

## Perception Control for Obstacle Detection by a Cross-country Rover<sup>1</sup>

Pierrick Grandjean      Larry Matthies  
Jet Propulsion Laboratory - California Institute of Technology  
4800 Oak Grove Drive  
Pasadena, California 91109

### Abstract

Perception control consists in optimally tuning sensor or processing parameters in order to increase perception efficiency under requirement constraints or while adjusting to the environment. In this paper, we address perception control in the task of obstacle detection for cross-country navigation. We show how to maximize the vehicle safety at a given velocity, or inversely how to derive the maximum speed for a given safety. This optimization problem requires the joint analysis of (i) how the vehicle velocity sets look-ahead requirements, (ii) how the computational cost of perception is related to the perception variables, window of attention and image resolution, and (iii) how the reliability of the obstacle detection system is related to these variables. This criterion relies on experimental performance statistics. Our system has been implemented and tested in outdoor operation.

### 1 Introduction

Perception, defined as signal acquisition and interpretation, is traditionally considered as a complex and computationally expensive process. However, current perception systems are often doing more than necessary for the task at hand. A parallel approach to brute-force sensor development is to consider perception as a controllable process, i.e. to use available knowledge for purposefully tuning the sensor's controllable variables, in order to optimally perform the task. This approach has also been termed *active perception* [1], *perception planning* [2], *task-directed perception* [3] or *sensor management*, and is receiving increasing attention.

Knowledge relevant to perception control includes (i) the environment (world model), (ii) the sensor, or perception system (including processing modules), (iii) the task itself (performance criterion). This knowledge

is usually inaccurate and/or incomplete. Perception parameters include the sensor location, physical and device acquisition parameters (aperture, focus, focal length, gain, . . .) and processing parameters. Optimally achieving a task means minimizing perception cost (computation time) and maximizing performance (reliability). Usually these are competing goals and modeling how changing parameters affects both performance and cost is essential.

This paper shows how to instantiate these general perception control issues for the specific task of obstacle detection during cross-country navigation of a robot vehicle. Obstacle detection is an essential component of robot vehicle systems for planetary exploration, defense reconnaissance, and other applications. In this context, the role of perception control is to allocate computing resources so as to increase navigation efficiency while staying within specified safety margins. One of the primary ways to reduce the computational cost of perception is to process sub-windows of the image. This focus-of-attention technique is often used in road-following and feature-tracking systems [4] [5] [6]. In this paper, we extend this by explicitly taking into account geometric uncertainties and by adding the image resolution as a controllable parameter.

In order to determine the perception parameters maximizing safety at a given driving speed, we must (i) derive a perception reliability measure and relate it to the controllable variables of the perception system, (ii) analyze how the computational cost of perception depends on these variables, (iii) understand how the driving speed and the computation time set requirements on the part of the path that must be observed at each step (look-ahead requirement) and (iv) show how the previous three issues are interrelated and how to solve the resulting optimization problem. The paper begins with a review of the design of the obstacle detection system in section 2, then addresses the above issues in sections 3 to 6. Section 7 discusses the implementation of the control strategy and the experiments

<sup>1</sup>This work was supported by the National Aeronautics and Space Administration (NASA) and the Defense Advanced Research Projects Agency (DARPA)

that have been performed to date.

## 2 Obstacle detection for cross-country navigation

We require that the vehicle drive continuously at a velocity for which "adequate" safety margins in the presence of obstacles can be assured. For simplicity, we have begun by developing a perception control methodology assuming that the response to detecting an obstacle is to stop before it. The methodology applies to any range imaging sensor and a variety of obstacle detection algorithms; in this paper, we apply it to a passive, stereo vision-based range imaging system and a very simple obstacle detection algorithm that JPL has previously demonstrated in successful outdoor navigation experiments with both NASA and military vehicles [7] [8].

The main processing steps of the stereo vision and obstacle detection system are as follows:

1. build bandpass-filtered image pyramids from an input stereo image pair;
2. perform cross-correlations on any single level of the stereo image pyramid to estimate disparity at every pixel (nominally) of the image pair at the chosen level of resolution;
3. compute the range from the disparity at every pixel,
4. apply obstacle detection algorithms to the resulting range image.

Steps (1) to (3) are described in detail in [8]. To date, the obstacle detection algorithm (step (4)) has been kept very simple to enable real-time implementation with a single 68040-based CPU. Obstacles are assumed to be near-vertical step displacements on an otherwise flat ground plane. As illustrated in figure 1, the algorithm checks for such obstacles by using pairs of pixels ( $i$  and  $j$ ) in the same column of the range image to measure the height displacement between the two pixels. For each pixel  $i$ , the included angle between  $i$  and  $j$  lines of sight is set to subtend a fixed obstacle size, denoted *stepheight*. If the height difference measured between  $i$  and  $j$  (denoted  $\hat{h}$ ) exceeds a given threshold  $t$ , then the pixel is marked as an obstacle. An illustrative example is shown in figure 2 where pixels at which an obstacle has been detected are outlined.

This obstacle detection system can be controlled at three levels:

- image resolution (from 64x64 to 512x512),
- window of attention: steps (2),(3) and (4) (stereo vision and obstacle detection) can each be performed in a specific sub-window.
- detection threshold, which may vary over the image.

To determine how to choose these parameters, we will develop a model of their effect on the reliability of perception, then derive the mutual constraints between the perception parameters and vehicle velocity.

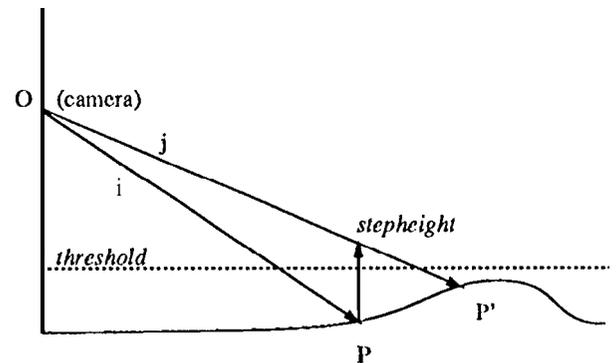


Figure 1: Obstacle detection algorithm

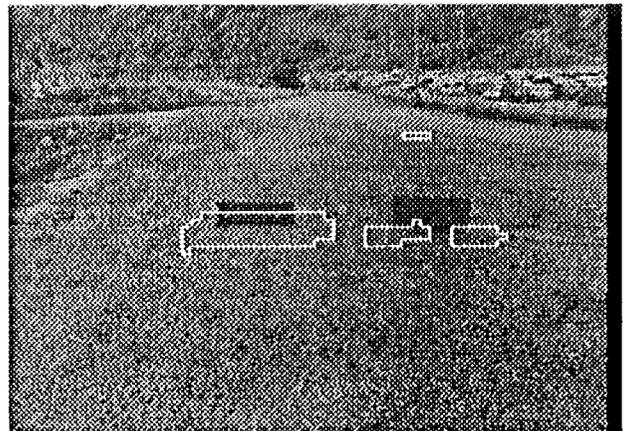


Figure 2: Example of obstacle detection result

## 3 Reliability criterion

The system performance is characterized by the occurrence rate of two possible failures: false alarms and

missed detections. A lower threshold  $t$  reduces the risk of missing an obstacle, but also increases the false alarm rate. Operating at a higher resolution reduces the rates of both errors, but at a higher cost. Decision theory offers a broad framework for tackling this kind of trade-off problem.

### 3.1 Decision theory basics

Suppose that we have to take a decision  $d$  in a space  $D$  whose consequences depend on an unknown state of nature  $w$  in a space  $\Omega$ , characterized by a probability distribution  $p(w)$ . The consequences are weighted by a loss function  $L(d, w)$ .  $L(w, d)$  is the loss caused by taking the decision  $d$  if  $w$  is the true state of nature. The expected loss, or *risk*, of taking the decision  $d$  without knowing  $w$  is:

$$R(d) = \int_{\Omega} L(d, w) p(w) dw$$

Usually, the loss of a correct decision is 0 and failures are associated positive losses. The problem is to find the decision that minimizes the risk.

### 3.2 Obstacle detectability

The process that we consider is the obstacle detector at one pixel. This process includes the detection decision rule: if the measured height  $\hat{h}$  is more than the threshold  $t$ , the outcome is O (for obstacle), if not the outcome is  $\emptyset$  (for traversable terrain).

The decision problem takes place at the level of the design of the decision rule, i.e. we want to specify the "best" threshold  $t$ . The decision parameter is  $t$  and the state of nature is the actual obstacle height  $h$  at the observed 3-D point  $P$  ( $h$  is 0 where there is no obstacle). The loss caused by the detection threshold  $t$  if the actual height is  $h$ ,  $L(t, h)$ , is defined from the elementary detection losses  $C(O, h)$  and  $C(\emptyset, h)$ .  $C(O, h)$  is the loss induced by detecting an obstacle if the actual height is  $h$ , and  $C(\emptyset, h)$  is the loss corresponding to a non-detection.

The loss induced by the threshold  $t$  is:

$$L(t, h) = p(\hat{h} \geq t|h) C(O, h) + p(\hat{h} < t|h) C(\emptyset, h)$$

The probability distribution of  $\hat{h}$  depends on  $h$ , but also on  $r$ , the horizontal distance  $r$  from the camera to  $P$ , and the resolution. Consequently,  $L(t, h)$  depends on  $r$  and the resolution.

The unknown terrain can be modeled by a flat area scattered with obstacles of different heights, characterized by a prior distribution of  $h$  at every point, which induces a distribution of  $h$  at every pixel,  $p_h(h)$ . Given this *a priori* information about the terrain, the risk (expected loss) of processing for a given pixel is:

$$R(t) = \int_{Jh} L(t, h) p_h(h) dh$$

Next, it would be logical to integrate the risk over the area of the sub-image being processed. This raises a number of complexities that we have not yet resolved, such as inter-pixel dependencies. Therefore, in this paper we formulate perception control in terms of the risk at a given distance, and leave the question of integration over a sub-image for future work. However, we will compensate for the number of pixels corresponding to the same image area when comparing different image resolutions.

### 3.3 Failure losses

Let  $H$  be the maximum height of steps that can be managed by the vehicle without effort. This height  $H$  is typically used to define the step height parameter in the detection algorithm. The correct behavior is to detect an obstacle if  $h \geq H$ , so

$$C(O, h)_{h \geq H} = C(\emptyset, h)_{h < H} = 0$$

**False alarm**  $C_f = C(O, h)_{h < H}$ : if an obstacle is detected where  $h < H$ , the system will either stop and examine the obstacle with more accurate means, or plan an avoidance trajectory. This loss is usually only time.

**Missed detection**  $C_m = C(\emptyset, h)_{h > H}$ : if an obstacle of  $h > H$  is not detected, two cases must be considered. If the vehicle has lower level sensors, it may be able to avoid locally the unexpected obstacle, and the loss is only time. If not, the failure will result in a crash and put an end to the mission. In the latter case, it is better to consider the false alarm risk and the crash risk as two different criteria.

Replacing these values in the previous formula gives:

$$R(t) = C_f \int_{h=0}^H p(\hat{h} \geq t|h) p_h(h) dh + C_m \int_{h=H}^{\infty} p(\hat{h} < t|h) p_h(h) dh$$

### 3.4 Experiments] evaluation

WC have experimentally evaluated the performance of our system from a statistical set of images. It consists of sets of 100 images a flat area with no obstacle and sets of 100 images of the same area with one obstacle placed at a given distance, for three different obstacle sizes (20cm, 30cm or 40cm) and three different distances (6m, 10.5m, 15m). The image in figure 2 is actually one of these images, with two obstacles at 10.5m. Both obstacles have been correctly detected, and we can also see a false alarm in the distance (at approximately 25m).

The probability of detection  $p(\hat{h} \geq f)$  has been registered for flat ground and for each obstacle size at each distance, for 15 different stepheights (9cm-79cm), 15 threshold values for each stepheight and at two resolutions (64 and 128). This globally yields a 5-D table, registering how the detection probability varies with the actual obstacle height  $h$ , the distance  $r$ , the stepheight, the threshold  $t$  and the resolution.

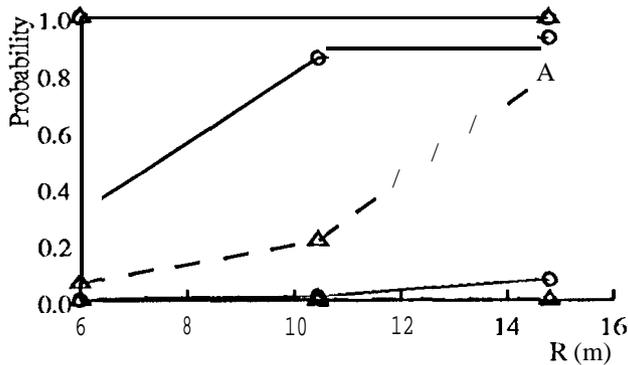


Figure 3: Detection statistics. From top to bottom: 30cm obstacle, 20cm obstacle, flat ground. Plain line, circles: resolution 64. Dashed line, triangles: resolution 128. The stepheight is 29cm and the threshold is 20cm

Figure 3 shows sections of this table for three actual heights (30cm, 20cm and flat ground), at resolution 64 and 128. As compared with a theoretical stochastic model of detection, that we derived from the disparity variance, these results showed that systematic range errors (mainly bias occurring at disparity discontinuities) have significant effects. The results shown here are not raw data but upper-bound values covering these effects. More details on this performance evaluation can be found in [7] and [9].

As the threshold value has no effect on the computation time, it can be determined *a priori* for each

distance and each resolution. Figure 4 shows the risk versus distance, at resolution 64 and 128, for a constant image area (i.e. 1 pixel at 64, 4 pixels at 128).

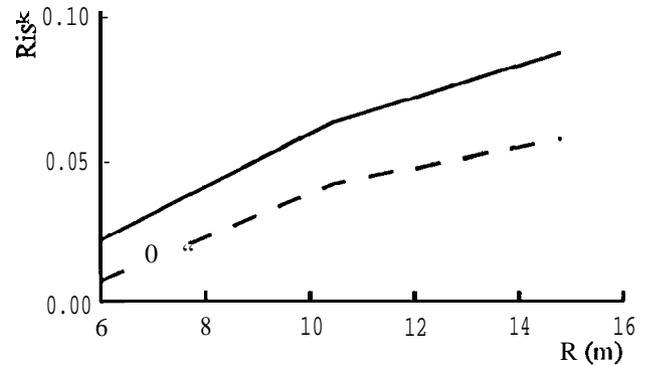


Figure 4: Global risk at resolution 64 (plain line), and 128 (clashed line). Missed detection loss is 100, false alarm loss is 1. The stepheight is 29cm. The obstacle distribution is  $p_h(0) = 0.8$ ,  $p_h(0.2) = 0.09$  and  $p_h(0.3) = 0.01$ ,

## 4 Look-ahead requirements

The most conservative way to ensure vehicle safety is to enable it to stop before colliding with any detected obstacle. This defines a minimum look-ahead distance and determines the part of the path that must be checked for obstacles at each processing step. This is illustrated by figure 5. At each processing step, the acquisition and processing of the stereo pair take time  $t_c$ . If an obstacle is detected, braking only begins after an actuation latency  $t_a$ , and the braking distance is  $v^2/2a$ , for initial speed  $v$  and deceleration rate  $a$ . Finally we must allow for the horizontal distance from the camera coordinate frame origin to the vehicle front bumper, noted  $d_c$ . Adding all these terms gives the total distance from the cameras to the place where the vehicle is able to stop:

$$r_0 = d_c + v(t_c + t_a) + v^2/(2a) \quad (1)$$

If the vehicle does not need to stop and begins a new perception step at  $t_1$ , then it will next be able to stop at distance:

$$r_1 = r_0 + v t_c = d_c + v(2t_c + t_a) + v^2/(2a) \quad (2)$$

Therefore, in the steady state, processing at each step the path segment between  $r_0$  and  $r_1$  ensures that the whole path is checked in time. We call the sub-image corresponding to the path segment the *window of attention*. This polygon is basically the projection of the

path segment onto the image plane, but geometric uncertainties about the system itself and the environment must be taken into account for computing it. This is detailed in section 6.

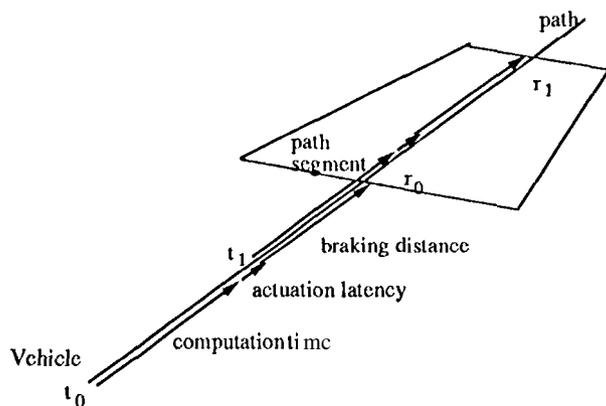


Figure 5: Path segment to be checked

Finally,  $t_c$  must be equal to the computation time required to process the window of attention. The computation time is a function of the size of the window of attention and the resolution<sup>2</sup>. We have determined the values of the coefficients of this function by benchmarking our algorithms. The area of the window of attention, through projection equations, is itself a function of  $r_0$  and  $r_1$  and the resolution (from now on denoted  $\rho$ ). Therefore, the computation time is also a function of  $r_0$ ,  $r_1$ , and  $\rho$ , and the third equation is:

$$t_c = T_c(r_0, r_1, \rho) \quad (3)$$

## 5 Optimization

The performance criterion that we want to optimize is the risk of the obstacle detector, as defined in 3, at the look-ahead distance  $r_1$ . It is a function of  $r_1$  and the resolution  $\rho$ , increasing with  $r_1$  and decreasing with resolution. The problem is to minimize the risk under the constraints (1), (2) and (3).

These three equations link the five variables  $r_0$ ,  $r_1$ ,  $t_c$ ,  $v$ ,  $\rho$ . Eliminating  $r_0$  and  $t_c$  in (3), using (1)

<sup>2</sup> More precisely, the computation time reduces to three terms: a constant term (images acquisition, communication delays and perception control), a term proportional to the area of the window of attention (triangulation and obstacle detection), and a term proportional to the window area and the resolution (stereo correlation, because the disparity search range is proportional to resolution).

and (2), yields an equation linking  $r_1$ ,  $v$  and  $\rho$ . Therefore, for a given resolution and a given velocity, solving this equation for  $r_1$  determines uniquely the window of attention and consequently the corresponding risk. Figure 6 shows the risk at distance  $r_1$ , versus velocity, at resolution 64 and 128.

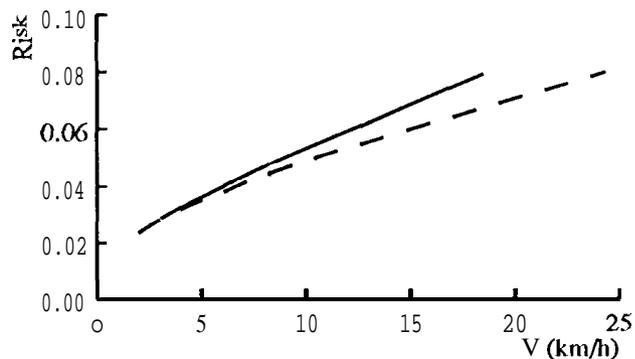


Figure 6: Risk versus velocity with focus of attention

If velocity is a task requirement, we select the resolution (and consequently the window of attention) corresponding to the lowest risk. In our particular case, the best resolution, between 64 and 128, is 128. Inversely, we can determine the maximum velocity that is compatible with a given level of risk. This latter way is more interesting if the risk is a probability of mission failure.

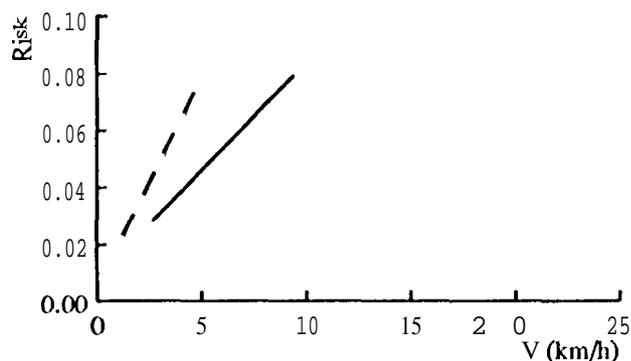


Figure 7: Risk versus velocity without focus of attention

In order to estimate the gain earned by focusing attention, let us now return to a system that processes the whole image (but detections are discarded if they are outside the path segment). In this case, the computation time only depends on resolution. The corresponding risk is shown on figure 7. The velocity compatible with a given risk is always much lower than for the attentional system, and that demonstrates the value of

focusing attention. Also, it is interesting to note that now resolution 64 is better than 128, because higher resolutions are much more costly in this case.

## 6 Focus of attention

The path segment that must be observed at each step is a 3-D polygon defined by two curvilinear distances along the path,  $r_0$  and  $r_1$ , and the width of the vehicle. The subpart of the left image where obstacles must be detected, which we call the detection polygon, is the projection of the path segment onto the image plane, after transformation in the camera reference frame, through the vehicle reference frame. This can be written as follows, with the following notations:  $D$  is the detection polygon,  $T_R$  is the vehicle attitude with respect to the path,  $T_{RC}$  is the calibrated transform between the vehicle and the (left) camera, and  $\Pi$  is the calibrated camera perspective projection.

$$D = \Pi \circ T_{RC} \circ T_R(P)$$

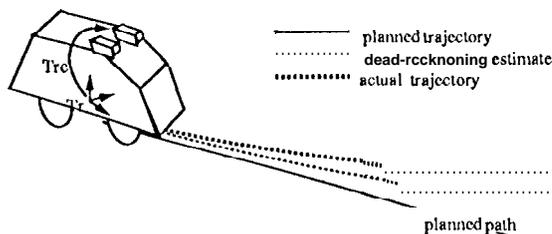


Figure 8: Path segment uncertainty

It is important, to take into account, the uncertainties of every component in this computation in order to ensure that the vehicle never enters a non-processed area. We achieve this by explicitly propagating uncertainties. We use a probabilistic, first-order representation of uncertainty (covariance matrices). First-order propagation requires only the jacobian matrices of every transformation with respect to each parameter.

Let us examine each term of the equation in turn. First, the 3-D path segment itself is uncertain, because the vehicle will not follow exactly the path for three reasons:

- the dead-reckoning inaccuracy,
- the trajectory execution control allowance,

- the inaccuracy of the terrain model on which the path has been planned.

We have so far bounded the two last uncertainties by constant, 3-D values, and the dead-reckoning inaccuracy by a variance growing as the square of traveled distance. Any representation of 3-D rotations can be used (Euler angles, quaternions, ...) and we have used the rotation vector representation. The camera calibration inaccuracy has been bounded by a uniform tolerance in the image plane.

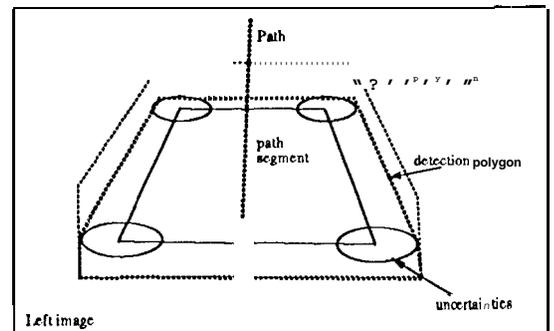


Figure 9: Sub-windows of attention: detection polygon, range polygon, disparity rectangle

The computation of the window of attention is illustrated by figure 9. First the uncertain vertices of the path segment are projected, and then the detection polygon is obtained as the convex hull of the possible locations of the projected vertices. The possible location of each uncertain vertex is approximated by the rectangle including the ellipse corresponding to a 95% probability of presence for a Gaussian distribution. Finally, as the performance evaluation showed that obstacles are generally detected at the few pixels preceding their base (because of bias effects), the polygon is enlarged downwards by the size of the correlation window.

Actually, two windows of attention are necessary, because the sub-window where range is required, called the range polygon, is not the same as the detection polygon. Indeed, applying the detection algorithm at a given pixel also requires a range measurement above it (recall figure 1). The range polygon is obtained by enlarging (upwards) the detection polygon by the number of pixels corresponding to *stepheight* at each row.

## 7 Illustrative examples and implementation issues

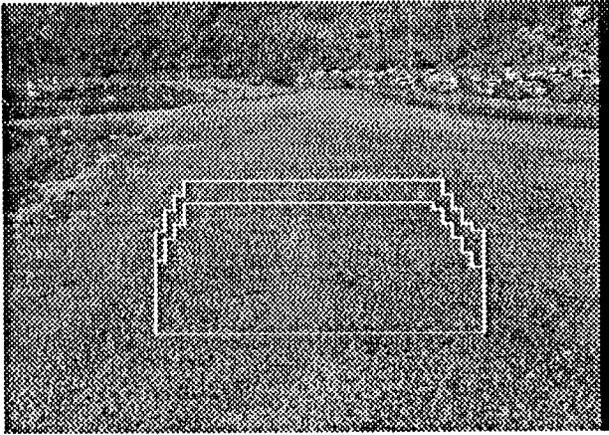


Figure 10: Window of attention: resolution 64

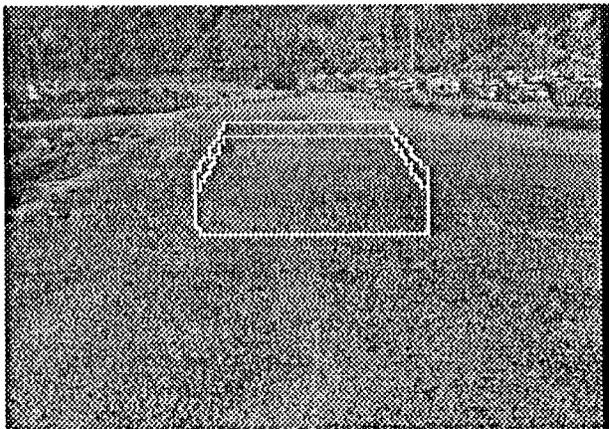


Figure 11: Window of attention: resolution 128

We have experimentally checked the validity of our design in the following (non-exhaustive) way. The detection system was running continuously with a static window of attention, and the vehicle was driven at constant speed along a straight path. Figures 10 and 11 show the range and the detection polygons corresponding to the same risk level, respectively at resolution 64 (at 11.2 km/h) and resolution 128 (at 13.54 km/h). The corresponding look-ahead distances are 10 m and 17 m, and the computation times are 0.88 s and 1.5 s. So far, we have only visually checked (using video means) that the parts of the path that are consecutively processed actually overlap (i.e. the path is entirely observed).

Computing the window of attention takes about 0.08 second on a 68040 processor, and we may wonder if it is necessary to compute it at each step. On one hand, some variables involved in the projection of the path segment arc dynamic: the vehicle attitude (with respect to the path), the path, and the terrain map may change. On the other hand, these frame-to-frame variations can be considered as additional uncertainties, and the same window of attention can be used for all frames. Such a static window is to be larger, but if the variations are small (straight path, flat terrain), the extra processing time may still be less than the window computation time.

An effective real-time control is still necessary to cope with real-world events that are not consistent with the models (such as unexpected computation or communication delays). A simple solution is to design a velocity controller that tries to reach the maximum allowed speed, but always check for the distance to the end of the last processed path segment, and brakes if it has to (i.e. if the next path segment obstacle map dots not come in time). If this happens, the focus window has also to be redefined to cope with the distance traveled during deceleration.

## 8 Future work

The above study is only preliminary in some aspects, and many further developments can already be envisioned:

- Performance models for more sophisticated obstacle detectors.
- Curved path: the window of attention must be dynamic and the velocity adjusted to curvature. Moreover, if the camera is mounted on a pan-and-tilt platform, they can be aimed so that the path segment lies entirely in the field of view.
- The response to a detection can be more sophisticated: the braking time can be used for examining more closely the obstacle, at a higher resolution for instance.
- The previous idea can be generalized in a coarse-to-fine obstacle detection strategy: the path segment following the first one can be processed at a lower resolution, and then the attention may be focused more precisely on the places where potential obstacles have been detected. Interestingly, the computation time will depend on the frequency of alarms but will still be predictable.

from frame to frame, enabling to adjust the vehicle velocity to the roughness of the terrain.

## 9 Conclusion

In this paper, we have addressed the problem of designing and optimally tuning an obstacle detection system, based on stereo vision and capable of focus of attention. Our main contribution has been to consider at the same time the relationship between the algorithm performance, the computation time and the system controllable parameters, and the look-ahead requirements due to continuous motion. We have shown how to determine the maximum velocity compatible with a guaranteed obstacle detection reliability, or inversely how to maximize the vehicle safety at a given driving speed.

### Acknowledgments

The authors wish to thank Brian Wilcox for valuable observations regarding stopping distances and error rates.

## References

- [1] R. Bajcsy. Active perception vs. passive perception. in *IEEE Workshop on Computer Vision: Representation and Control*, pages 55-62, Bellaire, MI, 1985.
- [2] S. Lacroix, P. Grandjean, and M. Ghallab. Perception planning for a multi-sensor interpretation machine. in *IEEE International Conference on Robotics and Automation*, Nice, France, 1992.
- [3] G. Hager and M. Mintz. Task-directed multi-sensor fusion. in *IEEE International Conference on Robotics and Automation*, pages 662-667, Scottsdale, AZ, 1989.
- [4] E.D. Dickmanns and B.D. Mysliwetz. Recursive 3-d road and relative ego-state recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):199-213, February 1992.
- [5] Y. Goto, S.A. Shafer, and A. Stentz. The driving pipeline: a driving control scheme for mobile robots. *International Journal of Robotics and Automation*, 4(1):8-18, 1989.
- [6] S. Singh and P. Keller. Obstacle detection for high-speed autonomous navigation. In *IEEE International Conference on Robotics and Automation*, page 2798, Sacramento, CA, April 1991.
- [7] L. Matthies. Toward stochastic modeling of obstacle detectability in passive stereo range imagery. in *IEEE Conference on Computer Vision and Pattern Recognition*, June 1992.
- [8] L. Matthies. Stereo vision for planetary rovers: stochastic modeling to near real-time implementation. *International Journal of Computer Vision*, 8(1):71-91, July 1992.
- [9] L. Matthies and P. Grandjean. Stereo vision and obstacle detection performance evaluation. Submitted to the 1999 *Conference on Computer Vision and Pattern Recognition*.