the values of the unit responding to it are updated, and, in the case that no unit responded appropriately, a new unit can be allocated to respond to it.

Each unit maintains an estimation of its quality, or interest level, which evaluates the color difference of its image domain with respect to its surroundings. In this way, taking the units with higher quality at a given time provides a concise description of the most interesting parts in the current image.

3 Algorithm description

The system manages a number of units with the following set of adjustable values:

- U_X, U_Y: Image plane coordinates of the unit location.
- U_R, U_G, U_B: Set of color values[1]
- w_X, w_Y: Weight values associated with the X and Y coordinates.
- w_R, w_G, w_B: Weight values associated with the color components
- Q: A quality measure.

The number of units is defined by a parameter of the system, *numunits*, that can be set by the user and is kept constant throughout the process. Despite of this, some of the units can be temporarily made inactive by the algorithm.

Initial values of U_X, U_Y, U_R, U_G, and U_B of each unit can be determined by picking a pixel at random from the image and assigning the corresponding values to them. All weights are initialized to some value, e.g. 0.1, and the Q value is set to 0. The exact initial figures are not important for the algorithm since they will gradually evolve towards the appropriate values.

At each step, a pixel I is selected at random from the image and its position coordinates, I_X, I_Y, and current color attributes, I_R, I_G, and I_B, are used by each unit U to compute two numbers: a weighted distance, $wdistXY$, in the image plane, and a weighted distance, $wdistRGB$, in the color space, using the formulas:

$$wdist^2_{XY}(I,U) = w_X(I_X - U_X)^2 + w_Y(I_Y - U_Y)^2 \tag{1}$$

$$wdist^2_{RGB}(I,U) = w_R(I_R - U_R)^2 + w_G(I_G - U_G)^2 + w_B(I_B - U_B)^2 \tag{2}$$

A unit U is said to respond to the input pixel I, iff $wdist_{XY}(I,U) < max_{XY}$ and $wdist_{RGB}(I,U) <$

[1]Of course, the choice of RGB, HLS, or any other set of pixel attributes is possible.

max_{RGB}, where max_{XY} and max_{RGB} are parameters of the system. According to (1), the locus in the plane image of the inputs with $wdist_{XY} = d$ is an ellipse centered at (U_X, U_Y) and principal semiaxes of length $d/(w_X)^{1/2}$ and $d/(w_Y)^{1/2}$, respectively. Similarly, from (2), the locus of the inputs with $wdist_{RGB} = d$ is an ellipsoid centered at (U_R, U_G, U_B), and principal semiaxes of length $d/(w_R)^{1/2}$, $d/(w_G)^{1/2}$ and $d/(w_B)^{1/2}$, respectively. Thus, weights provide a measure of the span of the receptive field of the unit in the corresponding dimension: the smaller the weight, the larger the interval of possible values.

If one or more units respond to the input, the unit among them with lowest spatial weighted distance $wdist_{XY}(I, U)$ is taken as the winner of the competition, and is updated using the values of the input I, as explained in section 3.1. If no unit is responding, an inactive one, or the unit with the lowest Q value, can be used to account for the current input I, as will be explained in section 3.3.

3.1 Updating of the winner unit

The location of the winner unit is shifted towards that of the input pixel by an amount proportional to its difference, according to:

$$U_X \longleftarrow U_X + \gamma(I_X - U_X) \tag{3}$$

$$U_Y \longleftarrow U_Y + \gamma(I_Y - U_Y), \tag{4}$$

with $0 < \gamma < 1$. Similarly, the U_R, U_G, U_B color values are shifted towards those of I using expressions completely analog to (3),(4).

The weight values w_i are updated so that the contribution of each component to the weighted distance approaches to 1:

$$w_i \longleftarrow \alpha w_i + \frac{1 - \alpha}{max(1, (I_i - U_i)^2)}, \tag{5}$$

with $0 < \alpha < 1$. The max operation with the value 1 in the denominator is just for the purpose of avoiding an eventual division by 0. With this updating rule, in the long term, the weight values will tend roughly to the mean value of the inverse of the squared difference between the unit values and the corresponding input values:

$$w_i \longrightarrow \left\langle \frac{1}{(I_i - U_i)^2} \right\rangle . \tag{6}$$

In this way, the range of values to which the unit will respond with a value of the weighted distance below the maximum allowed by max_{XY} and max_{RGB}, will gradually adapt to the actual extension of a similarly-colored region of the image. To see why, imagine the situation in which a region of similar pixels in the image extends beyond the domain of a given unit. In average, the inputs to which the unit responds will give large values of the weighted distance, so that rule (5) will tend to decrease the weights, that is, enlarge the receptive field of the unit. Conversely, if the unit covers a domain that goes beyond the real span of the corresponding region of the image, most inputs will fall short in the weighted distance, and weights will be shifted upwards, reducing the response domain of the unit.

To update the quality value Q of the winner unit U, we compute the non-weighted distances, $dist_{XY}(I, U')$ and $dist_{RGB}(I, U')$, from the input to the unit $U' \neq U$ with lowest spatial weighted distance, $wdist_{XY}(I, U')$, and the following updating rule is applied:

$$Q \longleftarrow \beta Q + (1 - \beta)\frac{dist_{RGB}(I, U')}{dist_{XY}(I, U')}. \tag{7}$$

With the updating rule (7), units that are near to units with very different color values will get large Q values.

3.2 Elimination of redundant units

A problem with the quality updating rule (7) is that, if eventually, two units are responding to roughly the same image domain, they will decrease the quality value of each other, even if the domain is really salient with respect to other nearby units. To avoid this, a test is performed before this update is done: The nearest unit U' is taken as an input pixel for the winner unit to see if the later responds to it. If the answer is affirmative, it means that both units are covering roughly the same image domain, and therefore they are redundant. In this case, the second unit is marked as inactive and excluded from subsequent competitions, and the Q update of the winner unit is made using the next nearest unit. Later on, inactive units may be recycled to account for pixels to which no unit responds, as we will see next.

3.3 Unit relocation

When no unit responds to an input, the first inactive unit available is taken in order to account for it. If there are no inactive units available, the active unit with lowest quality can also be used. However, to avoid undesired elimination of units that, due to temporal fluctuations, accidentally take an unfairly low quality value, this is done only when its current quality value is well below that which can be expected for the new location, as explained later.

For the new U_i values of the selected unit, the I_i values of the input pixel are taken. For the w_i weight values, those of the nearest active unit are used. The reason for this is to provide the unit with a response domain of a shape and volume comparable with that of nearby units. Note that appropriate weights cannot be arbitrarily fixed beforehand since they depend on the resolution, texture, and color ranges of the image, and a bad choice of weights could lead to a too small domain, in which case the probability to respond again to future inputs would be too small, or conversely, to a too broad domain, in which case the unit would be too general so as to always win in front of other more accurate units. In both cases, the unit will probably be soon eliminated because of its small Q value, but then, our purpose of creating a new unit to respond to a potentially useful domain would not be achieved.

The new Q value for the unit is computed using the updating rule (7), where 0 is taken as the previous value of Q. This is clearly an underestimation of the expected real value of Q by a factor $(1-\beta)$, and this is done in order to avoid the proliferation of units responding to sporadic pixels perhaps affected by noise. In the case that the unit was an already active unit, it is this new Q value that is compared with the previous one to decide if the relocation is made or not.

4 Results

We present some initial results of the tests performed on two example static images. For the first example a synthetic image formed by a number of patches of uniform colors is used. The second example is a real full-color image. Both of them have been processed with the same parameters, which are listed in Table 1.

PARAMETER	VALUE
numunits	20
α	.9
β	.9
γ	.1
max_{XY}	15
max_{RGB}	15

Table 1: *Parameters used in the examples*

No extensive exploration has been performed in the parameter space in order to optimize the results. Note that even having defined three different parameters for the update of the units, as far as the present tests concerns, a single one would have been enough simply making $\alpha = \beta = (1-\gamma)$. The same applies to max_{XY} and max_{RGB}. This seems to suggest that a precise parameter setting is not too critical.

4.1 Example 1

Fig. 1 shows a 160×120 artificially generated image. All pixels in each different color region have exactly the same RGB values, what we expect will make the convergence process of the units easier. In the figure, the small square at the center of the image is probably the most salient object to us (a red region surrounded by blue sky), thus we expect that it will be detected by the system and given a high quality rate. However, the small area occupied by this object makes relatively improbable that a pixel is selected on it. In fact, in the experiment presented here, only one pixel in this region was selected by the random process after 1,000 iterations.

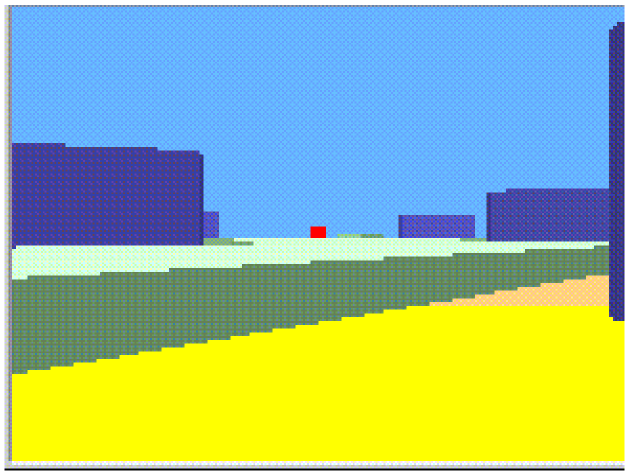

Figure 1: 160×120 *color image of Example 1.*

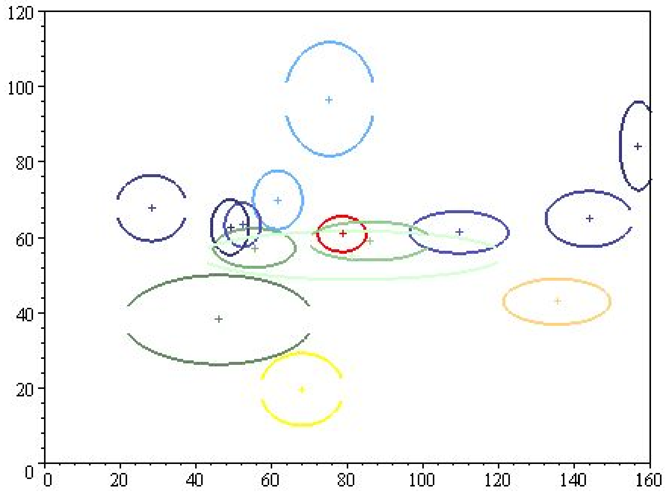

Figure 2: *Results for Example 1 after 10.000 pixel evaluations.*

Fig. 2 shows the result after 10,000 iterations, which is roughly the 50% of the number of pixels in the image. Only the 15 highest quality rated units are shown. Each unit U is represented by an ellipse with center (U_X, U_Y), color (U_R, U_G, U_B), and a size proportional to its receptive field. We can see that one unit at point (78.6, 60.8) has been allocated to respond to the red square, as well as other units are found for most of the different blocks and parts of the sky and ground regions. In the experiment shown here, the red square is detected as the most salient region, with a value of Q= 21.39, while the second most salient region, with Q = 18.55, corresponds to the unit placed at point (81.3, 55.2) of the image (corresponding to the bright horizontal stripe below the red square). For the sake of comparison, the least relevant unit represented in Fig. 2, with Q = 3.81, corresponds to the green ellipse located at (55.4, 57.2). Different experiments performed on the same image gave similar results, with the red square always appearing among one of the five most relevant units.

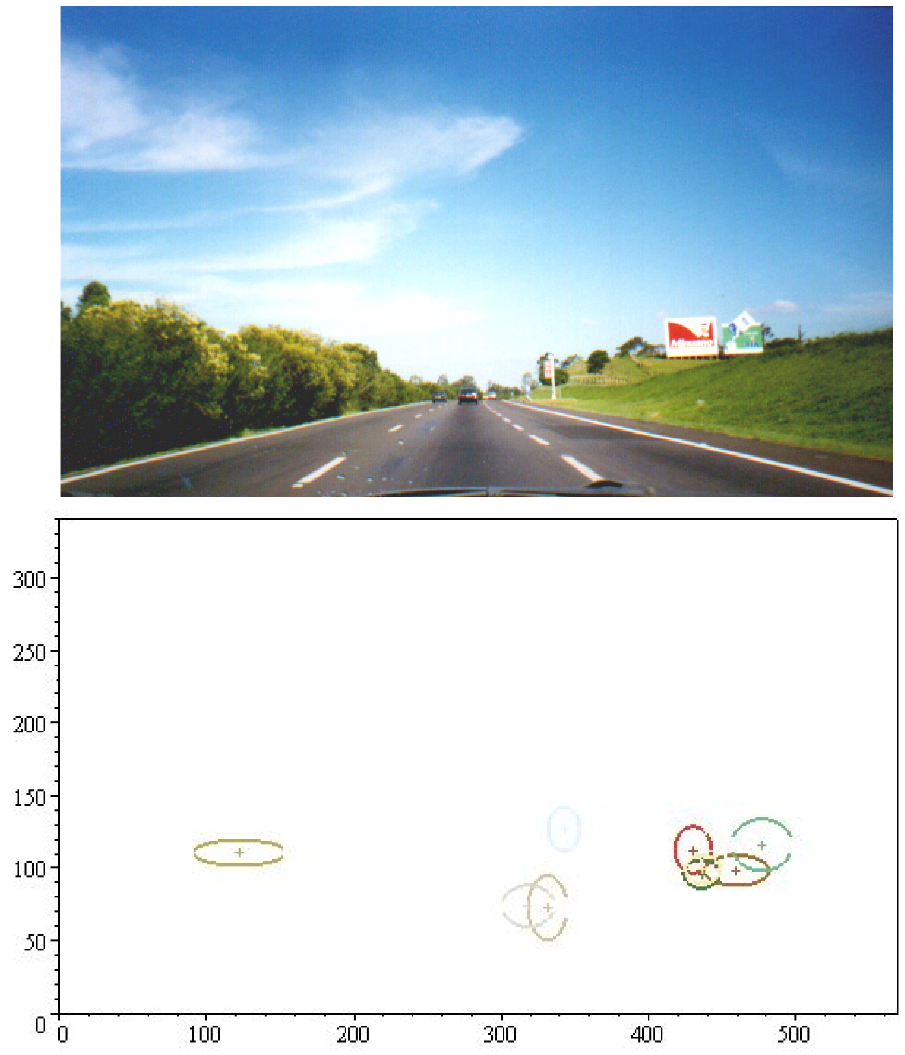

Figure 3

4.2 Example 2

The top of Fig. 3 shows a real 569 × 340 image with 8 bit precision RGB values. In this case the signs at the right of the road seem to attract our attention due to their contrasting red and green colors. The bottom of Fig. 3 shows the result of one experiment after performing 10,000 updates, what is about 5% of the number of image pixels. Only the 10 highest quality rated units are shown, though some of them are almost invisible due to its nearly white color. Among them, 6 units are concentrated on the different colors contained in the signs.

In this case, the maximum value of Q is 26.38, and corresponds to an almost completely white unit located at point (443.8, 112.2), with RGB values 252.9, 251.1, and 245.7, that corresponds to a white part of the sign. The remaining units in Fig. 3 reached Q values between 8.68 and 3.77.

In a series of experiments on this image, the region of interest associated with the signs was always detected with several of the highest Q rated units. This confirms that the system is able to reliably find the most relevant regions also in real images.

5 Future work

There are a number of aspects of the system described here that can be improved. Perhaps the most relevant would be to allow the domains of response of the units to take the shape of general ellipses instead of having their axes along the coordinate directions. This would allow to better account for regions that appear diagonally in the image plane, as well as for domains in which the color presents correlated variations of its components.

There are other variations of the system that can deserve further exploration, as for example:

- Allowing the number of units to increase as they are needed or decrease when they are not.
- Improve the quality estimation of units, maybe making it less dependent of other existing units, and more intrinsic to itself.
- Updating all the units responding to an input instead of only the winner one.
- Letting the system to automatically adjust the parameters max_{XY} and max_{RGB} to adapt to the image properties.

Some of these variations have already been partially explored, and some of them will be considered in the future. The results presented here are in no way the final stage of this investigation but the current state of an ongoing research that is expected to give better results as more work is done. Of course, an important test still to be done is to apply the system to a stream of input images instead of using a single constant image. We expect to carry out such test soon.

The next step after this work would be to incorporate this salience detection system with a higher level that takes the regions suggested by it and, after examining them more carefully, selects those that will constitute real landmarks to be used in a navigation task [1].

6 Summary and Conclusions

We have presented a salience detection system able to detect those regions of the image that are the most uniform and differentiated from the rest of the image, and therefore can be good candidates for constituting easy to identify visual landmarks, as required in outdoors navigation tasks.

The units defined in the system automatically adapt to the size of uniform regions of the image as well as to the range of color variations they contain. The relative salience of each detected region is estimated by a quality measure computed by each unit.

The system has shown to be robust: tests performed in real images show a great consistency in the localization of the most relevant regions, providing similar results in different runs.

The system is well suited to deal with a continuous stream of images, and provides a continuously updated output. The characteristics of the method allow an efficient use of the processing power available in each situation.

Acknowledgments

This work has been supported by the *Dirección General de Investigación del Ministerio de Ciencia y Tecnología* under the project "Autonomous robot navigation guided by visual targets" (DPI2000-1352-C02-01).

References

[1] Celaya, E. and Torras, C. (2002), "Visual Navigation Outdoors: the ARGOS Project", *7th. International Conference on Intelligent Autonomous Systems* (IAS-7), Marina del Rey, March 2002, pp. 63-67.

[2] Davison, A. and Murray, D. W. (1998): "Mobile robot localisation using active vision", Proc. *6th European Conf. Computer Vision*, 1998.

[3] Fernandez-Vidal, X.R., Garcia, J.A., Fdez-Valdivia, J. and Rodriguez-Sánchez, R. (1999): "Computing visual target distinctiveness through integral opponent-color features", *VIII National Symposium on Pattern Recognition and Image Analysis*, Vol. I, Bilbao, May 1999, pp. 429-436.

[4] Itti, L., Koch, C. and Niebur, E. (1998): "A model of saliency-based visual attention for rapid scene analysis", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11), 1998, pp. 1254-1259.

[5] Kortenkamp, D. M. (1993): "Cognitive maps for mobile robots: A representation for mapping and navigation", PhD. thesis, Computer Science and Engineering Department, University of Michigan, 1993.

[6] Todt E. and Torras C. (2000): "Detection of natural landmarks through multiscale opponent features", Proc. *15th Intl. Conf. on Pattern Recognition* (ICPR-2000), Barcelona, vol. 3, pp. 988-991, Sept. 2000.