Mathematical Analysis for Visual Tracking Assuming Perspective Projection

João P. Barreto, Paulo Peixoto, Jorge Batista, Helder Araujo [⋆]

Institute of Systems and Robotics
Dept. of Electrical Engineering
Polo II - University of Coimbra
3030 Coimbra
Portugal

Abstract. Applications of visual control of motion require that the relationships between motion in the scene and image motion be established. In the case of active tracking of moving targets these relationships become more complex due to camera motion. This work derives the position and velocity equations that relate image motion, camera motion and target 3D motion. Perspective projection is assumed. Both monocular and binocular tracking systems are analyzed. The expressions obtained are simplified to decouple the control of the different mechanical degrees of freedom. The simplification errors are quantified and characterized. This study contributes for the understanding of the tracking process, for establishing the control laws of the different camera motions,for deriving test signals to evaluate the system performance and for developing egomotion compensation algorithms.

1 Introduction

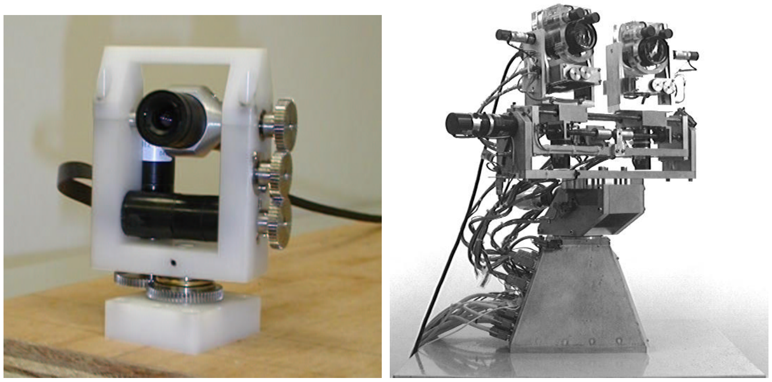


Fig. 1.: The Active Vision Systems at ISR. Right: The modular camera pan and tilt unit MVS. Left: The ISR-MDOF robot head.

Visual control of motion is a widely studied subject. Several papers have addressed the problem of controlling motion using visual information [1, 2, 3, 4]. In these applications, the relationships between image motion and 3D motion in the scene have to be established. In active tracking applications these relationships become more complex due to camera motion. Works like [5] and [6] address mainly the control aspects of tracking. Cameras are modeled as a constant gain and the effects of perspective projection are not considered. Affine models are assumed in [7] and approximations in the perspective model are considered in [8]. Many of these works rely on closed-loop control strategies to reduce the influence of modeling simplification. However some visual behaviors such the saccadic motion typically use open-loop configuration.

[⋆] Email:{jpbar,peixoto,batista,helder}@isr.uc.pt

In this paper we derive the mathematical relationships between image motion, camera motion and target 3D motion in the scene, both for monocular and binocular tracking applications. Perspective projection is considered by assuming a camera pinhole model. Both position and velocity equations are derived. The results obtained allow a better understanding of the tracking process as a regulation control problem. Strategies for both monocular and binocular tracking are developed. The simplifications of the equations derived to decouple the degrees of freedom of the vision system are discussed. Control laws for different visual behaviors (smooth pursuit, vergence and saccade) are established and approximation errors are studied and characterized.

We are using the results of the present work to derive test signals to characterize the performance and robustness of the active tracking algorithms. Egomotion compensation techniques are being studied as well. The knowledge of the relationship between velocity in image, camera motion and targets 3D velocity can also be useful in the development of high-level visual behaviors such as target segmentation in an environment with multiple moving targets.

2 Monocular Tracking

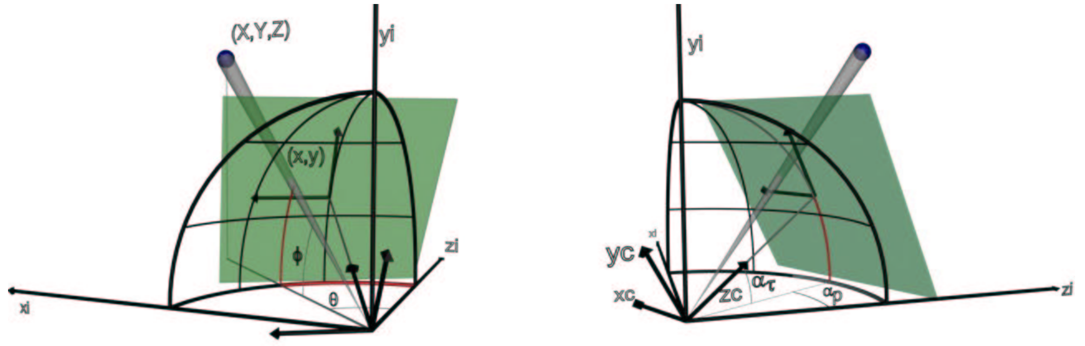


Fig. 2.: Image formation process in monocular tracking. The 3D scheme, exhibiting the variables and reference frames used in mathematical analysis, is shown from two different points of view.

Fig. 1(R) depicts our MVS unit. The camera has two degrees of freedom: pan and tilt. Both rotation axes go through the optical center. Camera undergoes pure rotation motion. The typical goal for a tracking application is to move the camera in such a way that target is projected in the center of image. In 3D space it means that the optical axis must be aligned with the target. This section derives the mathematical relationship between 3D target motion, camera motion and projection in image. Camera motion is typically known by using motor encoders.

The scheme of figure 2 describes the image formation process. Assume a standard pinhole model where camera performs a perfect perspective transform with center O (camera optic center) at a distance f (focal length) of the retinal plane. Vector $\mathbf{M} = (X, Y, Z)^t$ represents target cartesian coordinates in $\Re_i(O, \mathbf{x_i}, \mathbf{y_i}, \mathbf{z_i})$, (inertial reference frame) . Target 3D position can also be represented by spherical coordinates (ρ, θ, ϕ). $\Re_c(O, \mathbf{x_c}, \mathbf{y_c}, \mathbf{z_c})$ is the referential frame attached to the camera where $\mathbf{z_c}$ is aligned with the optical axis and $\mathbf{x_c}$ and $\mathbf{y_c}$ are aligned with horizontal and vertical axes in the image. The retinal plane is perpendicular to the optical axis. Target is projected at point (x, y) in image plane. This point is represented as an homogeneous 3D vector $\lambda\mathbf{m} = \lambda(x, y, 1)$ corresponding to a line of a given direction passing through the optical center (2D projective space). Camera first rotates in pan and then in tilt (Fick model). Camera pan and tilt positions are given by α_p and α_t (the rotation angles around $\mathbf{y}$ and $\mathbf{x_c}$). $\mathbf{R_p}$ and $\mathbf{R_t}$ are the pan and tilt rotation matrices

$$\boldsymbol{A} = \begin{bmatrix} s_x & s_\psi & u_0 \\ 0 & s_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} F & 0 & 0 \\ 0 & F & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

$\mathbf{A}$ is the matrix of the intrinsic parameters in the projection (see equation 1). s_x and s_y stand for the scaling along the horizontal and vertical axes of the image plane, s_ψ gives the skew between the axes and (u_0, v_0) are the principal point coordinates. Notice that $\mathbf{A}$ is an upper triangular matrix, thus it is always invertible. It is assumed the axes are orthogonal ($s_\psi = 0$), the optical axis intersects the retinal plane at the origin of the image referential ($u_0 = v_0 = 0$) and $s_x = s_y = F$.

2.1 Position equations

$$z\boldsymbol{m} = \boldsymbol{A}\boldsymbol{R_t^t}(\boldsymbol{\alpha_t})\boldsymbol{R_p^t}(\boldsymbol{\alpha_p})\boldsymbol{M} \tag{2}$$

Equation 2 expresses the relationship between target 3D coordinates $\mathbf{M}$ in the inertial frame and projective coordinates $\mathbf{m}$ in camera reference frame. Notice z is target Z coordinate in camera referential, $\mathbf{m} = (x, y, 1)^t$ the projective coordinates and (x, y) target coordinates in image plane.

$$\frac{z'}{z}\boldsymbol{R_p}(\boldsymbol{\varDelta\alpha_p})\boldsymbol{R_t}(\boldsymbol{\varDelta\alpha_t})\boldsymbol{R_t}(\boldsymbol{\alpha_t})\boldsymbol{A}^{-1}\boldsymbol{m'} = \boldsymbol{R_t}(\boldsymbol{\alpha_t})\boldsymbol{A}^{-1}\boldsymbol{m} \tag{3}$$

Camera moves $\varDelta\alpha_p$ in pan and $\varDelta\alpha_t$ in tilt. New camera rotation angles are $(\alpha_p + \varDelta\alpha_p, \alpha_t + \varDelta\alpha_t)$. Equation 3 establishes the relationship between $\mathbf{m'}$, the actual target projective coordinates, and $\mathbf{m}$, the projective coordinates before camera motion. Consider $\varDelta\alpha_p$ and $\varDelta\alpha_t$ are the tracking angular position errors in pand and tilt. If target spherical coordinates are (ρ, θ, ϕ) (see Fig. 2) then $\varDelta\alpha_p = \theta - \alpha_p$ and $\varDelta\alpha_t = \phi - \alpha_t$.

$$\boldsymbol{m} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} (FC(\alpha_t) - yS(\alpha_t))\tan(\varDelta\alpha_p) \\ F\frac{2(S(\alpha_t)^2 C(\varDelta\alpha_p) + C(\alpha_t)^2)\tan(\varDelta\alpha_t) - S(2\alpha_t)(C(\varDelta\alpha_p)-1)}{2(S(\alpha_t)^2 + C(\alpha_t)^2 C(\varDelta\alpha_p)) - S(2\alpha_t)(C(\varDelta\alpha_p)-1))\tan(\varDelta\alpha_t)} \\ 1 \end{bmatrix} \tag{4}$$

Whenever camera moves to compensate the angular position errors, the target is projected in the center of the image and $\mathbf{m'}$ becomes equal to $(0, 0, 1)^t$. Equation 4 is derived from 3 making $\mathbf{m'} = (0, 0, 1)^t$. It establishes the relationship between camera tilt position α_t, target position in image (x, y) and angular position errors $(\varDelta\alpha_p, \varDelta\alpha_t)$.

$$\tilde{\boldsymbol{m}} = \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{bmatrix} = \begin{bmatrix} F\cos(\alpha_t)\tan(\varDelta\alpha_p) \\ F\tan(\varDelta\alpha_t) \\ 1 \end{bmatrix} \tag{5}$$

Both x and y expressions are complex and highly non-linear. If the target is being tracked by the active vision system , it is reasonable to assume that most of time the target image is nearly in the center. This assumption is used to derive equation 5, a simplification of 4. Using 5 pan and tilt control can be decoupled. x position is only related with pan error and y position is related with tilt error.

$$\boldsymbol{E_{pos}} = \begin{bmatrix} E^x_{pos} \\ E^y_{pos} \end{bmatrix} = \begin{bmatrix} x - \tilde{x} \\ y - \tilde{y} \end{bmatrix} \tag{6}$$

$$\boldsymbol{\mu_{pos}} = \begin{bmatrix} \mu^x_{pos} \\ \mu^y_{pos} \end{bmatrix} = \sum_{i=1}^{N^2} \boldsymbol{E_{pos}(i)}P(i) \tag{7}$$

$$\boldsymbol{\varPhi_{pos}} = \begin{bmatrix} (\sigma^x_{pos})^2 & \phi^{xy}_{pos} \\ \phi^{xy}_{pos} & (\sigma^y_{pos})^2 \end{bmatrix} = \sum_{i=1}^{N^2} (\boldsymbol{E_{pos}(i)} - \boldsymbol{\mu_{pos}})(\boldsymbol{E_{pos}(i)} - \boldsymbol{\mu_{pos}})^t P(i) \tag{8}$$

Camera angular position (α_p, α_t) is obtained by reading motor encoders. Visual processing determines target position in image (x, y). The goal is to move the camera to compensate for the angular errors $(\varDelta\alpha_p, \varDelta\alpha_t)$. Equation 4 performs the exact computation of $(\varDelta\alpha_p, \varDelta\alpha_t)$ given (x, y). $\mathbf{E_{pos}}$ is the error in approximating equation 4 by 5 (see equation 6). $\mathbf{E_{pos}}$ is a function of α_t, $\varDelta\alpha_p$ and $\varDelta\alpha_t$. Consider $\alpha_t, \varDelta\alpha_p, \varDelta\alpha_t \in [-45\circ, 45\circ]$ and the interval discretized in N samples uniformly spaced. For each camera tilt position α_t there are N^2 possible combinations for $(\varDelta\alpha_p, \varDelta\alpha_t)$. The pan and tilt angular errors are assumed to be statistically independent. $P(i)$ is the joint probability function of $(\varDelta\alpha_p, \varDelta\alpha_t)$. Fixate α_t, for each pair $(\varDelta\alpha_p, \varDelta\alpha_t)$ there is a corresponding error vector $\mathbf{E_{pos}}$. Equation 7 computes the average error μ_{pos} for a certain camera tilt position α_t. Expression 8 computes the covariance matrix $\varPhi_{pos}$.

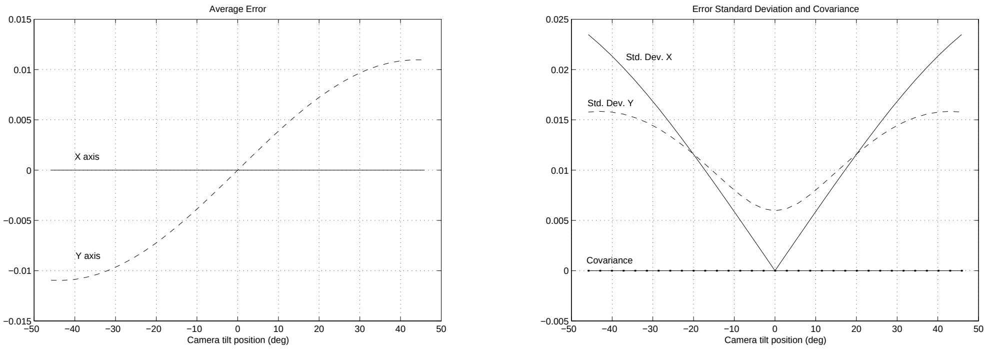


Fig. 3.: Quantitative analysis of error function E^x_{pos} and E^y_{pos} assuming a normal probability distribution for the angular errors in pan and tilt. Above: average error in X (-) and Y (- -). Below: error standard deviation in X (-) and Y (- -).

If the system is tracking the target then, most of time, target image is not far from the center. It is reasonable to assume that both pan and tilt errors ($\Delta\alpha_p$ and $\Delta\alpha_t$) are statistically described by a normal distribution of average 0°. $P(i)$, in equations 7 and 8, is a bidimensional probability density function for a normal distribution with zero average and standard deviation of 12○ in pan and 8○ in tilt. Fig. 3 depicts $\mathbf{E_{pos}}$ average, standard deviation and covariance as a function of α_t. The average error in X and the covariance ϕ^{xy}_{pos} are zero and μ^y_{pos} is an odd function. Both σ^x_{pos} and σ^y_{pos} (standard deviation in X and Y) increase with the module of camera tilt angle.

The data depicted in Fig. 4 is useful to understand the behavior of the error function E_{pos}. Different camera tilt positions were studied. First column is for $\alpha_t = -23°$, the second for $\alpha_t = 0°$ and the third for $\alpha_t = 23°$. Whenever α_t is known target position in image depends on the angular pan and tilt errors $(\Delta\alpha_p, \Delta\alpha_t)$. First row depicts the exact and approximated target positions in image for different tilt angles. The second row exhibits the corresponding error X coordinate E^x_{pos}. The third row shows the error in Y axis E^y_{pos}.

Observe the exact and estimated positions of target projection in image (first row). Assume the angular pan error $\Delta\alpha_p$ constant. In 3D space target is positioned somewhere in a vertical plane going trough the origin O of the inertial referential frame (see Fig. 2). The plane is projected in a line in the image. If $\alpha_t = 0$ the line is vertical and if $\alpha_t \neq 0$ the line has a slope whose module is inversely proportional to the module of camera tilt angle. The approximation of equation 5 always projects the plane in a vertical line in the image. Thus, as depicted in Fig. 4, $\tilde{x} = x$ whenever $\alpha_t = 0$ or $y = 0$. The approximation error in X axis E^x_{pos} is a function of α_t, $\Delta\alpha_p$ and $\Delta\alpha_t$. Notice that:

- $E^x_{pos}(0, \Delta\alpha_p, \Delta\alpha_t) = 0$
- $E^x_{pos}(\alpha_t, \Delta\alpha_p, \Delta\alpha_t) = -E_x(\alpha_t, -\Delta\alpha_p, \Delta\alpha_t)$
- $E^x_{pos}(\alpha_t, \Delta\alpha_p, \Delta\alpha_t) = E_x(-\alpha_t, \Delta\alpha_p, -\Delta\alpha_t)$

Consider the angular tilt error $\Delta\alpha_t$ constant. In 3D space the target is somewhere in a conic surface whose vertice is the origin O of the inertial referential frame (see Fig. 2). The image projection of these surfaces are the hiperbolic lines depicted in Fig. 4 (first row). Whenever $\Delta\alpha_t = -\alpha_t$ the conic surface degenerates in the OXZ plane which is projected in an horizontal line. The approximation of equation 5 generates horizontal lines in image. Therefore $\tilde{y} = y$ whenever $\Delta\alpha_t = -alpha_t$ or $\Delta\alpha_p = 0$. Some properties of E^y_{pos}, observable in Fig. 4, are itemized below:

- $E^y_{pos}(\alpha_t, 0, \Delta\alpha_t) = 0$
- $E^y_{pos}(\alpha_t, \Delta\alpha_p, -\alpha_t) = 0$
- $E^y_{pos}(\alpha_t, \Delta\alpha_p, \Delta\alpha_t) = E_y(\alpha_t, -\Delta\alpha_p, \Delta\alpha_t)$
- $E^y_{pos}(\alpha_t, \Delta\alpha_p, \Delta\alpha_t) = -E_y(-\alpha_t, -\Delta\alpha_p, -\Delta\alpha_t)$

The approximation error $\mathbf{E_{pos}} = [E^x_{pos}, E^y_{pos}]^t$ increases with both angular position errors $(\Delta\alpha_p, \Delta\alpha_t)$ and camera tilt position. If camera tilt angle α_t take a great value the approximation error becomes signif-

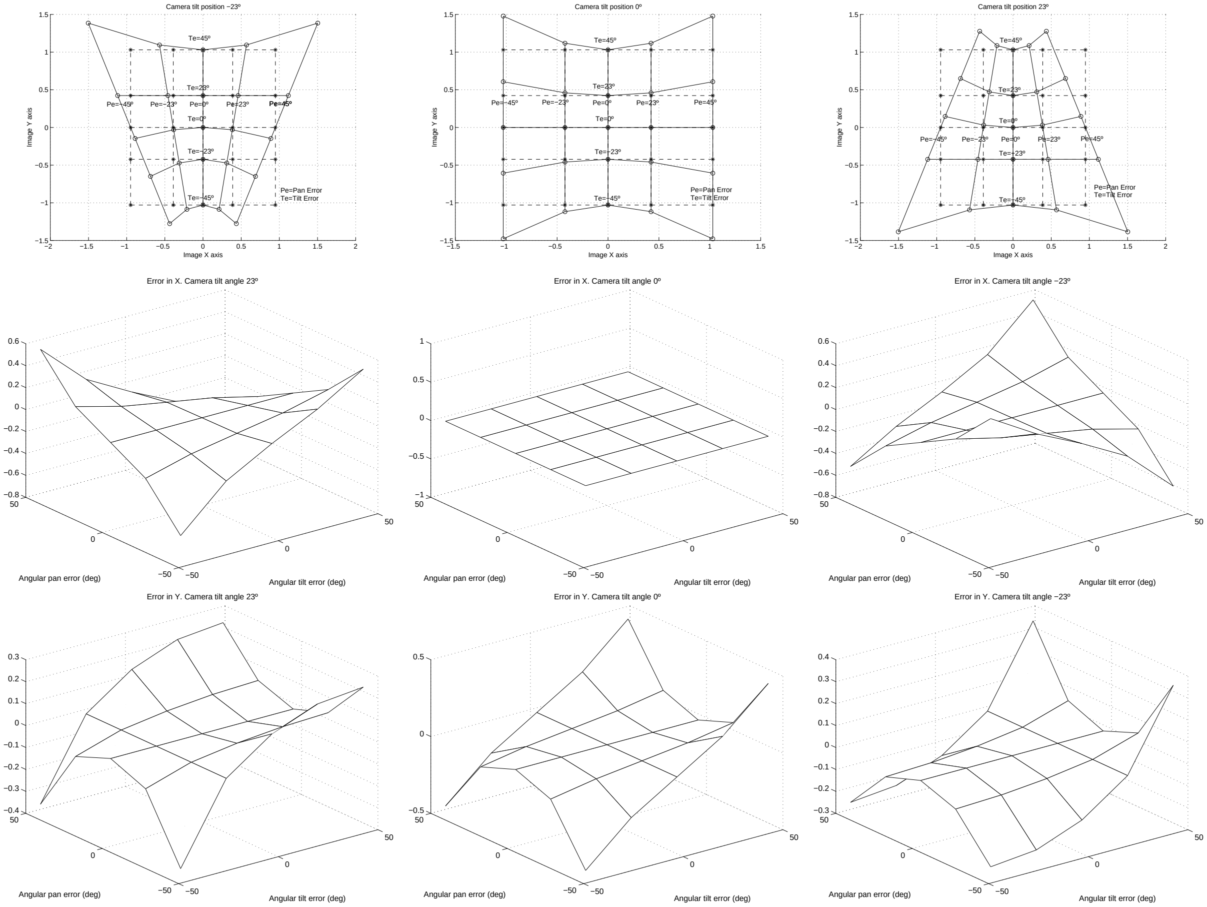


Fig. 4.: Approximation error $\mathbf{E_{pos}}(E^x_{pos}, E^y_{pos})$ assuming $F = 1$. Each column corresponds to a certain camera tilt position: first column $\alpha_t = -23°$, second column $\alpha_t = 0°$ and third column $\alpha_t = 23°$. First row: target projection in image (exact (o) and approximated (*) position). Second row: E^x_{pos} as a function of $(\varDelta\alpha_p, \varDelta\alpha_t)$. Third row: E^y_{pos} as a function of $(\varDelta\alpha_p, \varDelta\alpha_t)$.

icant even if the target is projected near the center of the image. Therefore to approximate equation 4 by equation 5, the operation range of tilt degree of freedom can not be large.

This last result yields important conclusions reagarding the mechanical construction of an active tracking system. We have been assuming a Fick model of rotation (pan and tilt). Our systems have been designed for tracking targets moving in the ground plane (see Fig 1). The pan operating range is much larger than tilt operating range. The values of α_t are always less than 23° (the mechanical limit for the tilt degree of freedom) and the errors in assuming the approximations of equation 5 are small. Consider a system, with the same purposes, designed with the camera first rotating in tilt and then in pan (Helmholtz model). Assuming this α_t would be replaced by α_p in the derived equations. The errors in using the approximations would be more significant because the operation range of α_p would be larger. Moreover the system mechanical construction would involve additional difficulties. It is always easier to achieve large operation ranges in the independent rotation than in the dependent one. For these reasons it is advisable to design the system with the camera rotating first around the angle where larger operation ranges are requested.

2.2 Velocity Equations

This section derives the mathematical expressions for the target velocity in the image. Velocity in the image depends on camera motion and target 3D velocity. Camera motion induces velocity in the image even when the scene is static. This self-induced motion is called in the literature egomotion. Target motion in 3D space also induces motion in image. Tracking is achieved when camera moves in such a way that egomotion cancels out the velocity induced by motion in the scene and target is kept in the same position in successive frames. In this paper it is assumed that camera intrinsic parameters are kept constant along time ($\dot{\mathbf{A}} = 0$).

$$\dot{z}\boldsymbol{m} + z\dot{\boldsymbol{m}} = \boldsymbol{A}\dot{\boldsymbol{R}}_{\boldsymbol{t}}^{\boldsymbol{t}}(\boldsymbol{\alpha_t})\boldsymbol{R}_{\boldsymbol{p}}^{\boldsymbol{t}}(\boldsymbol{\alpha_p})\boldsymbol{M} + \boldsymbol{A}\boldsymbol{R}_{\boldsymbol{t}}^{\boldsymbol{t}}(\boldsymbol{\alpha_t})\dot{\boldsymbol{R}}_{\boldsymbol{p}}^{\boldsymbol{t}}(\boldsymbol{\alpha_p})\boldsymbol{M} + \boldsymbol{A}\boldsymbol{R}_{\boldsymbol{t}}^{\boldsymbol{t}}(\boldsymbol{\alpha_t})\boldsymbol{R}_{\boldsymbol{p}}^{\boldsymbol{t}}(\boldsymbol{\alpha_p})\dot{\boldsymbol{M}} \tag{9}$$

$$\boldsymbol{K} = \begin{bmatrix} 1 \; 0 \; -x \\ 0 \; 1 \; -y \end{bmatrix}. \tag{10}$$

Equation 9 is derived by differentiating 2. $\dot{\mathbf{m}} = [\dot{x}, \dot{y}, 0]^t$ where $[\dot{x}, \dot{y}]^t$ is the velocity vector in image. $\dot{z}$ is target velocity along Z axis in camera reference frame. In order to derive image velocity vector, matrix $\mathbf{K}$ is defined in such a way that $(\dot{x}, \dot{y})^t = \frac{1}{z}\mathbf{K}(\dot{z}\mathbf{m} + z\dot{\mathbf{m}})$.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \boldsymbol{K}\boldsymbol{A}\boldsymbol{P_t}\boldsymbol{A}^{\boldsymbol{-1}}\boldsymbol{m}\dot{\alpha}_t + \boldsymbol{K}\boldsymbol{A}\boldsymbol{R}_{\boldsymbol{t}}^{\boldsymbol{t}}(\boldsymbol{\alpha_t})\boldsymbol{P_p}\boldsymbol{R_t}(\boldsymbol{\alpha_t})\boldsymbol{A}^{\boldsymbol{-1}}\boldsymbol{m}\dot{\alpha}_p + \frac{1}{z}\boldsymbol{K}\boldsymbol{A}\boldsymbol{R}_{\boldsymbol{t}}^{\boldsymbol{t}}(\boldsymbol{\alpha_t})\boldsymbol{R}_{\boldsymbol{p}}^{\boldsymbol{t}}(\boldsymbol{\alpha_p})\dot{\boldsymbol{M}} \tag{11}$$

Equation 11 is derived from 9. $\mathbf{P_p}$ and $\mathbf{P_t}$ are the diferential generators of the abelean groups $\mathbf{R_p^t}()$ and $\mathbf{R_t^t}()$. Velocity in image $(\dot{x}, \dot{y})^t$ both depends on camera velocity of motion $(\dot{\alpha}_p, \dot{\alpha}_t)$ (egomotion) and 3D velocity in the scene $\dot{\mathbf{M}}$. The egomotion terms do not depend on scene 3D coordinates because camera describes pure rotation motion.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = -\boldsymbol{W_{ego}} \begin{bmatrix} \dot{\alpha}_p \\ \dot{\alpha}_t \end{bmatrix} + \boldsymbol{W_{tgt}} \begin{bmatrix} \dot{\theta} \\ \dot{\phi} \end{bmatrix} \tag{12}$$

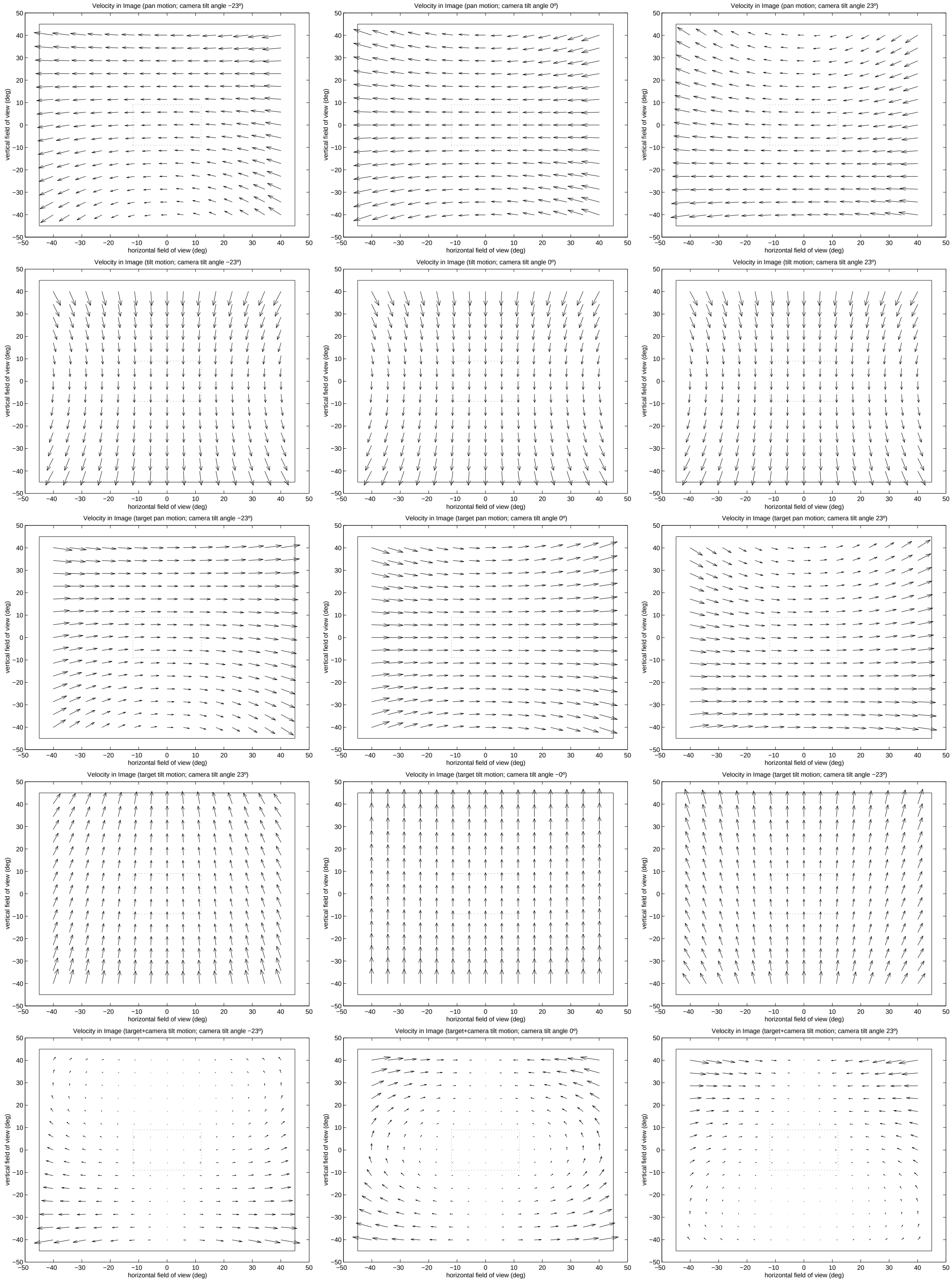
The target velocity in 3D space can be described in rectangular $(\dot{X}, \dot{Y}, \dot{Z})$ or spherical $(\dot{\rho}, \dot{\theta}, \dot{\phi})$ coordinates. When the target motion is described in rectangular coordinates, velocity in the image depends on $\dot{X}$, $\dot{Y}$ and $\dot{Z}$. On the other hand, whenever spherical coordinates are used velocity in the image depends only on $\dot{\theta}$ and $\dot{\phi}$. This happens because $\dot{\rho}$ is the velocity component along the projective ray that can not be observed in the image (for a point target). Spherical coordinates simplify the derivation and understanding of velocity equations. Equation 12 yields the target velocity in the image. The first term refers to egomotion, showing the contribution of camera motion to image velocity. Matrix $\mathbf{W_{ego}}$ is a 2x2 weighting matrix that depends on the image point coordinates and camera tilt angle. The second term refers to the velocity in the image due to the target motion in the scene. $\mathbf{W_{tgt}}$ is a 2x2 matrix which is a function of x, y and α_t. Both $\mathbf{W_{ego}}$ and $\mathbf{W_{tgt}}$ are easily computed from equation 11. Fig. 5 depicts the velocity fields induced in image. The first two rows show the velocities due to camera pan and tilt motion (egomotion). The third and fourth rows exhibit the velocity fields induced by target motion in space.

$$\begin{bmatrix} \dot{\alpha}_p \\ \dot{\alpha}_t \end{bmatrix} = \boldsymbol{W_{ego}^{-1}}\boldsymbol{W_{tgt}} \begin{bmatrix} \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} 1 & \frac{x(FS(\alpha_t)+yC(\alpha_t))}{(yS(\alpha_t)-FC(\alpha_t))\sqrt{x^2+(FC(\alpha_t)-yS(\alpha_t))^2}} \\ 0 & -\frac{\sqrt{x^2+(FC(\alpha_t)-yS(\alpha_t))^2}}{yS(\alpha_t)-FC(\alpha_t)} \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{\phi} \end{bmatrix} \tag{13}$$

The goal in a tracking application is to move the camera in a such a way that egomotion compensate the image velocity induced by motion in the scene. Assuming that the target velocity is $(\dot{\theta}, \dot{\phi})$, perfect tracking is achieved whenever camera pan and tilt velocities are computed by equation 13. $\mathbf{W_{ego}}$ is a singular matrix only when $y = \frac{F}{\tan(\alpha_t)}$. This happens whenever target projection lays on an horizontal line that contains the intersection point of the pan rotation axis with the image plane. That is the only case in which perfect velocity regulation can not be achieved. Notice that if $\dot{\phi} = 0$, perfect tracking is achieved by making $\dot{\alpha}_p = \dot{\theta}$ and $\dot{\alpha}_t = 0$. The velocity induced in the image by the target motion can be compensated for by camera pan rotation. However if $\dot{\alpha}_t \neq 0$ the camera must move in pan and tilt to keep target position in image. For this case, the residual velocity by making $\dot{\alpha}_p = 0$ and $\dot{\alpha}_t = \dot{\phi}$ can be observed in Fig. 5 (last row)).

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} \approx \begin{bmatrix} F\cos(\alpha_t)(\dot{\alpha}_p - \dot{\theta}) \\ F(\dot{\alpha}_t - \dot{\phi}) \end{bmatrix} \tag{14}$$

Similarly to what was done for position analysis, equation 12 is simplified to decouple pan and tilt control. Equation 12 is obtained from equation 14 assuming $(x, y) = (0, 0)$. The image velocity in Y axis only depends on camera tilt velocity and $\dot{\phi}$ and perfect tracking is achieved whenever $\dot{\alpha}_t = \dot{\phi}$. The same happens for the X axis, camera pan velocity and $\dot{\theta}$.



Velocity in Image (pan motion; camera tilt angle −23º)

Velocity in Image (pan motion; camera tilt angle 0º)

Velocity in Image (pan motion; camera tilt angle 23º)

Velocity in Image (tilt motion; camera tilt angle −23º)

Velocity in Image (tilt motion; camera tilt angle 0º)

Velocity in Image (tilt motion; camera tilt angle 23º)

Velocity in Image (target pan motion; camera tilt angle −23º)

Velocity in Image (target pan motion; camera tilt angle 0º)

Velocity in Image (target pan motion; camera tilt angle 23º)

Velocity in Image (target tilt motion; camera tilt angle 23º)

Velocity in Image (target tilt motion; camera tilt angle −0º)

Velocity in Image (target tilt motion; camera tilt angle −23º)

Velocity in Image (target+camera tilt motion; camera tilt angle −23º)

Velocity in Image (target+camera tilt motion; camera tilt angle 0º)

Velocity in Image (target+camera tilt motion; camera tilt angle 23º)

Fig. 5.: Velocity field in image. Each column corresponds to a certain camera tilt position: first column $\alpha_t = -23°$, second column $\alpha_t = 0°$ and third column $\alpha_t = 23°$. First row: $(\dot{\alpha_p}, \dot{\alpha_t}) = (1, 0)$(rad/s) and $(\dot{\theta}, \dot{\phi}) = (0, 0)$(rad/s). Second row: $(\dot{\alpha_p}, \dot{\alpha_t}) = (0, 1)$(rad/s) and $(\dot{\theta}, \dot{\phi}) = (0, 0)$(rad/s). Third row: $(\dot{\alpha_p}, \dot{\alpha_t}) = (0, 0)$(rad/s) and $(\dot{\theta}, \dot{\phi}) = (1, 0)$(rad/s). Fourth row: $(\dot{\alpha_p}, \dot{\alpha_t}) = (0, 0)$(rad/s) and $(\dot{\theta}, \dot{\phi}) = (0, 1)$(rad/s). Fifth row: $(\dot{\alpha_p}, \dot{\alpha_t}) = (0, 1)$(rad/s) and $(\dot{\theta}, \dot{\phi}) = (0, 1)$(rad/s). Large rectangle (-): image with a field of view of 86°x86°. Small rectangle (:): image with a field of view of 24°x18°.

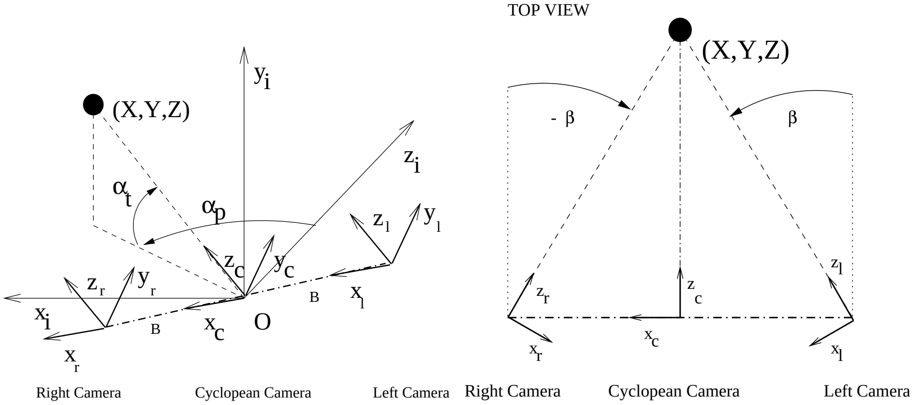


Fig. 6.: Binocular System. Left:Neck pan and tilt. Right: Vergence Control

2.3 Binocular Tracking

In previous section the position and velocity equations for monocular tracking were derived. In this section we focus on the equations for binocular tracking. As depicted in figure 6 two cameras are mounted in a moving platform with length $2B$ (the baseline). The platform is able to rotate in pan (α_p angle) and tilt (α_t angle). Each eye has and additional pan degree of freedom called vergence (β_L and β_R angles). It is assumed that vergence is symmetric ($\beta_L = -\beta_R = \beta$). Binocular tracking is achieved by controlling the three independent degrees of freedom: pan (α_p), tilt (α_t) and vergence (β).

Imagine a third camera positioned in the center of the platform called the cyclopean eye. Neck pan and tilt degrees of freedom align cyclopean Z (forward-looking) axis with target. Vergence control adjusts both camera positions so that target images are projected in the corresponding image centers. Assuming that target is foveated with symmetric vergence angles, these only depend on target motion along the cyclopean Z (forward-looking) axis i.e. on the cyclopean depth of the object. Therefore binocular tracking can be split in two sub-problems: cyclopean eye control and vergence control. Tracking the target with the cyclopean eye is essentially a monocular tracking problem as studied in last section. To perform such a tracking it is necessary to transfer visual information from both retinas to the cyclopean camera. Vergence control is achieved using retinal flow disparity.

2.4 Pan and Tilt Control

$$\begin{cases} z_l \boldsymbol{m_l} = \boldsymbol{AR_p^t(\beta)R_t^t(\alpha_t)R_p^t(\alpha_p)M} + \boldsymbol{AR_p^t(\beta)t} \\ z_r \boldsymbol{m_r} = \boldsymbol{AR_p^t(-\beta)R_t^t(\alpha_t)R_p^t(\alpha_p)M} - \boldsymbol{AR_p^t(-\beta)t} \end{cases} \tag{15}$$

Vector $\mathbf{M}$ represents target cartesian coordinates in the cyclopean inertial frame. Target is projected in the left image at (x_l, y_l) and in the right image at (x_r, y_r). The corresponding homogeneous vectors are $\lambda\mathbf{m_l}$ and $\lambda\mathbf{m_r}$. z_l and z_r are target Z coordinates in left and right camera reference frames. Equation 15 establishes the relationship between $\mathbf{m_l}$, $\mathbf{m_r}$ and $\mathbf{M}$. α_p and α_t are platform pan and tilt angles, β is the symmetric vergence angle and $\mathbf{t} = (B, 0, 0)^t$ where $2B$ is the baseline.

$$\begin{cases} z_l \boldsymbol{m_l} = z\boldsymbol{AR_p^t(\beta)A^{-1}m} + \boldsymbol{AR_p^t(\beta)t} \\ z_r \boldsymbol{m_r} = z\boldsymbol{AR_p^t(-\beta)A^{-1}m} - \boldsymbol{AR_p^t(-\beta)t} \end{cases} \tag{16}$$

Assume (x, y) are target projection coordinates in the cyclopean eye, $\lambda\mathbf{m}$ is the corresponding homogeneous vector and z represents target cyclopean depth. Equation 16 is derived from 15 using 2. The goal of the neck pan and tilt control is to align cyclopean Z axis (forward-looking) with the moving target. This task is similar to the monocular tracking problem. It is necessary to compute target position and velocity in the cyclopean eye from the position and velocity measurements in left and right image.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}\frac{F^2(x_l+x_r)}{S(\beta)^2 x_l x_r - F^2 C(\beta)^2 + FS(\beta)C(\beta)(x_l - x_r)} \\ -\frac{1}{2}\frac{F^2 C(\beta)(y_l+y_r) + FS(\beta)(y_l x_r - y_r x_l)}{S(\beta)^2 x_l x_r - F^2 C(\beta)^2 + FS(\beta)C(\beta)(x_l - x_r)} \end{bmatrix} \tag{17}$$

Equation 16 is a system of six non-linear equations. Assume target image coordinates are known in both left and right camera ((x_l, y_l) and (x_r, y_r)). There are five unknowns in 16: target position in the cyclopean image (x, y) and target Z coordinate in left, right and cyclopean camera referential (z_l, z_r and z). These five unknowns can be determined by solving the system of equations. The solution for (x, y) is given in 17.

$$\begin{aligned}\begin{bmatrix}\dot{x}_{l/r}\\ \dot{y}_{l/r}\end{bmatrix} &= \pm\boldsymbol{K_{l/r}AP_pR_p^t(\pm\beta)A^{-1}m_{l/r}}\dot{\beta} + \boldsymbol{K_{l/r}AR_p^t(\pm\beta)P_tR_p(\pm\beta)A^{-1}m_{l/r}}\dot{\alpha_t}\\ &+\boldsymbol{K_{l/r}AR_p^t(\pm\beta)R_t^t(\alpha_t)P_pR_t(\alpha_t)R_p(\pm\beta)A^{-1}m_{l/r}}\dot{\alpha_p} \mp \tfrac{1}{z_{l/r}}\boldsymbol{K_{l/r}AR_p^t(\pm\beta)P_tt}\dot{\alpha_t}\\ &\mp\tfrac{1}{z_{l/r}}\boldsymbol{K_{l/r}AR_p^t(\pm\beta)R_t^t(\alpha_t)P_pR_t(\alpha_t)t}\dot{\alpha_p} + \tfrac{1}{z_{l/r}}\boldsymbol{K_{l/r}AR_p^t(\pm\beta)R_t^t(\alpha_t)R_p^t(\alpha_p)\dot{M}}\end{aligned} \tag{18}$$

Equation 18 computes the velocity in left and right retina. It is obtained by differentiating equation 15. The first five terms correspond to velocity induced by camera motion (egomotion). Notice that fourth and fifth term depend on scene depth because camera performs translational motion. The last term correspond to the image velocity induced by target motion in space.

$$\begin{bmatrix}\dot{x}_{ego_{l/r}}\\ \dot{y}_{ego_{l/r}}\end{bmatrix} = -\boldsymbol{W_{ego_{l/r}}^{rot}}\begin{bmatrix}\dot{\beta}\\ \dot{\alpha_t}\\ \dot{\alpha_p}\end{bmatrix} + \boldsymbol{W_{ego_{l/r}}^{tra}}\begin{bmatrix}\dot{\alpha_t}\\ \dot{\alpha_p}\end{bmatrix} \tag{19}$$

$$\mathbf{W_{ego_{l/r}}^{tra}} = \frac{B}{z_{l/r}}\begin{bmatrix}0 & (FS(\beta) \pm x_{l/r}C(\beta))C(\alpha_t)\\ 0 & \pm(FS(\beta) + y_{l/r}C(\beta)C(\alpha_t))\end{bmatrix} \tag{20}$$

Equation 19 computes the image velocity component due to camera motion. $\mathbf{W_{ego_{l/r}}^{rot}}$ is a 2x3 weighting matrix corresponding to egomotion induced by camera rotation motion. $\mathbf{W_{ego_{l/r}}^{tra}}$ is a 2x2 weighting matrix that is related with egomotion component due to camera translation (see equation 20). B is the distance from the camera to neck pan and tilt joints. The camera describes translational motion if $B > 0$ and $\dot{\alpha_p} \neq 0$. Whenever B is zero or the neck pan degree of freedom is stopped, the second term of equation 19 is zero and egomotion is only due to camera rotation. The velocity in image induced by camera translation is proportional to the ratio $\frac{B}{z_{l/r}}$. Usually target distance to system is much larger than baseline length. The ratio $\frac{B}{z_{l/r}}$ is nearly zero and the egomotion induced by camera translation can be neglected.

$$\begin{bmatrix}\dot{x}_{ego_l/r}\\ \dot{y}_{ego_l/r}\end{bmatrix} \approx F\begin{bmatrix}\pm1 & 0 & C(\alpha_t)(1+\frac{B}{z_{l/r}}S(\beta))\\ 0 & C(\beta) & \pm S(\alpha_t)(S(\beta)+\frac{B}{z_{l/r}})\end{bmatrix}\begin{bmatrix}\dot{\beta}\\ \dot{\alpha_t}\\ \dot{\alpha_p}\end{bmatrix} \tag{21}$$

Assuming that target is nearly in the center of image, equation 19 simplifies to equation 21 by making $(x_{l/r}, y_{l/r}) = (0, 0)$.

$$\begin{cases}\begin{bmatrix}\dot{x}_{tgt_l}\\ \dot{y}_{tgt_l}\end{bmatrix} = \frac{1}{z_l}\boldsymbol{K_lAR_p^t(\beta)A^{-1}X_{tgt}}\\ \begin{bmatrix}\dot{x}_r\\ \dot{y}_r\end{bmatrix} = \frac{1}{z_r}\boldsymbol{K_rAR_p^t(-\beta)A^{-1}X_{tgt}}\end{cases} \tag{22}$$

Consider $(\dot{x}_{tgt}, \dot{y}_{tgt})^t$ as being the velocity induced in cyclopean image by target motion in scene. This velocity vector is $(\dot{x}_{tgt}, \dot{y}_{tgt})^t = \frac{1}{z}\mathbf{KX_{tgt}}$, where $\mathbf{X_{tgt}}$ is a 3x1 vector such as $\mathbf{X_{tgt}} = \mathbf{AR_t^t}(\alpha_\mathbf{t})\mathbf{R_p^t}(\alpha_\mathbf{p})\dot{\mathbf{M}}$ (see equation 11). Assume $(\dot{x}_{tgt_l}, \dot{y}_{tgt_l})^t$ and $(\dot{x}_{tgt_r}, \dot{y}_{tgt_r})^t$ are image velocities induced by target motion in left and right retina. 22 is system of equations derived from the last term of 18. This system is used to obtain an analytical solution for $(\dot{x}_{tgt}, \dot{y}_{tgt})^t$.

$$\begin{bmatrix}\dot{x}_{tgt}\\ \dot{y}_{tgt}\end{bmatrix} \approx \begin{bmatrix}\frac{\dot{x}_{tgt_l}+\dot{x}_{tgt_r}}{2\cos(\beta)^2}\\ \frac{\dot{y}_{tgt_l}+\dot{y}_{tgt_r}}{2\cos(\beta)}\end{bmatrix} \tag{23}$$

Assuming that target is projected in the center of both retinas, the velocity induced by target motion in the cyclopean image is given by equation 23.

2.5 Vergence Control

Eye pan is called vergence. Figure 6 exhibits a schematic of vergence control. The vergence point is the intersection of left and right optical axis. Assuming symmetric vergence this point is always in the cyclopean Z axis. The system is verged whenever $\tan(\beta) = \frac{B}{z}$ where z is target Z coordinate in the cyclopean reference frame. This section performs the mathematical study for vergence control.

$$\mathit{\Delta}\beta = \arctan(\frac{F\cos(\beta)(x_l - x_r) + 2\sin(\beta)x_l x_r}{F(2F\cos(\beta) - \sin(\beta)(x_l - x_r))}) \tag{24}$$

Assume the vergence angle is β and $\mathit{\Delta}\beta$ is the correction angle that verges the system in the target ($\tan(\beta + \mathit{\Delta}\beta) = \frac{B}{z}$). $\mathit{\Delta}\beta$ is called the vergence error in position and is given by equation 24.

$$\begin{bmatrix} \dot{x}_{l/r} \\ \dot{y}_{l/r} \end{bmatrix} = \pm\frac{1}{z_{l/r}}\boldsymbol{K_{l/r}AP_pR_p^t(\pm\beta)}(z\boldsymbol{A^{-1}m} \pm \mathrm{t})\dot{\beta} + \frac{1}{z_{l/r}}\boldsymbol{K_{l/r}AR_p^t(\pm\beta)A^{-1}}(\dot{z}\boldsymbol{m} + z\dot{\boldsymbol{m}}) \tag{25}$$

Equation 25 is derived by differentiating equation 16. It gives the velocity in left and right retina in order to velocities in the cyclopean eye. The first term corresponds to egomotion induced by vergence degree of freedom. While neck pan and tilt are used to keep the target aligned with cyclopean Z axis, the goal of vergence control is to compensate for target motion along the cyclopean optical axis.

$$\dot{x}_l - \dot{x}_r = -\frac{2F\dot{\beta}}{B} - \frac{2F\sin(\beta)^2\dot{z}}{B} \tag{26}$$

$$\dot{\beta} = \frac{\dot{x}_{tgt_l} - \dot{x}_{tgt_r}}{2F} \tag{27}$$

Assume that cyclopean eye is performing perfect tracking ($(x, y) = (0, 0)$ and $(\dot{x}, \dot{y}) = (0, 0)$) and that target is projected in the center of both retinas ($tan(\beta) = \frac{B}{z}$). Equation 26 is obtained applying these assumptions to 25. It gives the velocity disparity in X axis that is the sum of an egomotion term and target velocity along cyclopean Z axis. To achieve perfect tracking this sum must be zero. If $\dot{x}_{tgt_l} - \dot{x}_{tgt_r}$ is the disparity due to target motion in scene, equation 27 gives the vergence velocity to achieve perfect tracking.

References

1. J.L. Crowley, J.M. Bedrune, M. Bekker, and M. Schneider. Integration and control of reactive visual processes. In *Third European Conference in Computer Vision*, volume 2, pages 47–58, Stockolm,Sweden, May 1994.
2. J. L. Crowley. Coordination of action and perception in a surveillance robot. *IEEE Expert*, 2, November 1987.
3. B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Robot. and Automat.*, 8(3):313–326, June 1992.
4. H. I. Christensen, J. Horstmann, and T. Rasmussen. A control theoretic approach to active vision. In *Asian Conference on Computer Vision*, pages 201–210, Singapore, December 1995.
5. P.I. Corke. *Visual Control of Robots: High-Peformance Visual Servoing.* Mechatronics. John Wiley, 1996.
6. P. Krautgartner and Vincze M. Performance evaluation of vision-based control tasks. In *ICRA98–IEEE Int. Conf. on Robotics and Automation*, pages 2315–2320, Leuven, Belgium, 1998.
7. P. Nordlund and Uhlin T. Closing the loop: detection and pursuit of a moving object by a moving observer. *Image and Vision Computing*, 14:265–275, 1996.
8. D. Murray, P. McLauchlan, I. Reid, and P. Sharkey. Reactions to peripheral image motion using a head/eye plataform. In *Proc. of IEEE International Conference on Computer Vision*, pages 403–411, 1993.

This article was processed using the TEX macro package with SIRS2000 style