Visual Control of Autonomous Vehicles with an On-Board Camera

P. Marinov

***Abstract*—The idea of autonomous vehicles (AUVs) being used in logistics is gaining popularity. APCs capable of performing complex tasks such as autonomous navigation are an increasingly promising tool in logistics. Autonomous navigation requires sensors with high potential. The camera provides images from which it is possible to extract a large number of measurements for the APS - position, speed, distance from objects, etc. The area uniting vision and control is called visual control. The advantage of connecting vision with control is that the camera sensor has a great wealth of information. Its use has an undeniable advantage to perform complex tasks that require great precision. The work examines problems of visual control in the presence of a camera on board the APS. Control is sought only through the images captured by the camera under additional image analysis time constraints in order to update the command most frequently. The relationship between vision and control (visual assessment) is explored for the purpose of control based on the information provided by the video analysis. The combined study of the constituents of visual control leads to a new control enabling greater overlap between control law and image analysis and more efficient use of APS in logistics.**

***Index Terms*—autonomous vehicles, visual control, logistic**

I. Introduction

Since the beginning of the 21st century, the idea of autonomous vehicles (AUVs) being used in logistics has gained popularity. Over the years, advanced APCs capable of performing complex tasks such as autonomous navigation are an increasingly promising tool in logistics.

The complexity of the autonomous navigation task requires sensors with high measurement potential to be able to provide relevant control information. One of the sensors providing the most information is video cameras. Adding visuality to APS control laws gives more information increasing the degree of autonomy and interaction with the environment. The camera provides images from which it is possible to extract a large number of measurements such as the position of the APC in space, its speed of movement, the distance from objects of interest, as well as more semantic measurements of the environment such as the type of object present in the visual field. The area uniting vision and control is called visual control.

The work examines problems of visual control in the presence of a camera on board the vehicle. It seeks to control the camera only through the images captured by it, necessary for tasks such as tracking stationary objects on the ground, regardless of the movements of the APS. The task is considered under additional image analysis time constraints in order to update the command as often as possible.

Most research on visual control has focused on two separate strands:

– the development of new control laws using a highly simplified environment that allows easy extraction of information from the image,

– or new general image analysis algorithms focused on measuring the information in the image for easy integration into the control chain.

A combined study of the two topics constituting visual control is proposed here, leading to a new control allowing for greater overlap between control law and image analysis.

This work investigates the relationship between vision and control (visual assessment) for the purpose of control based on the information provided by video sequence analysis. This makes it possible to move the latter from a current position to a desired position with respect to the observed scene. The advantage of combining vision with control compared to other techniques is that the camera sensor has a great wealth of information, its use has an undeniable advantage to perform complex tasks that require high precision.

II. State of the Problem

A classic APS approach is with an embedded camera (Fig. 1 (a)) and APS or with a remote camera (Fig. 1 (b)). There is also the possibility of simultaneous use of a built-in and remote camera. This work describes the case of an embedded camera, which is most promising in logistics.

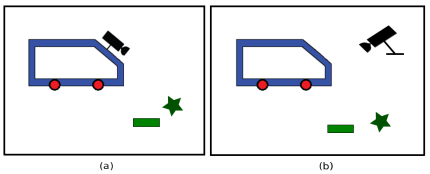

Fig. 1. Connection between APS and on-board (a) and remote (b) camera

The first works and concerning the interaction of control and vision are based on the principle of open control loop. The APC moves, the vision sensor sends back information about its position in the observed scene, then, the APC moves again after changing its position in that same scene. The types of control developed in these studies, due to the simplified task formulation, are effective only in cases where the observed scene is static.

A closed-loop vision control was first used by Shirai and Inoue 1973. Thus, the vision sensor improves the

P. Marinov is with the Department of Computer systems, Technical university of Sofia, (e-mail: pmarinova@tu-sofia.bg)

Received: 24.10.2024
Published: 09.12.2024
https://doi.org/10.47978/TUS.2024.74.04.019

positioning of the APS. Other methods then emerged, which can be divided into four main categories, depending on the use of visual information and the type of command. The first criterion, based on the measurement in the image, makes it possible to distinguish the techniques known as 3D visual or situational control (position-based) and the techniques based on image estimation or 2D estimation (image-based control). The other criterion makes it possible to distinguish the so-called indirect (dynamic look and move) and direct (direct visual servo) visual evaluations. As a summary, all visual management techniques can be classified into three groups according to three criteria:

The table in Fig. 2 Classification based on APS measurement and management. Hybrid visual controls are typically hybrid in their measurement (2D and 3D) rather than their control.

Criteria	Direct control	Indirect control
2D measure	Direct 2D visual assessment (Fig. 6)	Indirect 2D visual assessment (Fig. 4)
3D measure	Direct 3D visual assessment (Fig. 5)	Indirect 3D visual assessment (Fig. 3)
Hybrid	Hybrid visual control	

Fig. 2. Classification of visual control methods

III. VISUAL CONTROL

The distinction between indirect and direct visual control is based on the calculation of movement (the first criterion).

A. *Indirect visual control*

The main advantage of indirect visual control techniques is that they allow control of the motion of the APS separated from the image analysis algorithm and its limitation in terms of computation time.

An added advantage is that the APS can accept speed or position instructions in the Cartesian plane, making them easy to execute. Thus, the indirect visual control is stable, simple and adaptable to different types of APS. Its performance depends solely on its design and the delay from the image analysis in the loop. The measurement of the error to be stabilized can be done in Cartesian space or in the image plane.

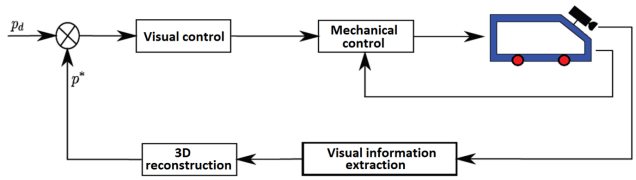

Fig. 3. Indirect 3D control with 3D position p in Cartesian space

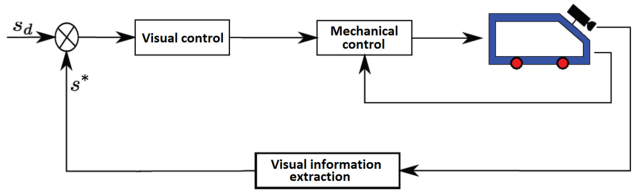

Fig. 1.4 – Indirect 2D control with 2D position p in the image

B. *Direct visual control*

Direct visual control methods differ from indirect visual control techniques in the role of the visual control device. [1] All governing law is done in the same governing device. Here it is necessary to provide an assessment of the condition of the vehicle at high speed. This method is similar to classic automatic control methods.

The increase in computer computing power in recent years has made it possible to have image analysis algorithms fast enough to be embedded in the visual control chain. It also leads to an increase in the frequency of acquisition of images from the cameras and to the acquisition of very high visual rates of the estimates.

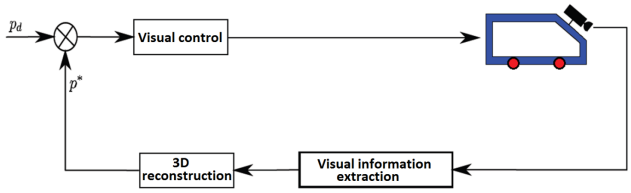

Fig. 5. Direct 3D control with 3D position p in Cartesian space

C. *2D/3D visual control*

The difference between 2D and 3D visual control techniques is the measurement used in the control device circuit.

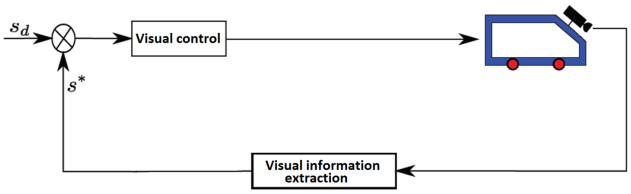

Fig. 6. Direct 2D control with 2D image position

D. *3D visual control*

3D visual control methods are based on 3D measurement, including 3D reconstruction of visual information. [2] [3] A single camera is often used in this approach. In this case, the reconstruction of the target's position (and orientation) requires prior knowledge of the scene geometry. When the 3D system has a stereo head or even more than two cameras, the reconstruction can be performed without any geometric knowledge of the scene.

The geometric elements used in these methods are relatively simple – usually a point or a line. Fig. 5. illustrates the principle of direct 3D visual control. Fig. 3. illustrates indirect 3D visual control. p_d is the setpoint vector representing the desired object position coordinates with respect to the camera, and p* is the measurement vector estimated using a 3D reconstruction algorithm.

Limitation to simple objects and a priori knowledge are significant limitations of 3D methods. More attractive are 2D visual control techniques or hybrid techniques that do not require full 3D reconstruction and are much less error sensitive.

E. *2D visual assessments*

2D visual assessment techniques use 2D measurement. The idea is to no longer use a 3D size reconstructed from visual information, but to directly use the visual information in the 2D image. [4] The goal is to approximate the visual information (geometrical features) $s*$ measured in the image to the desired visual information s_d.

A major advantage is the small amount of information – only depth and camera calibration parameters – and in most

cases a rough estimate works very well. Model-free visual control methods in which no parameter is known in advance have also been developed. They are based on online identification of the interaction matrix or on various types of optimization.

The disadvantages of model-free techniques are the convergence of the optimization and the computation time of that optimization.

Another drawback of 2D visual control is the problem called advance/retreat. It is due to the restriction for 2D visual control that the trajectories of the elements in the image are straight lines (Fig. 7.). The current camera position corresponds to points A, B, C, D in the image, and the desired position corresponds to points A*, B*, C*, D*. The dots move following the arrows, but such dot movement corresponds to pulling the camera back along its optical axis, in theory to infinity. Hybrid methods are designed to avoid this problem.

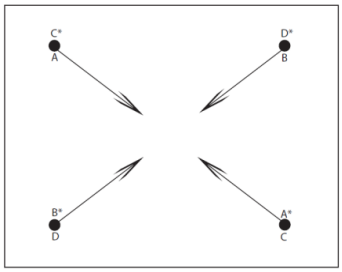

Fig. 7. The forward/backward problem

Fig. 1.6 illustrates the principle of direct 2D visual control. Figure 1.4 illustrates indirect 2D visual serving. In both cases, s_d is the set point vector representing the desired visual information in the image, and s* is the measurement vector of the visual information estimated using an image analysis algorithm.

F. *Hybrid visual control*

Some hybrid visual control techniques are useful for the visual control of autonomous vehicles ideas considered here.

2D visual assessment ½

2D visual control 1/2, [5] is a hybrid technique based on measurement information and commands defined directly in the image and in the camera reference frame. It is used in tasks where the camera is very far from the desired position [MCB98].

Fig. 8. represents a 2D 1/2 visual control, where r_d and t_d are the desired rotation and translation vectors, and r^* and t^* are the measured rotation and translation vectors. The main advantage of this technique is that it requires very little a priori information compared to classical 3D techniques. In fact, convergence is ensured without knowledge of the 3D model of the object and only with an approximation of the desired depth of the object.

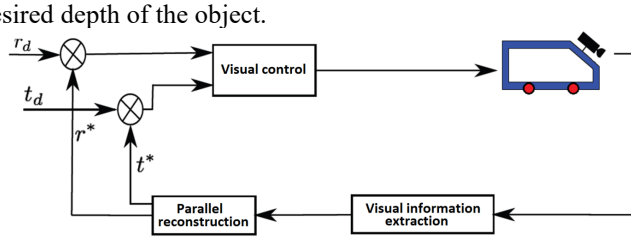

Fig. 8. 2D assessment ½

d2D/dt visual assessment

2D, 3D or 2D 1/2 visual control techniques are based on the use of visual information such as geometric elements and reference measurement of the position of the latter in the image. This requires robust and accurate algorithms for extracting and tracking geometric elements and their presence in the image. To circumvent these requirements, visual d2D/dt dynamic visual control, which is based on the speed of motion in the image, can be used. The command to move the APS is determined by the correspondence of the speed s_d of the measured movement to the speed s∗ of the desired movement (Fig. 9.).

In this case, the measurement of visual information is dynamic and used directly in the control law. [6] Reconstructed visual information is also used in the construction of the control law [QGS95, CAD95, SVS97], but this technique approaches classical 2D visual methods.

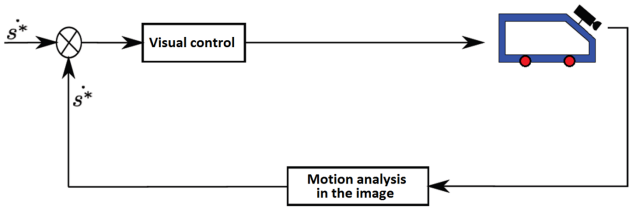

Fig. 9. Block diagram of d2D/dt control

G. *Summarizing*

Various visual control techniques were proposed here. These systems have the advantage of allowing many constraints to be met, but are often financially prohibitive for large-scale commercialization.

Rather, the practical approach is to offer a complete visual control method that is easy to implement and configure and does not require the deployment of very precise devices.

An important criterion for selecting a visual control technique is the mechanical simplicity of the system to be controlled. Unless necessary, the use of complex visual control techniques using strong control-level constraints (trajectory generation, guarantee that the object remains in the visual field, forward/backward problem, etc.) as hybrid techniques is not justified for management.

On the other hand, most of the manufacturers of control devices are not interested in the complete chain of processing of the incoming information. In other words, the topics of image analysis and control are rarely considered together to achieve the coupling of control with vision. Generally, the focus is on one of two aspects of this chain, namely image analysis or control. Reflecting on both the control approach and image analysis allowed for greater integration of these two areas in the visual control chain. This would make it possible to overcome certain time constraints present in many visual control techniques.

An image analysis algorithm based on a global/local approach is discussed here. Methods for estimating the position of the object in the image are proposed.

IV. MOTION ESTIMATION IN THE IMAGE

Whatever the scheme used by each of the visual control methods, one thing is common - the need to work in real time, that is, to be able to update the control of the executive mechanisms in a very short time compared to the speed of

movement of the APS. In the case of visual control, the strongest time constraint usually concerns the image analysis algorithm. When the application has markers on objects of interest, the image analysis algorithm becomes quite simple. Conversely, when the target application allows the user to select the extracted visual information, the image analysis algorithm is more complex and requires more intensive computations.

The most critical part of visual processing techniques is the extraction of motion in an image. It needs to be stable and precise while keeping the time limit.

Except for visual techniques based on dynamic measurements (velocity in the image), the majority of visual systems use visual information from geometric elements (position of points, lines). There are two main approaches to measuring this geometric visual information:

– The first approach consists in estimating the optical flow in the image. These methods are called motion analysis techniques. Optic flow is defined as the apparent 2D motion in an image given by spatiotemporal variations in light intensity. From the apparent velocity in the optical flow image, it is possible to find the desired geometric information. For example, in the case of a point, it is easy to find the position of the point by its original location.

– The second main approach is to extract geometric shapes and trace them image by image. These are the methods for extracting geometric shapes. In these methods, the first phase consists of extracting the geometric shapes from the image using a detector. The second phase consists of finding them in the next image using matching. Unlike previous methods, it does not estimate the motion in the image, but only the motion of the geometric shape extracted in the first place.

A. Motion estimation by image motion analysis

This method estimates the field of apparent velocity vectors in the image. Motion analysis algorithms start from the hypothesis that image intensity remains constant during motion or that it varies in a model able way. [7]

This hypothesis leads to the difference equation between the shifted images, i.e. between the images at times t and $t + \delta t$, where $\delta t = +/- 1$. The equation is called the difference equation in the shifted frame and is written as follows:

$$DFD = I(x + d_x, y + d_y, t + \delta t) - I(x, y, t) = 0 \quad (0)$$

the light intensity at point $p = (x,y)$ at time t and $d = (dx,dy)$ the displacement vector of point p between times t and $t + \delta t$ in the image plane. In the case of the direct direction, for the images $I(t)$ and $I(t + 1)$ at times t and $t + 1$, the evaluation reduces to finding the vector: $d(x,y,t) = (dx(x,y,t),dy(x,y,t))$ for the point $p = (x,y)$ using the two images related by the hypothesis of conservation of light intensity: $I(x,y,t) = I(x + dx,y + dy,t + 1)$ in the case of inverse estimation, the displacement vector is estimated using the relation: $I(x,y,t) = I(x - dx,y - dy,t + 1)$ Fig 10 summarizes

Fig. 10. Transformation of spatial coordinates

In fact, the movement in the image is not directly observed, but its consequence on the intensity of light in the image. In the context of motion estimation, mathematically the task is ill-posed. It has no unique solution, and when one of two matching points is not visible in the image there is no solution.

In motion analysis methods, the estimation of motion in an image is based on spatial and temporal gradients of light intensity. In the case of an image that is discrete information, the gradients are approximated by finite differences.

B. Constraint equation of apparent motion

Constraint equation of apparent motion

The method relies on the apparent motion constraint equation (MCE) for motion in the image. From the PIR equation and under the assumption that the light intensity is constant, we can write for one image pixel $p = (x,y)$:

$$\frac{dI(x,y,t)}{dt} = 0 \quad (1)$$

where x and y vary with time. By differentiating Eq. 1 [8] yields:

$$I(x + d_x, y + d_y, t + \delta t) = I(x,y,t) + \frac{\partial I(x,y,t)}{\partial x} d_x + \frac{\partial I(x,y,t)}{\partial y} d_y + \frac{\partial I(x,y,t)}{\partial t} \delta t \quad (2)$$

by substituting Eq. (2) in Eq. (0) the difference DFD is:

$$DFD(x,y,t) = \frac{\partial I(x,y,t)}{\partial x} d_x + \frac{\partial I(x,y,t)}{\partial y} d_y + \frac{\partial I(x,y,t)}{\partial t} \delta t = 0 \quad (3)$$

and after dividing Eq. (3) of δt, the apparent motion constraint equation (MCE) also called the optical flow equation is obtained:

$$\frac{\partial I(x,y,t)}{\partial x} v_x(x,y,t) + \frac{\partial I(x,y,t)}{\partial y} v_y(x,y,t) + \frac{\partial I(x,y,t)}{\partial t} = 0 \quad (4)$$

where $v_x(x,y,t) = dx/dt$ and $v_y(x,y,t) = dy/dt$ are the x and y velocity components, respectively. In matrix form, for all pixels in the image Eq. (2.4) has the form:

$$(\nabla I) v + I_t = 0 \quad (5)$$

where $\nabla I = (\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})^T$ is the spatial gradient of the image and It is the temporal gradient and $v = (v_x,v_y)^T$ is the transposed velocity vector.

It can be assumed that the intensity along the motion trajectory changes slowly enough and that the spatial gradient along the motion direction is constant:

$$\frac{d\nabla I}{dt} = 0 \quad (6)$$

but requires that distortions and rotations in the image be negligible. Eq. 6 takes the following form:

$$\begin{pmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial y^2} \end{pmatrix} v + \frac{\partial(\nabla I)}{\partial t} = 0 \quad (7)$$

where $\frac{\partial^2 I}{\partial x^2}, \frac{\partial^2 I}{\partial x \partial y}, \frac{\partial^2 I}{\partial y^2}$ are the second derivatives in the image. Eq. 7 is difficult to implement because of the high-frequency nature of the second derivative operator.

Parametric method

In the parametric method, a parametric motion model is introduced, which is the additional constraint to estimate the velocity field. Models account for deformations in the image. The most common transformations are affine, projective (linear) and nonlinear (quadratic) Fig. 11. In the affine model, the transformations are in orthographic projection (rotation, square to parallelogram

transformation), while in the linear projective model, the transformations are in perspective projection. In the affine model with equation:

$$v_x = a_1 + a_3x + a_5y \qquad v_y = a_2 + a_4x + a_6y \tag{8}$$

the unknowns x and y are the coordinates and v_x and v_y the pixel velocity. The projective model is described by equations:

$$v_x = \frac{a_1 + a_3x + a_5y}{1 + a_7x + a_8y} \quad v_y = \frac{a_2 + a_4x + a_6y}{1 + a_7x + a_8y} \tag{9}$$

The most realistic model for describing the motion of an object in an image is the quadratic model. It accounts for the basic transformations in the image, but has no specific physical correspondence. It is described by the equations:

$$\begin{aligned} v_x &= a_1 + a_3x + a_5y + a_7x^2 + a_9xy + a_{11}y^2 \\ v_y &= a_2 + a_4x + a_6y + a_8x^2 + a_{10}xy + a_{12}x^2 \end{aligned} \tag{10}$$

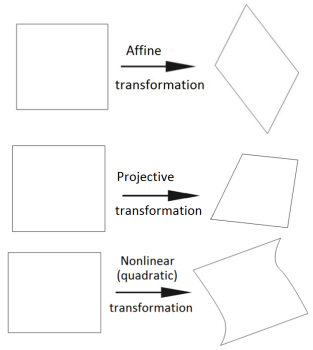

Fig. 11 – Spatial coordinate transformations

Differential method

The differential method of Horn and Shunck [9] [10] is well suited for small displacements and is based on Taylor's order. Its goal is to find the vector field that satisfies an apparent motion constraint equation in each of the image pixels. Let J_{flux} be the error with respect to the equation of the pixel $p(x,y,t)$ of the image:

$$J_{flux} = (\nabla I)v + I_t \tag{11}$$

All neighbouring pixels are assumed to have similar motion. This uses the $J_{uniformite}$ error, which is low if the velocity vector field is smooth.

The method iteratively minimizes the energy $J(v)$:

$$J^2_{uniformite} = \|\nabla v_x\|^2 + \|\nabla v_y\|^2 \tag{12}$$

with λ weighting factor. To minimize the error, the Euler-Lagrange equations are used. Convergence occurs when the error is less than a selected threshold or when a maximum number of iterations is reached.

V. Motion Estimation by Tracking Geometric Elements

In the first image, simple geometric elements (points, lines, segments, contours) are extracted and then followed frame by frame in the rest of the sequence. Tracking is done by matching them in two consecutive images under a set criterion. Some methods use the equation of motion constraint to track geometric elements and their operation is somewhat similar to the methods explained above. [11]

The main advantage of these methods is their simplicity and speed. But they depend a lot on the extraction density of the geometric elements. Furthermore, they are poorly effective in occlusion.

Extract points in an image

Here it involves only the extraction and mapping of points and geometric shapes from points in an image, because this is an element that occurs in all real objects. There are three main classes of point detectors in the literature:

– The first class is contour based. [12] [13] CSS detector is based on edge detection from which points of interest are extracted.

– The second class uses pattern correlation using circular masks with SUSAN detector. [14] Each point in the image is scanned with the circular mask and the pixels included in the mask having a value close to that of the center pixel are counted. Depending on a set threshold, it is determined whether a point of interest is detected or not.

– The third class is based on direct measurement in the image. This one is the most used. It is more robust than the other two classes and requires neither contour extraction nor approximate knowledge of interest points. The following is a brief presentation of some point detectors belonging to the third class.

One such detector, invariant under rotation and translation, uses the second derivatives of an I Beaudet image. [15] [16] If H is the Hessian of an image I at point p with coordinates (x,y):

$$H = \begin{pmatrix} \frac{\partial^2 I(p)}{\partial x^2} & \frac{\partial^2 I(p)}{\partial x \partial y} \\ \frac{\partial^2 I(p)}{\partial y \partial x} & \frac{\partial^2 I(p)}{\partial y^2} \end{pmatrix} \tag{13}$$

with $\frac{\partial^2 I}{\partial x^2}, \frac{\partial^2 I}{\partial x \partial y}, \frac{\partial^2 I}{\partial y^2}$ second derivatives, then the detector is the determinant of H multiplied by a constant C

$$k = C\left(\frac{\partial^2 I}{\partial x^2}\frac{\partial^2 I}{\partial y^2} - \frac{\partial^2 I}{\partial x \partial y}\right)^2 \tag{14}$$

A point p is considered a point of interest when the absolute value of k is greater than a threshold fixed empirically.

The KR detector may be based on the curvature of the image surface multiplied by the gradient norm at the point $p = (x,y)$ [17]:

$$RK = \frac{\frac{\partial^2 I(p)}{\partial x^2}\frac{\partial I(p)}{\partial x}^2 + \frac{\partial^2 I(p)}{\partial y^2}\frac{\partial I(p)}{\partial x}^2 - 2\frac{\partial^2 I(p)}{\partial xy}\frac{\partial I(p)}{\partial x}\frac{\partial I(p)}{\partial y}}{\frac{\partial I(p)}{\partial x}^2 + \frac{\partial I(p)}{\partial y}^2} \tag{15}$$

The detector is built based on the autocorrelation of the Gaussian smoothed image. [18] The autocorrelation is calculated over a window W of a certain size. Let $p = (x,y)$ be a point in image I, then the Harris detector is:

$$M(p) = G(\sigma) * \sum_{p \in W} \begin{pmatrix} \frac{\partial I(p)}{\partial x} & \frac{\partial I(p)}{\partial x}\frac{\partial I(p)}{\partial y} \\ \frac{\partial I(p)}{\partial x}\frac{\partial I(p)}{\partial y} & \frac{\partial I(p)}{\partial y} \end{pmatrix} \tag{16}$$

where $G(\sigma)$ is the Gaussian variance and are the spatial gradients of image I at point p. The eigenvalues $[\lambda_1,\lambda_2]$ of the matrix $M(p)$ allow to distinguish whether the point $p = (x,y)$ is a point of interest or not.

– If λ_1 and λ_2 are small compared to a threshold fixed according to the image, the region under consideration has a constant intensity.

– If $\lambda_1 >> \lambda_2$, the domain has a contour.

– If λ_1 and λ_2 are large compared to the threshold, the region has a corner and therefore a point of interest.

The Harris detector best meets two main Schmid criteria

[20]: repeatability (equal number of points detected during a change in scene illumination) and information content (detected points being different from each other) can be concluded, that

Characterization of extracted points

The problem of finding invariants in an image consists of searching for quantities characteristic of the image, regardless of viewpoint, lighting conditions, geometric transformations, scale changes, etc. All the detectors presented in the previous paragraph are invariant to rotations and translations in the image.

There are geometric invariants using the geometry of a set of points (Euclidean distance between two points or angle between three points). There are differential invariants accounting for the point intensity value and its derivatives., Lowe defined a SIFT (Scale Invariant Feature Transform) [19] feature invariant descriptor that is based on the gradients in eight directions of the pixels near the point of interest and yields, for each point of interest, a descriptor of dimension 128 .It is invariant to changes in illumination and geometric transformations.

Dot matching

Point matching consists of finding the primitives extracted in one image in the next image using a similarity criterion. This correspondence is used to estimate the transformation (displacement in the case of visual processing) between the two images.

The simplest criteria for measuring similarity is a window correlation criterion around the point of interest. For example, SSD represents the sum of the squared differences between the parts corresponding to the two windows. The SAD represents the sum of the absolute values of the differences between the corresponding parts. The transformation between the two images is calculated using the previous correlation criteria. It is based on the assumption that the offset between the images is small (a realistic assumption in the case of a video sequence). We can then consider that if a point of interest is in image $I(t)$ at position (x,y), then that point in image $I(t + 1)$ will be found near (x,y). And therefore the correlation peak will give the location of the corresponding point in the second image.

There are other methods for computing transformations such as relaxation techniques or techniques involving graph theory.

Motion estimation method

As part of an APS visual control system, motion estimation must be performed with a fast, accurate and robust image analysis algorithm. In the previous paragraph, a brief overview of the different classes of possible methods for performing the visual control task of APS was given.

Different methods are described and compared with the constraints of the APS visual control task. As a direct consequence of this, an image analysis algorithm based on a combination of motion and geometric analysis methods is proposed that is robust to disturbances while maintaining good target position precision.

For this purpose, we start from two existing algorithms, chosen for their good properties for each of the two classes of methods. The proposed algorithm is based on the KLT algorithm [21] for point feature extraction and the RMRm algorithm for image motion analysis. This approach can be compared to global/local search approaches. In fact, in the proposed algorithm, the motion analysis behaves as a global motion search in the image, and the tracking of geometric elements as a local refinement of the target search. The developed algorithm is presented below.

VI. Pyramid analysis

A. Analysis with reduced resolution

A sequence of images of the same scene but at different resolution levels is analyzed. To generate the series, the original full-resolution image is convolved (shrunk) with a low-pass filter. This operation is similar to defocusing the camera observing the scene. Filtering selects the best

$$G(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{x^2}{2\sigma^2}} \quad \text{and} \quad G(y) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{y^2}{2\sigma^2}} \tag{17}$$

Here, σ is the Gaussian variance, allowing the filter bandwidth to be adjusted.

Since the image is a discrete signal, it is necessary to make a discrete approximation of $G(x)$ and $G(y)$ as multidimensional vectors to perform the convolution. For speed, sequentially shrink the image with the two filters $G(x)$ and $G(y)$. With variance $\sigma = 1$, the size is $p = 4$ with a filter:

$$G(x,y) = G(x)G(y) = \frac{1}{246}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} \tag{18}$$

$$G(x) = \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix}, \quad G(y) = \begin{bmatrix} 1 \\ 4 \\ 6 \\ 4 \\ 1 \end{bmatrix}$$

where 1/246 is the normalization factor (the sum of the kernel factors).

A multi-resolution representation is therefore a series of images representing the same captured scene, but at progressively lower levels of resolution by applying a low-pass filter.

B. Pyramidal structure

The filtering operation allows to subsample an image without losing information. This approach leads to an image representation with a pyramidal structure with n levels (Fig. 11).

The base of the pyramid contains the original image, which comes from a camera, for example. Each subsequent level of the pyramid contains a lower resolution image than the previous one representing a subsampling of the previous image by a factor of N along each axis. It is obtained by filtering the image from the previous level (Fig. 12) Then the image is reduced by a factor N keeping only one pixel of N^2 to obtain a smaller image. In practice, the filtering and subsampling operations are performed simultaneously.

The pyramidal structure has several advantages:

– The image dimensions of each level of the pyramid as a sub-sample are divided by N compared to the previous level, making the pyramid representation a very compact structure. If $N = 2$, it requires only 33% more memory than the full resolution image alone.

– Filtering is quick and easy. Each level is generated by linearly filtering the previous level. The same filter can be reused because the previous frame has already been

reduced, which shifts the effective cutoff frequency. A small filter can be applied level by level, which makes building a pyramid very fast.

In the considered estimation algorithm is used a pyramid structure with a v-shaped estimation scheme (Fig. 11). It is generated using a sequence of images captured by a camera. A pyramid of this type is used in the KLT and RMRm algorithms. In Fig. 12 at each level of the pyramid the factor is 2 in each direction.

The pyramidal approach eliminates noise by reducing the scale and resolution of the image. Moreover, the coarse-resolution image does not contain the small details of the original image, which improves the convergence of the algorithm. For example, for a large inter-image offset, the pyramidal structure makes it possible to initialize the estimate at a smaller scale level and therefore with reduced inter-image offset.

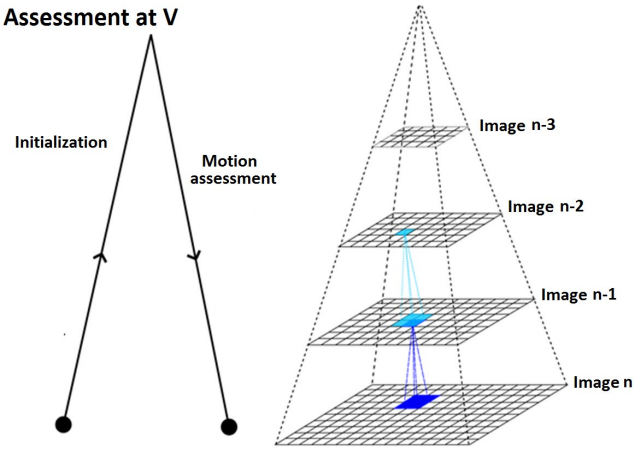

Fig. 2.8 11 Pyramid representation

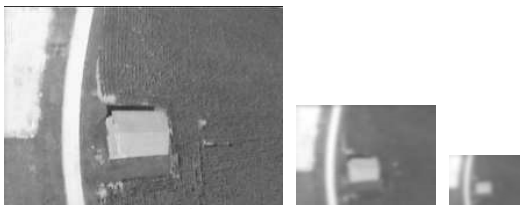

Fig. 12 – Gaussian pyramid

C. KLT algorithm

For extracting points in the image and tracking these points, the KLT algorithm is very effective. First, the general idea of the algorithm is discussed and then its specific implementation.

General idea

Given the presence of noise, motion model approximations, and the need for more than one equation per pixel, the displacement of several adjacent pixels described by a single model is considered to solve for motion. The feature points P that are tracked do not represent a pixel, but the center of an analysis window W containing a set of pixels p. All feature point extraction and observation calculations are done in this window.

The first step is to extract the tracking points. The Harris detector is used for this purpose. Let G be the following matrix:

$$G = \sum_{p \in W} \begin{pmatrix} {g_x}^2 & g_x g_y \\ g_x g_y & {g_y}^2 \end{pmatrix} \tag{19}$$

with g_x and g_y being the spatial gradients along x and y at point p. A feature point is considered a good candidate for monitoring if:

$$\min(\lambda_1,\lambda_2) < \lambda.$$

where λ_1 and λ_2 are the eigenvalues of G and λ is a threshold defined according to the observed scene.

The second step is to trace the extracted points in the following image. An image I at times t and $t + \tau$ is given. Under the hypothesis that the light intensity is conserved, then:

$$I(x,y,t+\tau) = I(x + v_x, y + v_y, t) \tag{20}$$

where $v = (v_x,v_y)$ is the velocity of point $P = (x,y)$. Thus, an image at time $t + \tau$ can be obtained by moving any point in the current image at time t, with an appropriate transformation.

The vector $v = (v_x,v_y)$ is a function of both the position of the point P and all motions of the pixels in the analysis window W. An affine model of the amount of motion, taking deformations into account, allows a better representation of the form:

$$v = DP + d \text{ with } D = \begin{pmatrix} d_{xx} & d_{xy} \\ d_{yx} & d_{yy} \end{pmatrix} \text{ and } d = (d_x, d_y)^T \tag{21}$$

D is the deformation matrix and d the translation of the center of the analysis window defined by the point P. The displacement of P between the original image at time t and the next image at time $t + \tau$ is:

$$I((1+D)P + d, t + \tau) = I(P,t) \tag{22}$$

where I is the 2x2 identity matrix.

The estimation of the motion vector v is reduced to an optimization problem of finding the elements of the matrix D and the coordinates of the vector d minimizing the residual for the analysis window:

$$E = \sum_{p \in W} [I(p + v, t + \tau) - I(p,t)]^2 \tag{23}$$

where W is the analysis window, p is a pixel of that window. We can approximate the Light Intensity Model to be approximated by the first-order summation of Taylor's series:

$$I((p + v), t + \tau) = \mathrm{I}(p, t + \tau) + \nabla I(p, t + \tau)^T v \tag{24}$$

Where $\nabla I(p, t + \tau)^T \cdot v = (g_x, g_y)$ represents the spatial gradient of the image computed at point p. The residual is:

$$E = \sum_{p \in W} [I(p, t + \tau) + \nabla I(p, t + \tau)^T . v_k - I(p,t)]^2 \tag{25}$$

The minimization of E is implemented iteratively with the estimate of the velocity v_k at iteration k as follows:

$$Tz_k = e \tag{26}$$

and gives a 6x6 linear system to solve. Here

$$z_k = \begin{pmatrix} d_{xx} \\ d_{yx} \\ d_{xy} \\ d_{yy} \\ d_x \\ d_y \end{pmatrix}, \quad e = \sum_W [I(p,t) - I(p, t + \tau)] \begin{pmatrix} xg_x \\ xg_y \\ yg_x \\ yg_y \\ g_x \\ g_y \end{pmatrix}$$

and

$$T = \sum_W \begin{pmatrix} U & V \\ V^T & G \end{pmatrix} \text{ where } V = \begin{pmatrix} xg_x^2 & xg_x g_y & yg_x^2 & yg_x g_y \\ xg_x g_y & xg_y^2 & yg_x g_y & yg_y^2 \end{pmatrix}$$

$$U = \begin{pmatrix} x^2 g_x^2 & x^2 g_x g_y & x y g_x^2 & x y g_x g_y \\ x^2 g_x g_y & x^2 {g_y}^2 & x y g_x g_y & x y g_y^2 \\ x y g_y^2 & x y g_x g_y & y^2 g_x^2 & y^2 g_x g_y \\ x y g_x g_y & x y g_y^2 & y^2 g_x g_y & y^2 g_y^2 \end{pmatrix} \text{ and } G = \begin{pmatrix} {g_x}^2 & g_x g_y \\ g_x g_y & {g_y}^2 \end{pmatrix}$$

A convergence criterion is introduced for the behavior of the residual E to distinguish between good and bad points. If the criterion is met, the point is considered a point of interest and continues to be tracked, otherwise the point is eliminated.

Actual realization

The actual realization is implemented in three stages:

• Create points and follow the first stage of the general idea of the algorithm.

• Tracking and is based on the second stage of the main idea of the algorithm, introducing certain simplifications.

• The third stage is checking points to eliminate bad points.

The first stage is described above.

In the second tracking stage, the estimation of the affine deformation matrix D in the analysis window W depends strongly on its size. The limitation of real-time operation requires the use of small window sizes. Additionally, to achieve faster results is to consider only pure translations.

Thus the vector v reduces to:

$$v = d \tag{27}$$

which simplifies the calculation of the residual E:

$$E = \sum_{p \in W} [I(p, t+\tau) + \nabla I(p, t+\tau)^T . d_k - I(p,t)]^2 \tag{28}$$

where d_k is the estimate of vector d at iteration k. Which gives a 2x2 matrix G and to solve a linear system of two equations with two unknowns for each iteration k:

$$G d_k = e \tag{29}$$

$$G = \sum_{p \in W} \begin{pmatrix} {g_x}^2 & g_x g_y \\ g_x g_y & {g_y}^2 \end{pmatrix} \text{ and } e = \sum_{p \in W} [I(p,t) - I(p, t+\tau)] \begin{pmatrix} g_x \\ g_y \end{pmatrix}$$

The residual E is minimized with the iterative Newton-Raphson method.

In the third stage when checking points, two criteria are used to eliminate bad points, which are applied during observation:

– One criterion uses convergence: if Newton-Raphson does not converge after a maximum number of iterations, then the point is eliminated.

– The second criterion uses the average value of the intensity difference between the two calculation windows: if the average value is greater than a fixed threshold, the point is eliminated.

At the end of the third stage, Gaussian pyramid construction is applied to each tracking image with different resolutions. The division factor is 2, i.e. the size of an image at level n corresponds to dividing by two the size of the image at level n-1. The evaluation starts at the lowest resolution (highest level of the pyramid) and then projects to the next level until the highest resolution is reached. The pyramidal structure allows the Newton-Raphson method to be initialized at a coarser, less noisy level to obtain better performance. This also makes it possible to calculate larger displacements.

Fig. 14 illustrates the KLT algorithm. Here, the area spanning the points of interest simply corresponds to the area of the image to be tracked, and the magnification of this area showing the different computational windows W and their centers: the points of interest P.

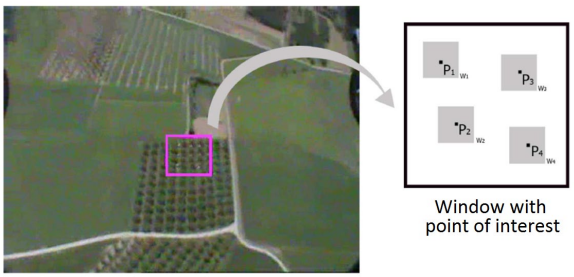

Fig. 13 Illustration of the KLT algorithm

RMRm algorithm

The RMRm algorithm is used to estimate the global motion pattern in the image. It uses the following affine 2D model:

$$v_x = a_1 + a_2 x_i + a_3 y_i \qquad v_y = a_4 + a_5 x_i + a_6 y_i \tag{30}$$

where $p = (x,y)$ is the position of a point in the image, $(a_1,...,a_6)$ are the model coefficients and (v_x,v_y) is the velocity vector at point p. To obtain the coefficients of the model, the equation of the apparent restriction of motion for each point p is solved:

$$\frac{dI}{dt}(p,t) = V(p,t) \nabla I(p,t) + I_t(p,t) = 0 \tag{31}$$

where $\nabla I(p_i,t)$ is the spatial gradient at point p_i, $I_t(p_i,t)$ is the temporal gradient at point p_i, and $V(p,t) = (v_x,v_y)$.

In general, the RMRm algorithm uses two consecutive images obtained from the camera and consists of two steps. In the first step, two Gaussian pyramids are constructed. The separation factor between each level is two. The image size at level n corresponds to half the image size at level $n - 1$. This allows the estimation of large displacements. In the second step, the assessment itself is performed. Let θ_t be the six-parameter vector of the affine motion model at time t to be estimated. The first evaluation of this vector is done at the top of the pyramid (lowest resolution). It is minimized with respect to θ_t by the following criterion:

$$C(\theta_t) = \sum_{p \in S} \rho(r(p, \theta_t)) \tag{32}$$

where

$$r(p, \theta_t) = \nabla I(p,t) . V_{\theta_t}(p) + I_t(p,t)$$

with p the set of points in the image, With the evaluation support, $\nabla I(p,t)$ the spatial gradient of I at point p, $I_t(p,t)$ the time derivative of I at point p, $V_{\theta t}$ the velocity vector computed at point p by applying the model defined by the current estimate value of θ_t and ρ a robust estimator. In this case, ρ is the two-weighted Tukey loss function [22]. This estimator allows during the iterative process to weigh the adequacy of the motion of point p with the current estimate of the motion model θ_t. The weaker this adequacy, the closer the weight is to zero. This estimator allows discarding points whose motion is abnormal with respect to the general motion and therefore makes the estimation more robust. Given the nonlinearity of $C(\theta_t)$, a process based on successive approximations of $C(\theta_t)$ is used. The estimate of θ_t at iteration k is:

$$\widehat{\theta_t^k} = \widehat{\theta_t^{k-1}} + \widehat{\Delta\theta_t^k} \tag{33}$$

where $\Delta\theta_t^k$ represents the refinement given by the estimate at iteration k. Let pk be the estimate of the position of point

p at iteration *k*. Each increment is obtained by minimizing the following error with respect to $\Delta\theta_t^k$:

$$D(\Delta\theta_t^k) = \sum_{p \in S} \rho(r'(p, \Delta\theta_t^k)) \tag{34}$$

The process is repeated and increments are accumulated until a predefined convergence criterion is reached. The estimate at the finest resolution level is initialized from the value obtained at the coarser resolution level. The RMRm algorithm uses an iterative weighted least squares estimation process. This requires only the calculation of the derivatives of the spatiotemporal intensity function.

The RMRm algorithm takes into account variations in light intensity and Eq. 31 takes the form:

$$\frac{dI}{dt}(p,t) = V(p,t)\nabla I(p,t) + I_t(p,t) = -\xi \tag{35}$$

where ξ is a scalar representing the change in light intensity to be estimated.

VII. CONCLUSION

In summary, the techniques used relate to the study of visual control, i.e. for the management of a closed system, thanks to visual information extracted from the images received by the camera using a suitable image analysis algorithm.

In order to perform visual control tasks of APS, a robust motion estimation is required, taking into account the quality of the processed images and the required high precision due to the size of the characteristic objects in the images. For this purpose, it is proposed to combine two motion estimation methods: a method for stable dominant motion estimation using a parametric model and a method based on the extraction and tracking of characteristic points in the image. The two methods are based on the two algorithms summarized above, KLT for point tracking and RMRm for motion model estimation.

REFERENCES

[1] Peter I. Corke, Visual control of robots. High-Performance Visual Servoing, CSIRO Division of Manufacturing Technology, Australia, University of Southern Queensland, Toowoomba, QLD4350 378 p.)

[2] Y. Shirai, H. Inoue, Guiding a robot by visual feedback in assembling tasks, Pattern Recognition, 1973, vol. 5, pp. 99-108.

[3] N.P. Papanikopoulos, P.K. Khosla, T. Kanade, Visual tracking of a moving target by a camera mounted on a robot: A combination of vision and control, IEEE Trans. Robot. and Automation, 1993, vol. 9, no. pp. 14-35.

[4] Motion illusions as optimal percepts, Yair Weiss, Eero P. Simoncelli and Edward H. Adelson, nature neuroscience volume 5, no 6, june 2002, p. 598-604,DOI: 10.1038/nn858).

[5] 2-1/2-D Visual Servoing, E. Malis, F. Chaumette, and S. Boudet, IEEE Tans. on Robotics and Automation, vol. 15, no. 2, April 1999, p. 238-250.

[6] A. Crétual and F. Chaumette, Positioning a camera parallel to a plane using dynamic visual servoing. 1997, In Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 1, pp. 43–48.

[7] S. Ghuffara, N. Brosch, N. Pfeifera and M. Gelautz, Motion estimation and segmentation in depth and intensity videos, Integrated Computer-Aided Engineering, 2014, vol. 21, 203–218 203 DOI 10.3233/ICA-130456, IOS Press).

[8] Y. Liu, S.F., Lourenco, Perception of Apparent Motion is Constrained by Geometry, not Physics, Journal of Vision, 2019, vol.19,10,37b, DOI:10.1167/19.10.37b.

[9] B.K.P. Horn, B.G. Schunck, Determining optical flow, Artificial Intelligence, 1981, 17(1), 185203.

[10] A. Maykol, G. Pinto, A. P. Moreira, P. G. Costa, M. V. Correia, Revisiting Lucas-Kanade and Horn-Schunck, Journal of Computer Engineering and Informatics, Apr. 2013, vol. 1, Iss. 2, PP. 23-29, DOI: 10.5963/JCEI0102001.

[11] P.L. Bazin, J.M. Vezien, A. Gagalowicz, Shape and Motion Estimation from Geometric Primitives and Parametric Modelling, MVA2000, IAPR Workshop on Machine Vision Applications, Nov. 28-30,2000, The University of Tokyo, Japan.

[12] F. Mokhtarian, R. Suomela, Robust image corner detection through curvature scale space, IEEE Trans. Pattern Anal. Mach. Intell. 20, 1998, 12, pp. 1376–1381.

[13] F. Mokhtarian, F. Mohanna, Performance evaluation of corner detectors using consistency and accuracy measures, Computer Vision and Image Understanding, 2006, vol. 102, pp. 81–94.

[14] S.M. Smith, J.M. Brady, SUSAN - a new approach to low level image processing, International Journal of Computer Vision, 1997, vol. 23, Iss. 1, pp. 45-78.

[15] P. Beaudet, Rotationally invariant image operators, 1978, Proc. 4th Int. Joint Conference on Pattern Recognition, pp. 579-583

[16] Junqing Wang, Weichuan Zhang, A Survey of Corner Detection Methods, 2nd International Conference on Electrical Engineering and Automation, 2018, ICEEA 2018.

[17] L. Kitchen and A. Rosenfeld, Gray level corner detection, Pattern Recogn. Lett., 1982, vol. 1, pp. 95–102.

[18] C. Harris, M. Stephens, A combined corner and edge detector. In Proceedings of the 4th Alvey Vision Conference, 1988, pp. 147-151.

[19] D.G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision, 2004.

[20] M. Stephen, J.M. Smith, A new approach to low level image processing. International Journal of Computer Vision, 1997, vol. 23, pp. 45-78.

[21] B.D. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th international joint conference on Artificial intelligence, 1981, vol. 2, pp. 674–679, IJCAI'81, San Francisco, CA, USA.

[22] (V. Belagiannis, Ch. Rupprecht, G. Carneiro, and N. Navab, Robust Optimization for Deep Regression, University of Adelaide, Australia, arXiv:1505.06606v2 [cs.CV] 22 Sep 2015.