共通VideoAPI対応視線位置計測システムの開発

電子技術総合研究所

松田圭司、永見武司

本装置の目的

・視覚に関する心理学的研究
・医療検査
・マンマシンインターフェースへの応用

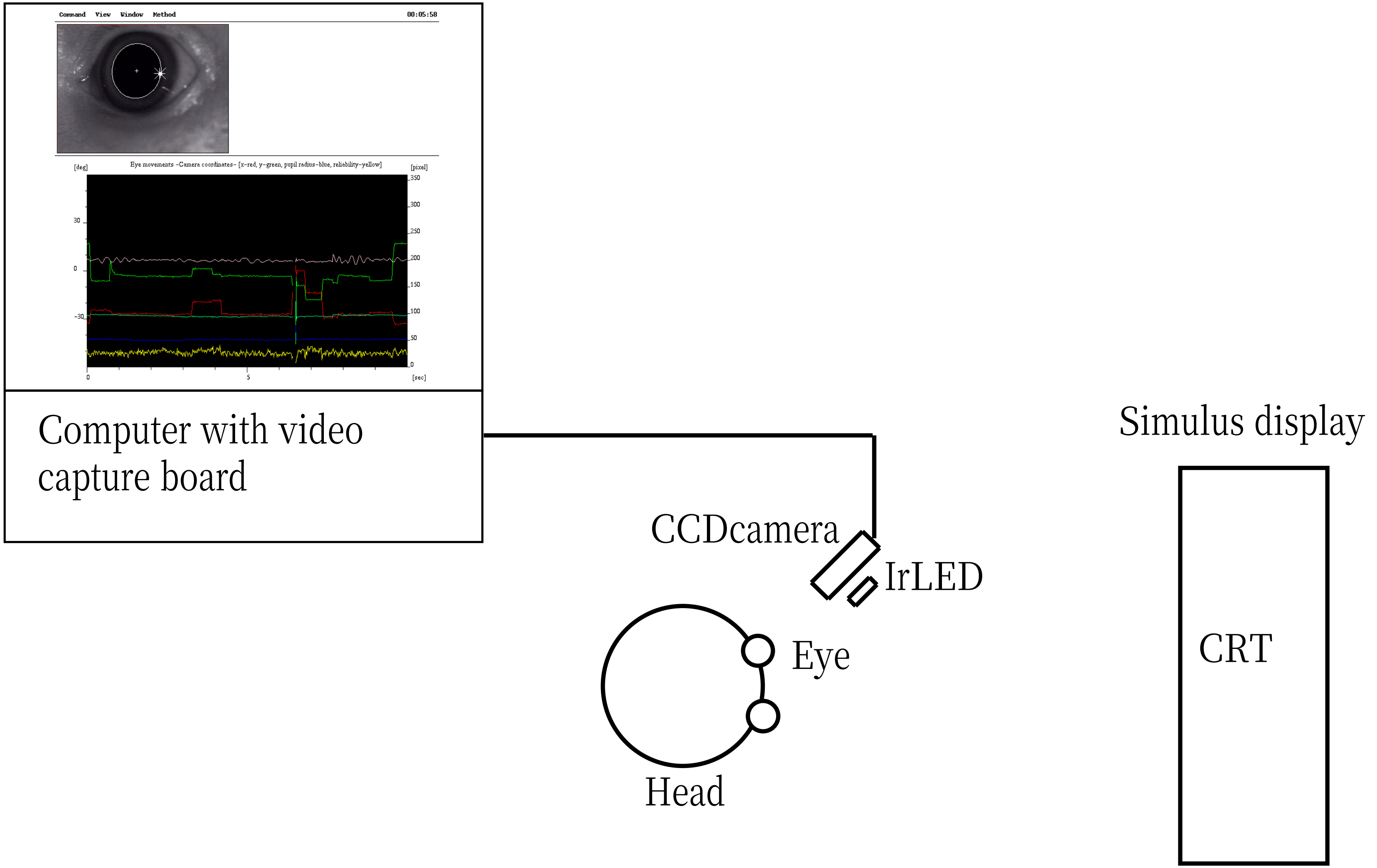
目的を実現するための課題

・十分な精度
・コンパクトな装置環境
・コストの低減

→ 共通ビデオAPIにより実現

・簡便な操作性
・被験者の負担軽減

→ 頭部の動きを許容するアルゴリズム


Command View Window Method
00:05:58
Computer with video capture board
CCDcamera
IrLED
Eye
Head
Simulus display
CRT

装置概要

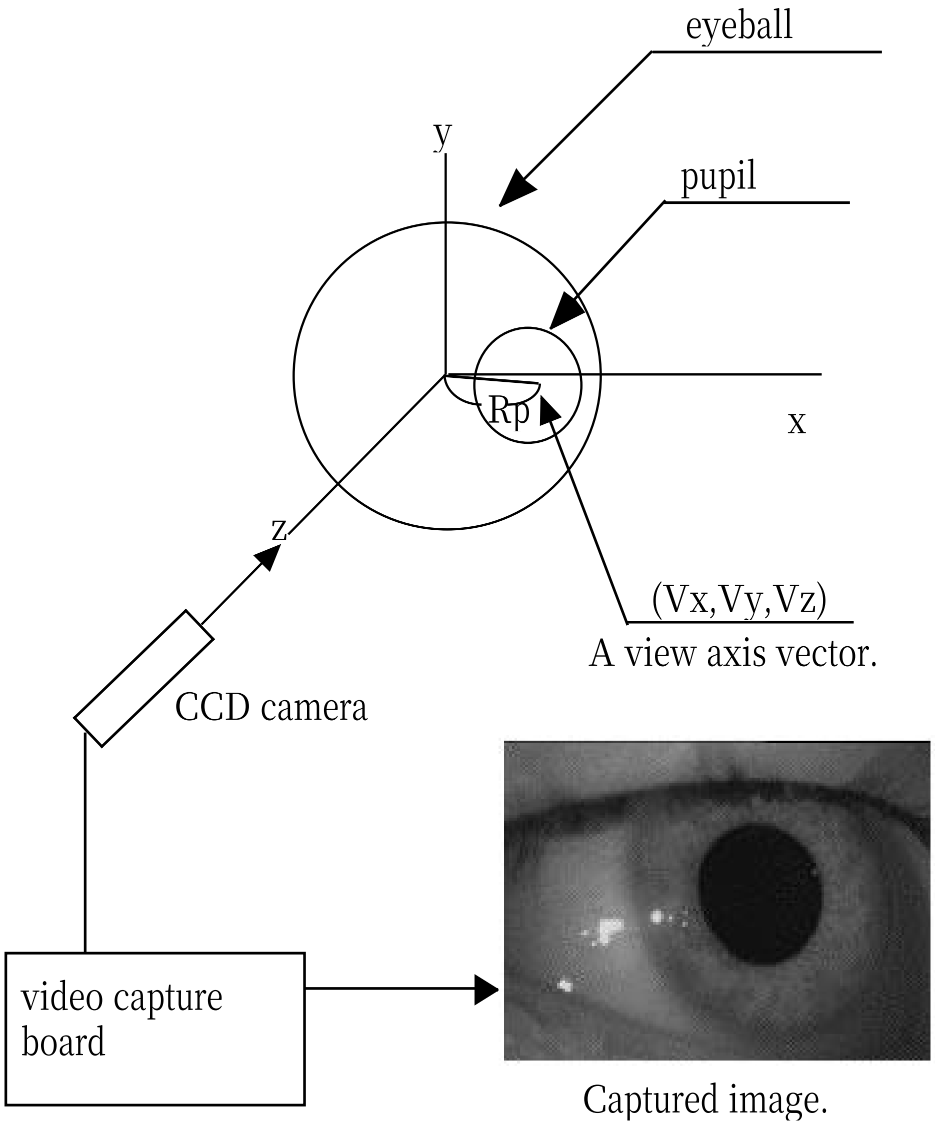
共通ビデオアプリケーションインターフェース

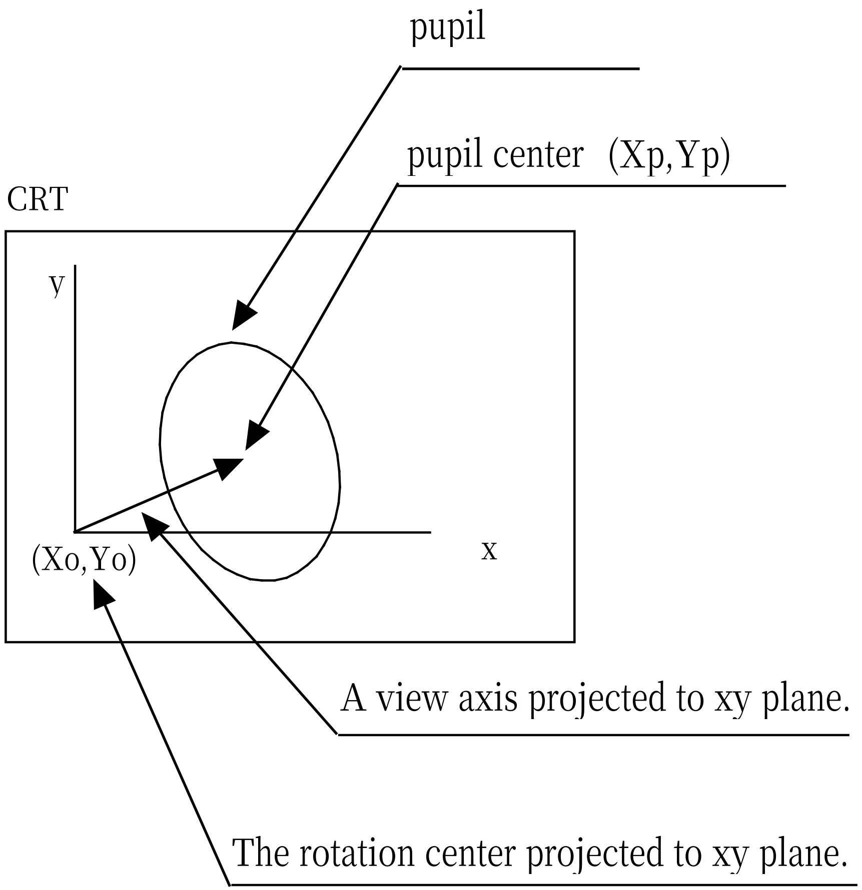
様々なハードウェア上で動作するために、共通ビデオAPIを介してハードウェアにアクセスする

・ハードウェア初期化命令	void EPH_Init()
・画像取込み命令	void EPH_GetImage();
・任意の座標の濃淡値を読み出す命令	int EPH_Read(int x,int y);
・ハードウェア終了命令	void EPH_End();

現在動作しているハードウェア

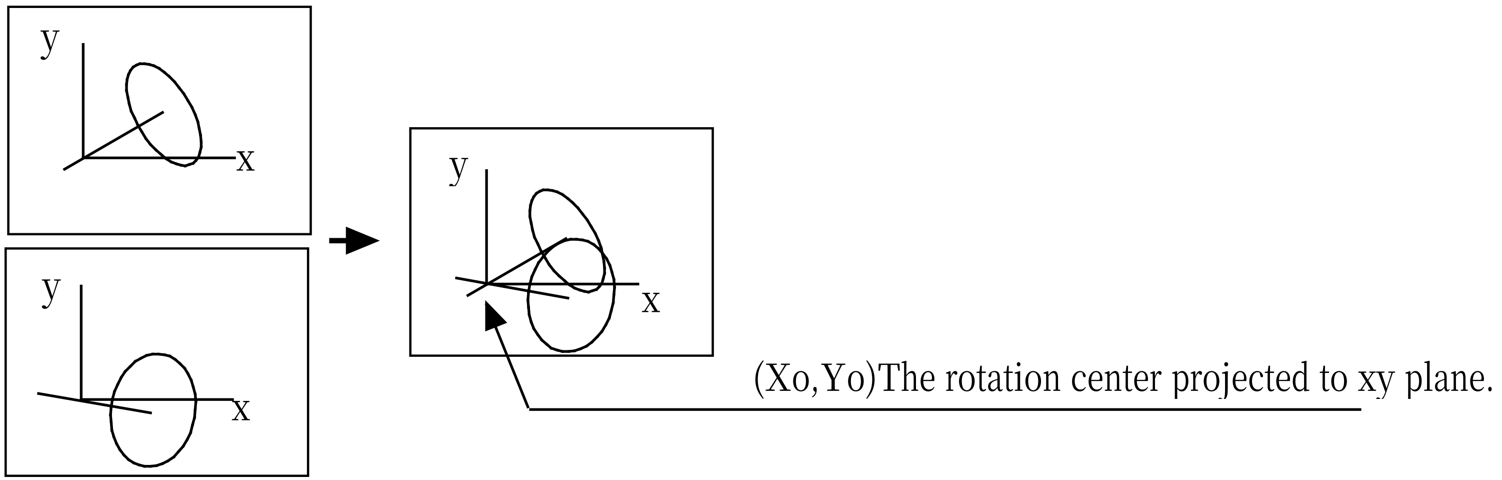
・Video for Linux に適合しているビデオ入力装置（Linux , 60Hz対応）
・IBM Smart Capture Card (FreeBSD & notebook)
・アルゴクラフト社　ビデオ入力装置（Linux)



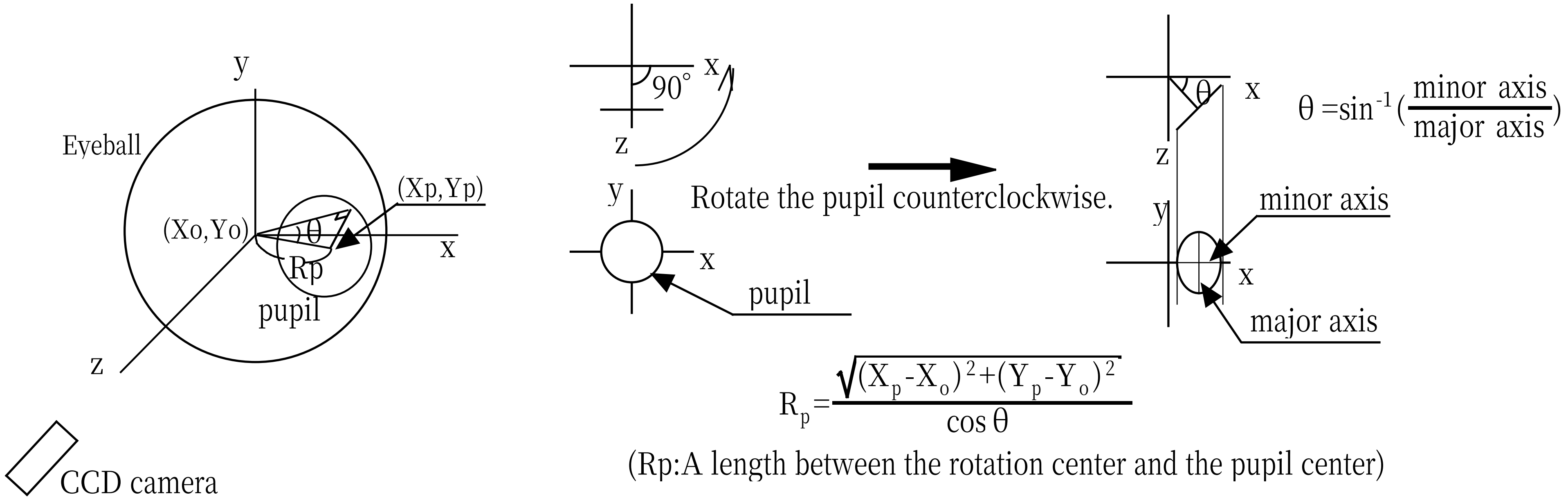


$$(v_x, v_y, v_z) = \left(\frac{X_p - X_o}{R_p}, \frac{X_p - Y_o}{R_p}, \frac{\sqrt{R_p^2 - (X_p - X_o)^2 - (X_p - Y_o)^2}}{R_p} \right)$$

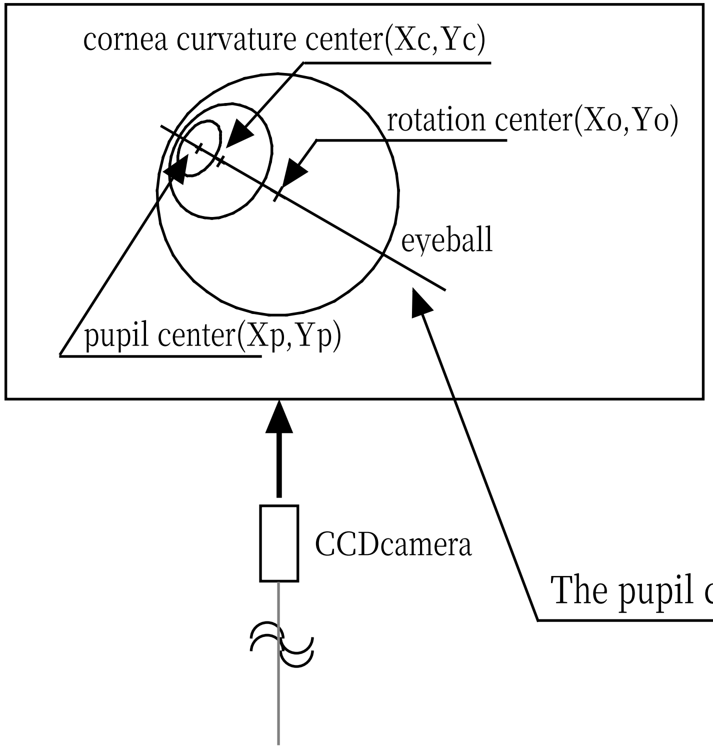
頭部の動きを許容しない従来のシステム



The rotation center projected to xy plane exists on the extended line of ellipses' minor axises.



眼球回転中心位置、瞳孔中心回転半径を求める



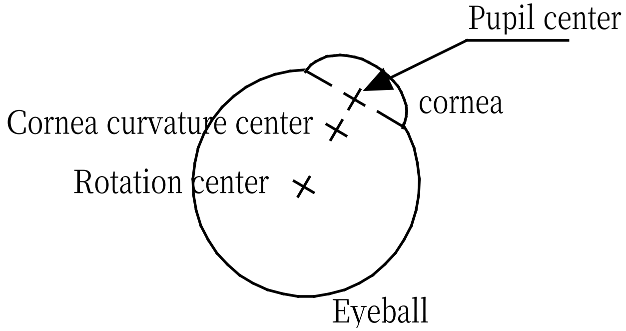
The view axis in the camera coordinate system.

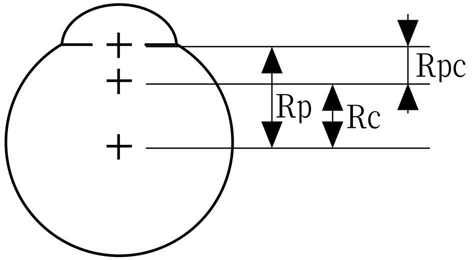
$$(V_x, V_y, V_z) = \left(\frac{X_p - X_c}{R_{pc}}, \frac{Y_p - Y_c}{R_{pc}}, \frac{\sqrt{R_{pc}^2 - (X_p - X_c)^2 - (Y_p - Y_c)^2}}{R_{pc}} \right)$$

The rotation center.

$$(X_o, Y_o) = \left(\frac{R_c X_p - R_p X_c}{R_c - R_p}, \frac{R_c Y_p - R_p Y_c}{R_c - R_p} \right)$$

The pupil center , the cornea curvature center and rotation center exist on the same line.

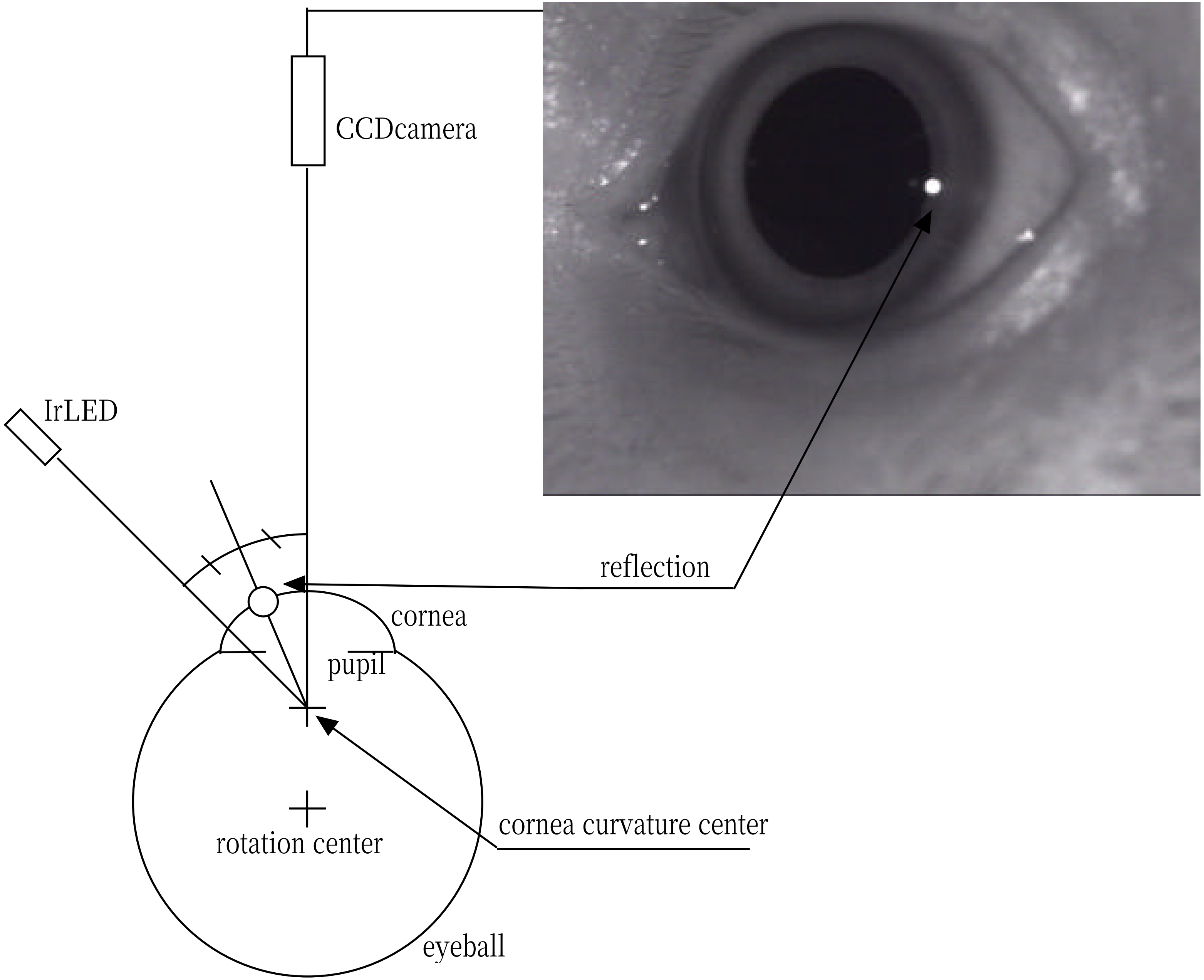




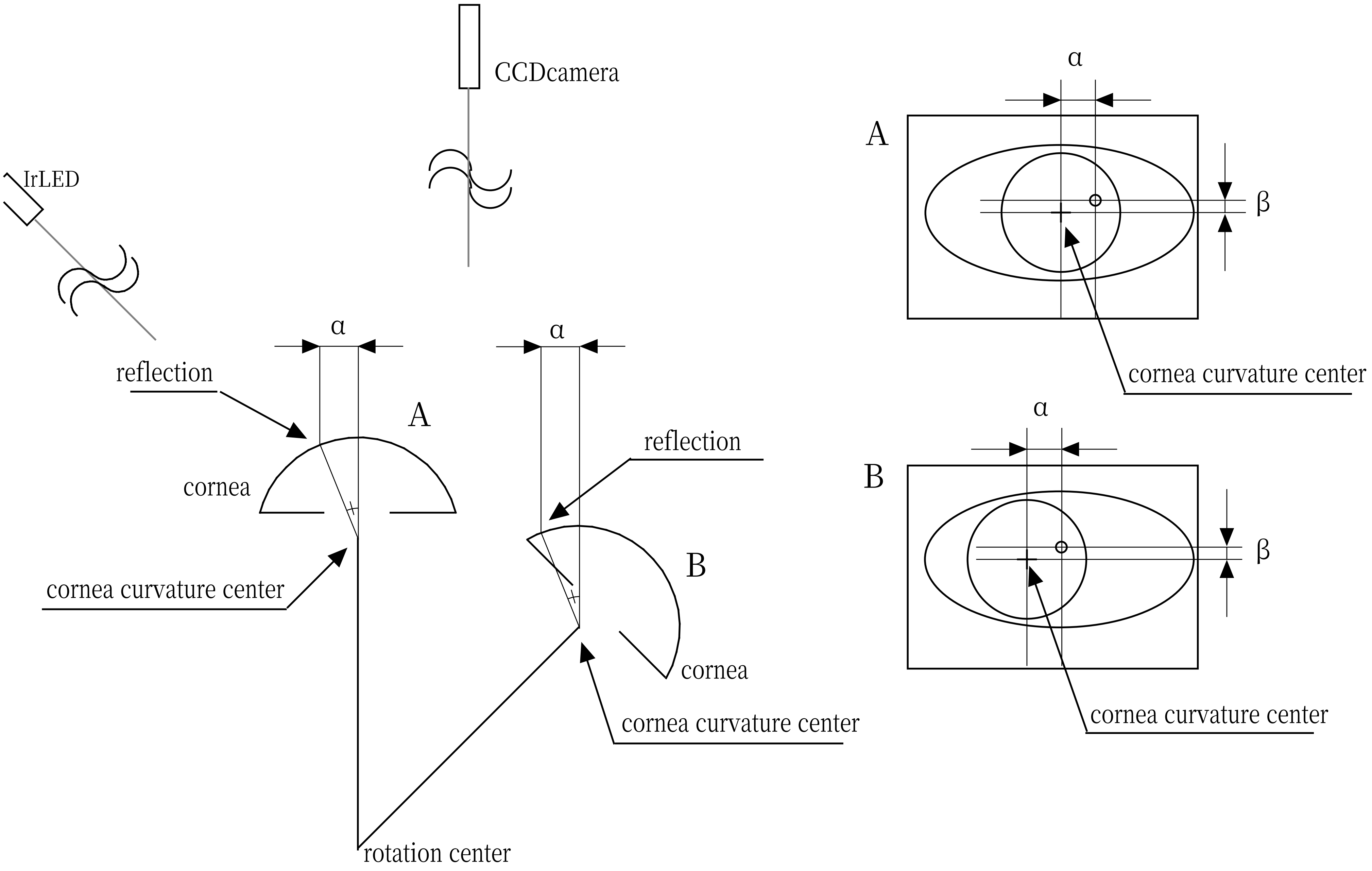
$$Rpc : Rp = \sqrt{(Xp-Xc)^2+(Yp-Yc)^2} : \sqrt{(Xp-Xo)^2+(Yp-Yo)^2}$$

$$Rc = Rp - Rpc$$

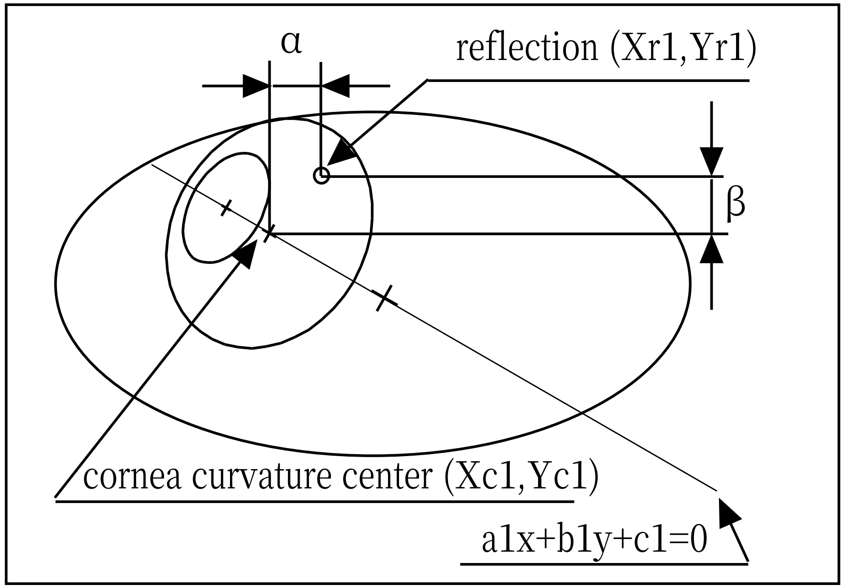
頭部の動きを許容するシステムの原理


CCDcamera
IrLED
reflection
cornea
pupil
cornea curvature center
rotation center
eyeball

反射点出現位置

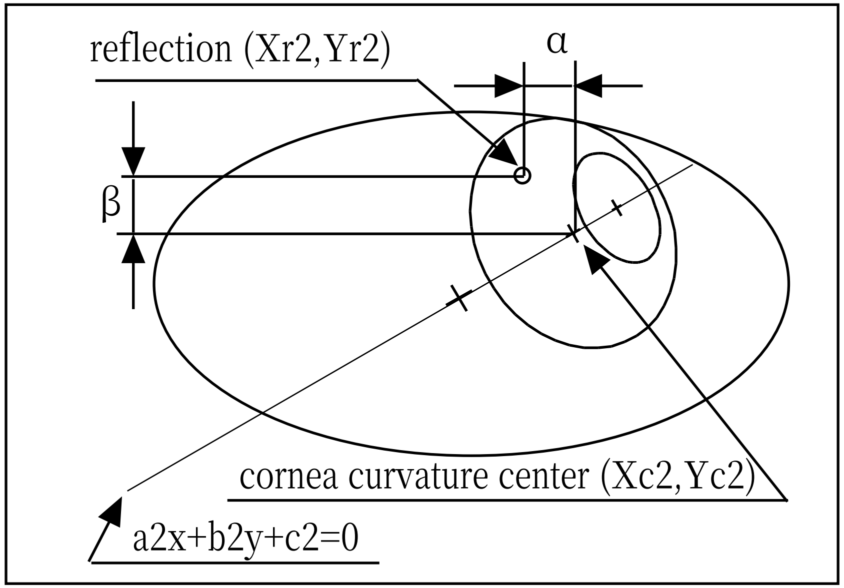


反射点と角膜曲率中心の位置関係



$$\begin{cases} Xc1 = Xr1 - \alpha \\ Yc1 = Yr1 - \beta \\ a1Xc1 + b1Yc1 + c1 = 0 \end{cases}$$

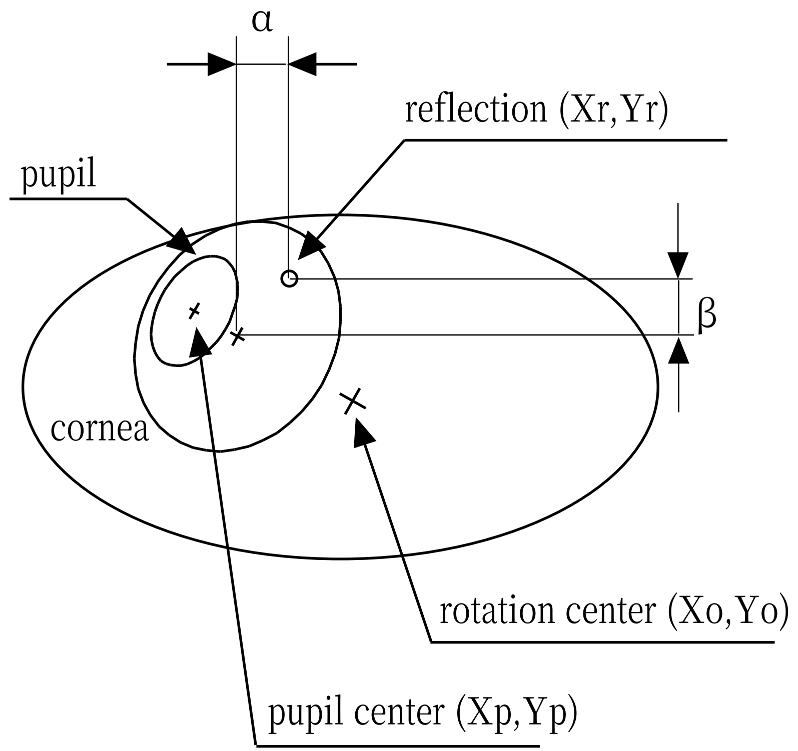
$$a1(Xr1 - \alpha) + b1(Yr1 - \beta) + c1 = 0$$

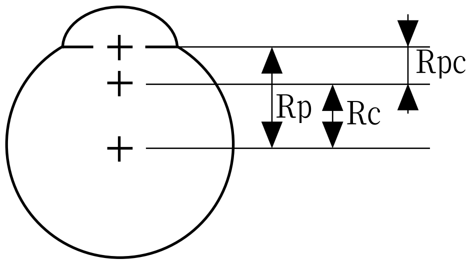


$$\begin{cases} Xc2 = Xr2 - \alpha \\ Yc2 = Yr2 - \beta \\ a2Xc2 + b2Yc2 + c2 = 0 \end{cases}$$

$$a2(Xr2 - \alpha) + b2(Yr2 - \beta) + c2 = 0$$

反射点のオフセットを求める

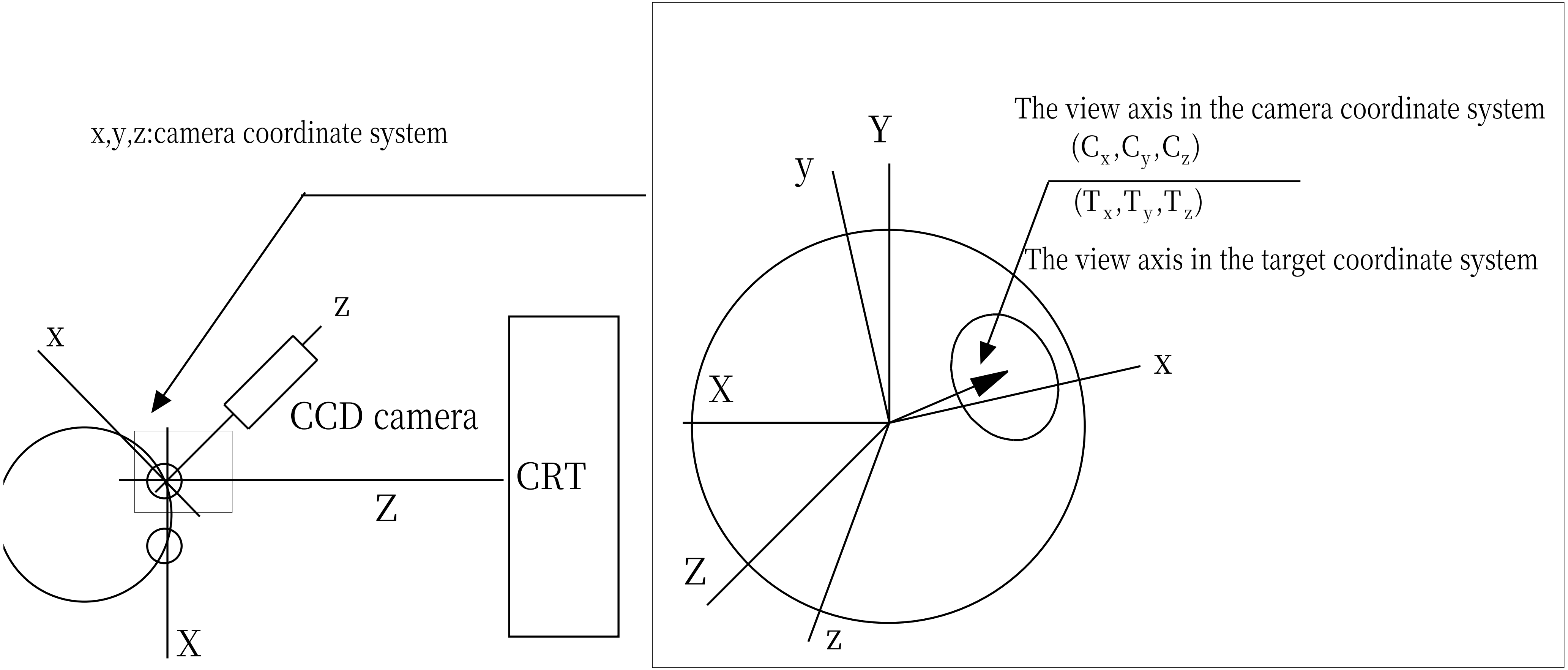




視線ベクトル $$(V_x,V_y,V_z)=\left(\frac{X_p-X_r+\alpha}{R_{pc}},\frac{Y_p-Y_r+\beta}{R_{pc}},\frac{\sqrt{R_{pc}^2-(X_p-X_r+\alpha)^2-(Y_p-Y_r+\beta)^2}}{R_{pc}}\right)$$

眼球回転中心座標 $$(X_o,Y_o)=\left(\frac{R_cX_p-R_p(X_r-\alpha)}{R_c-R_p},\frac{R_cY_p-R_p(Y_r-\beta)}{R_c-R_p}\right)$$

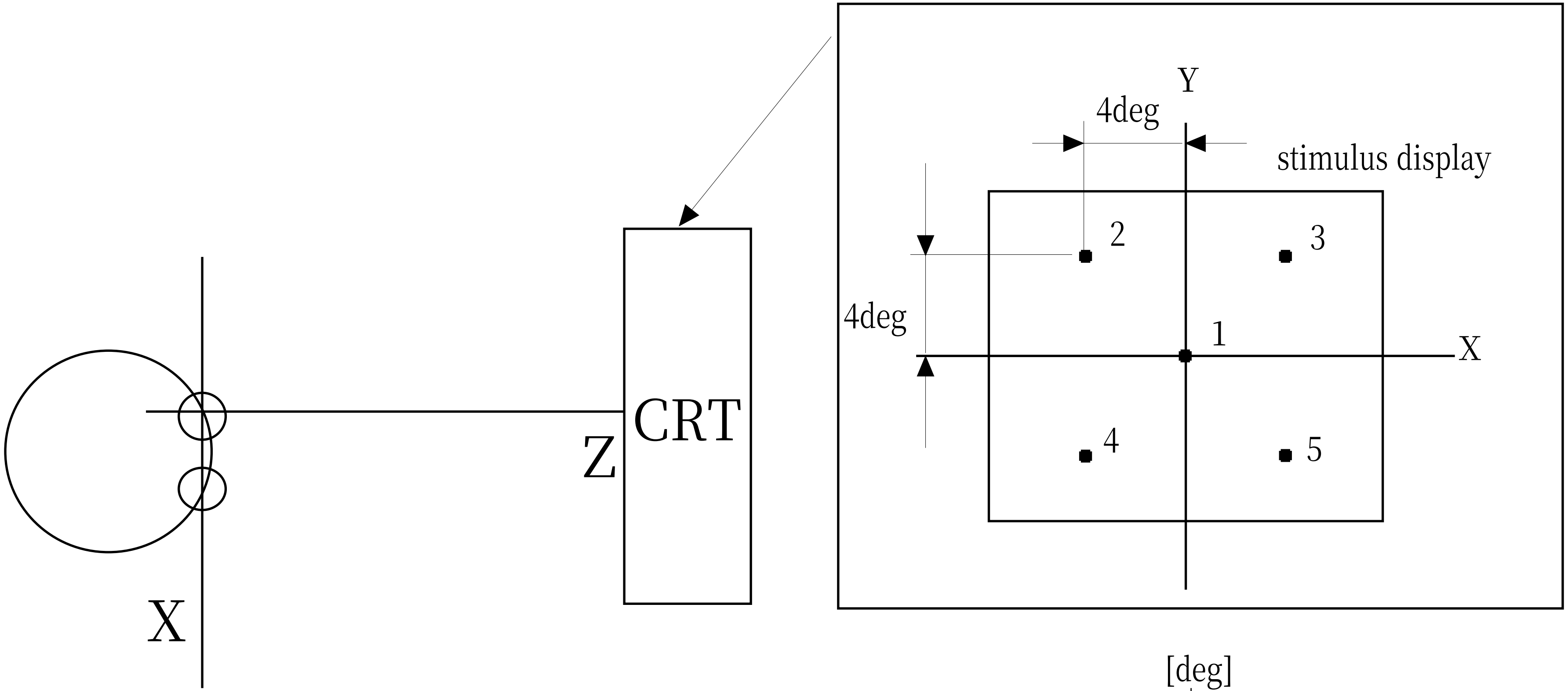
瞳孔中心座標、反射点より視線ベクトル及び眼球回転中心位置を求める

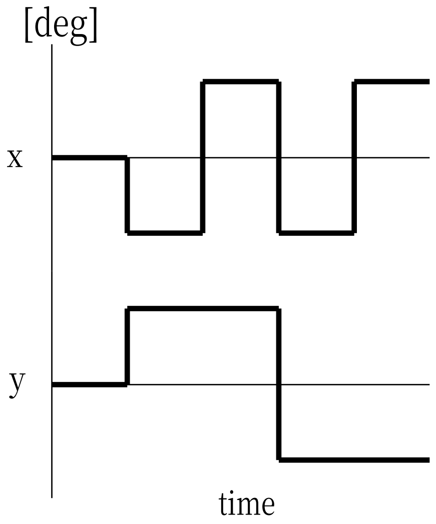


X,Y,Z:the target coordinate system

$$\begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} C_x \\ C_y \\ C_z \end{bmatrix}$$

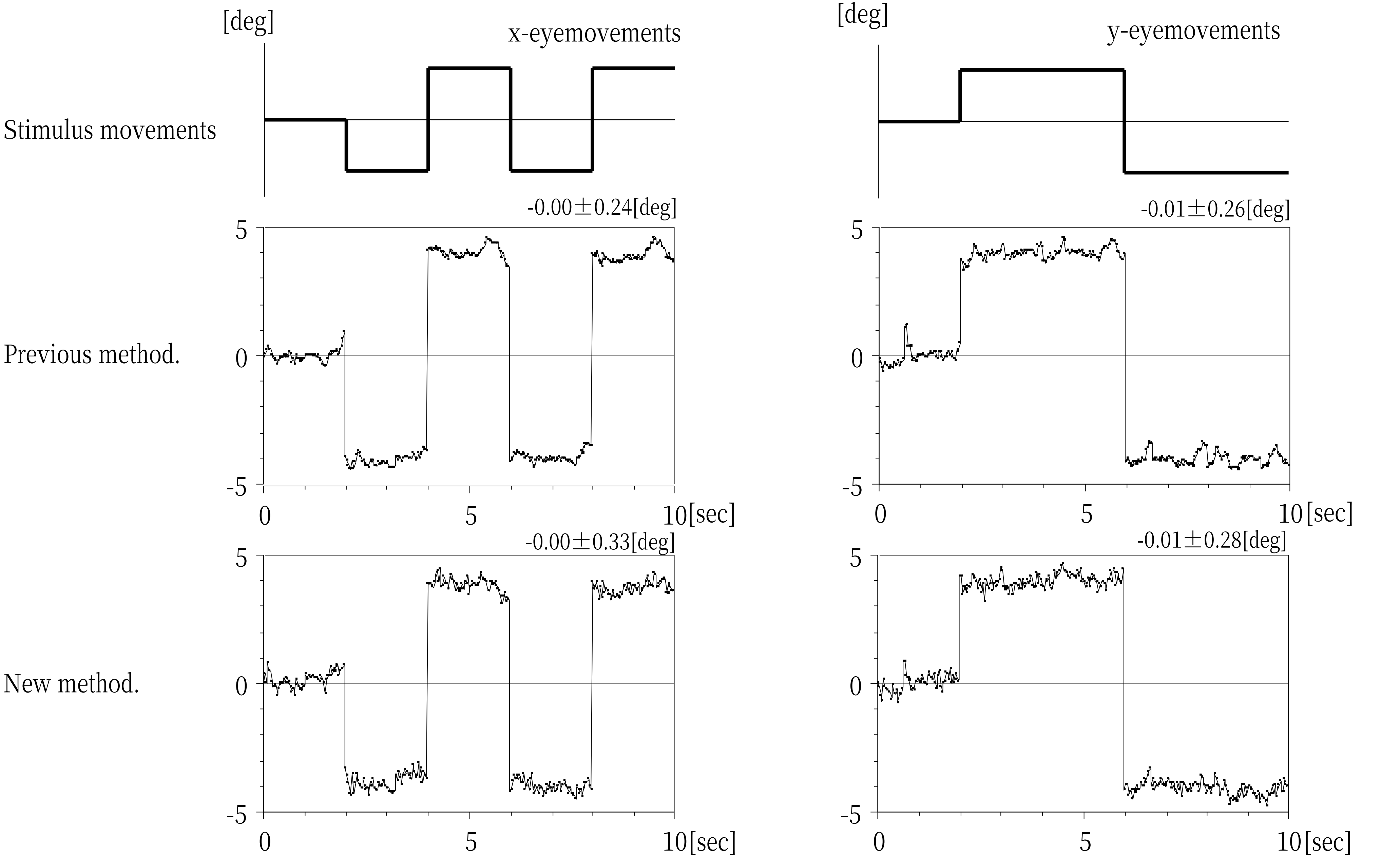
座標変換行列



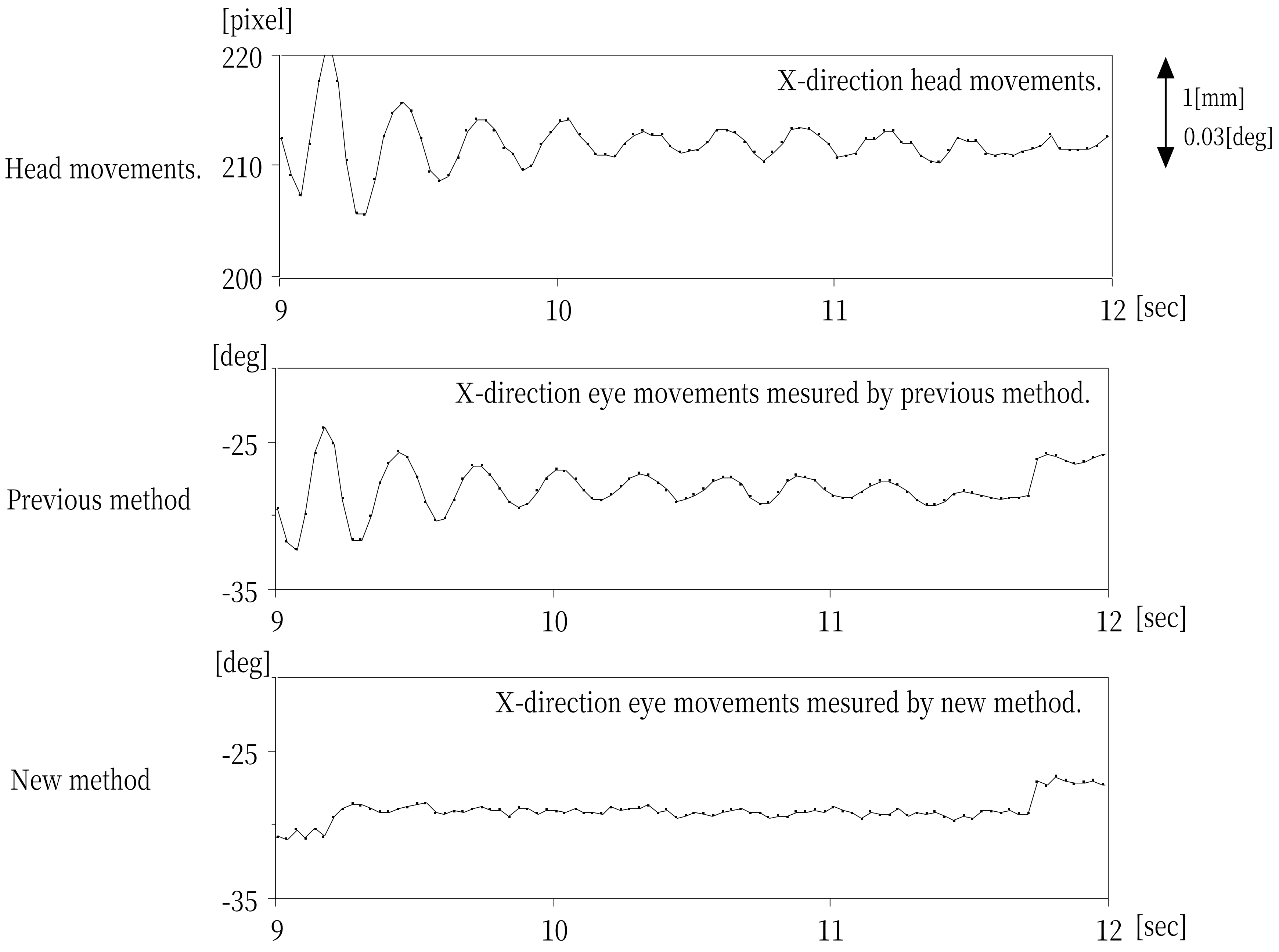


$$\begin{bmatrix} a & d & g \\ b & e & h \\ c & f & i \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} C_{xi}^2 & \sum_{i=1}^{n} C_{xi} C_{yi} & \sum_{i=1}^{n} C_{zi} C_{xi} \\ \sum_{i=1}^{n} C_{xi} C_{yi} & \sum_{i=1}^{n} C_{yi}^2 & \sum_{i=1}^{n} C_{yi} C_{zi} \\ \sum_{i=1}^{n} C_{zi} C_{xi} & \sum_{i=1}^{n} C_{yi} C_{zi} & \sum_{i=1}^{n} C_{zi}^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^{n} C_{xi} T_{xi} & \sum_{i=1}^{n} C_{xi} T_{yi} & \sum_{i=1}^{n} C_{xi} T_{zi} \\ \sum_{i=1}^{n} C_{yi} T_{xi} & \sum_{i=1}^{n} C_{yi} T_{yi} & \sum_{i=1}^{n} C_{yi} T_{zi} \\ \sum_{i=1}^{n} C_{zi} T_{xi} & \sum_{i=1}^{n} C_{zi} T_{yi} & \sum_{i=1}^{n} C_{zi} T_{zi} \end{bmatrix}$$

座標変換行列を求める


[deg]
x-eyemovements
[deg]
y-eyemovements
Stimulus movements
-0.00±0.24[deg]
-0.01±0.26[deg]
Previous method.
-0.00±0.33[deg]
-0.01±0.28[deg]
New method.
5
0
-5
0
5
10[sec]

実験結果



新しい方法による頭部の動き補償

まとめ

・共通VideoAPIに対応したプログラムを作成した。

・Video for Linuxを用いることにより、60Hzでの計測が可能となった。

・角膜反射点を利用することより、頭部の動きを許容する視線位置計測システムの構築ができた。

・従来よりもコストが下がり、適応できるハードウェアの種類も多いため、様々なアプリケーションへの波及を期待することができる。

・http://www.etl.go.jp/~matsuda/eye/よりソフトウェアの入手が可能である。