

## 1. はじめに

我々は、サル、人間を対象にした視線位置検出システムを開発している[1]-[7]。脳機能解明のために、無麻酔行動中のサルを実験動物に使う研究、人の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などの場面において、簡便な視線位置計測システムが求められている[8]-[11]。我々の開発したシステムは、頭部を固定した被験者の眼球をカメラで撮影し、画像処理によって視線位置を計測するものである。システムは次の装置より構成される (Fig.1)。

- 1) 照明用赤外線LED
- 2) 瞳孔画像を撮影するCCDカメラ
- 3) ビデオ入力装置
- 4) 画像処理コンピュータ
- 5) 視覚刺激提示装置

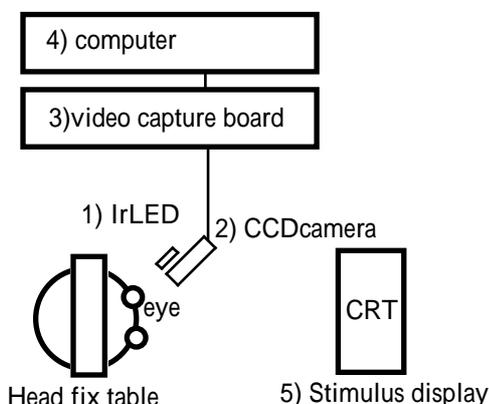


Fig.1 System outline.

このシステムは、以下のような特徴を持つ。画像処理により視線位置計測を行う無侵襲計測のため、被験者に対する負担が少ない。視線位置に加え、心理実験などに用いられる瞳孔径の測定が可能である。瞳孔の撮影できる範囲ならば、任意の位置にカメラを設置することができる。画像処理に関わるプログラムは、ハードウェアに依存する部分とそれ以外の部分が独立しており、様々なプラットフォームへの実装が容易である。システムを構成する装置は、すべて一般市販品のみを使用しているため、容易に入手ができる上に、安価である。

## 2. 視線位置計測の原理

本システムは、「瞳孔をCCDカメラで撮影し、画像処理により視線位置を計測する」ものである。計測を行う上で、いくつかの条件を仮定している。

- 1) 頭部、カメラ、視覚刺激提示装置は動かない。
- 2) 眼球は、ある一点を中心に回転する。
- 3) 瞳孔は、真円である。
- 4) カメラに写る画像は、平行投影されている。
- 5) 視線方向とは、瞳孔の法線方向である。

これらの前提により、カメラに写った瞳孔画像から視線方向を求める。

瞳孔を空間に存在する厚みのない円盤と考える。この円盤 (瞳孔) は、ある点を中心に回転する球 (眼球) の一部を切り取った面 (虹彩) に含まれ、かつその面の中心と、円盤の中心は一致する。円盤の中心座標を  $(X_c, Y_c, Z_c)$  とする。球の中心の座標 (回転中心座標) を  $(X_o, Y_o, Z_o)$  とする。円盤の中心  $(X_c, Y_c, Z_c)$  から、球の中心  $(X_o, Y_o, Z_o)$  までの距離を  $R$  (回転半径) とする。xy平面は、CCDカメラ面に平行とする。ここで、設定されている座標系をカメラ座標系とする。円盤の法線ベクトル  $(V_x, V_y, V_z)$  は、 $(X_c - X_o, Y_c - Y_o, Z_c - Z_o)$  と表すことができる。前提より、これが、視線方向を表すベクトルとなる (Fig.2)。

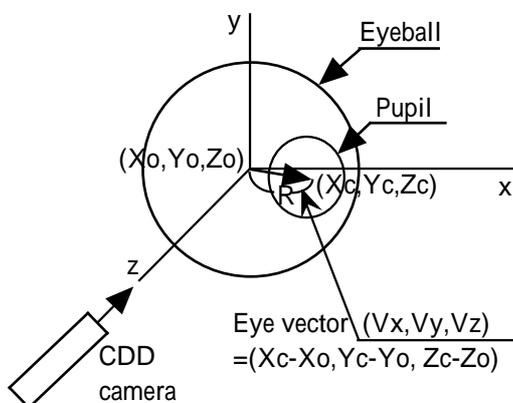


Fig.2 Camera Coordinates.

円盤のxy平面への平行投影は、カメラによって撮影された瞳孔に相当する。平行投影されるため、回転中心は  $(X_o, Y_o)$  に、円盤の中心は  $(X_c, Y_c)$  に投影される。このように問題を設定しなおすと、平面に投影された値より、円盤の法線ベクトルを求めることが目的となる (Fig.3)。

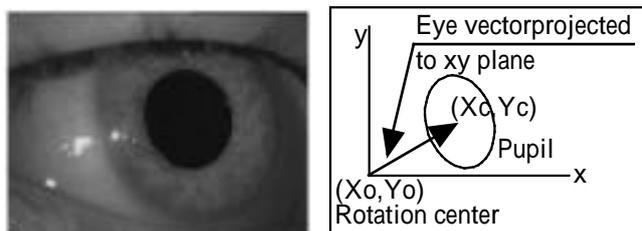


Fig.3 A captured image.

法線ベクトルのx、y成分については、平行投影なので、そのまま使用することができる。法線ベクトルのz成分については、次の式により求めることができる。

$$Z_c - Z_o = \sqrt{R^2 - (X_c - X_o)^2 - (Y_c - Y_o)^2} \dots \text{Eq.1}$$

以上をまとめると、眼球の回転中心座標を  $(X_o, Y_o)$ 、回転半径を  $R$ 、瞳孔の中心座標を  $(X_c, Y_c)$  としたとき、カメラ座標系において、視線ベクトル  $(V_x, V_y, V_z)$  は、次の式で表すことができる。

$(V_x, V_y, V_z)$

$$=(X_c-X_o, Y_c-Y_o, \sqrt{R^2-(X_c-X_o)^2-(Y_c-Y_o)^2}) \dots \text{Eq.2}$$

以後の使用を顧慮しベクトルの長さを1に規格化した視線ベクトルを以下のように定義する。

$(C_x, C_y, C_z)$

$$=(\frac{X_c-X_o}{R}, \frac{Y_c-Y_o}{R}, \frac{\sqrt{R^2-(X_c-X_o)^2-(Y_c-Y_o)^2}}{R}) \dots \text{Eq.3}$$

以上より、カメラ座標系における、視線方向を求めるためには、

- 1) 眼球回転中心
- 2) 瞳孔回転半径
- 3) 瞳孔中心座標

が必要であることがわかる。1)、2)は、実験準備段階での校正作業により決定する。

### 1) 眼球回転中心測定法

眼球回転中心は次の原理を用いて測定する。平面に投影された円盤は、xy平面において楕円になる。この楕円の短軸の延長線上に平面に投影された回転中心が存在する。そこで、複数の瞳孔画像を撮影し、それぞれの瞳孔を楕円で近似する。楕円の短軸の延長線の交点を最小自乗法を用いて求める (Fig.4)。

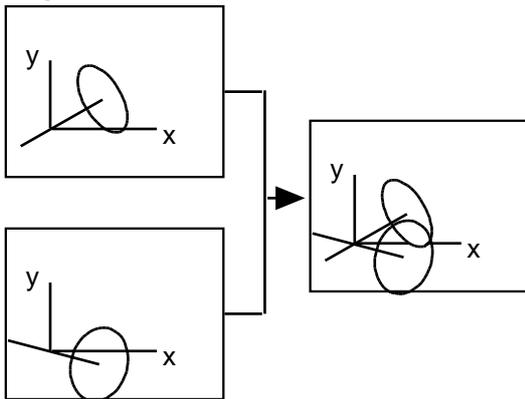


Fig.4 Calculate the center of the eyeball projected to xy plane.

実際の測定においては、被験者の眼球の動きを数秒間撮影することにより計測することができる (Fig.5)。

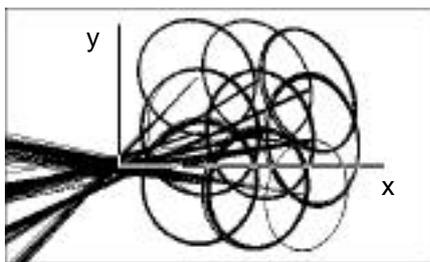


Fig.5 A sample of getting a rotation center.

### 2) 瞳孔回転半径測定法

瞳孔回転半径は、次の方法により測定する。xy平

面と視線ベクトルのなす角度を  $\theta$  とする。瞳孔がカメラに正面を向く時、つまり  $\theta$  が  $90^\circ$  の時は、瞳孔は真円となり、短軸と長軸の比は、1になる (Fig.6)。眼球の回転により、短軸と長軸の比は、変化してゆく。この関係は次の関数で表すことができる。

$$=\sin^{-1}(\frac{\text{minor axis}}{\text{major axis}}) \dots \text{Eq.4}$$

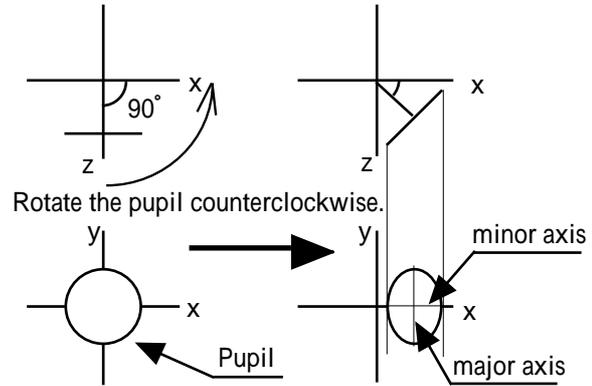


Fig.6 Relation between angle and axis ratio.

また、回転半径Rは、Eq.5によって、xy平面に投影された視線ベクトルと角度  $\theta$  を用いて表すことができる (Fig.7)。

$$R = \frac{\sqrt{(X_c-X_o)^2+(Y_c-Y_o)^2}}{\cos \theta} \dots \text{Eq.5}$$

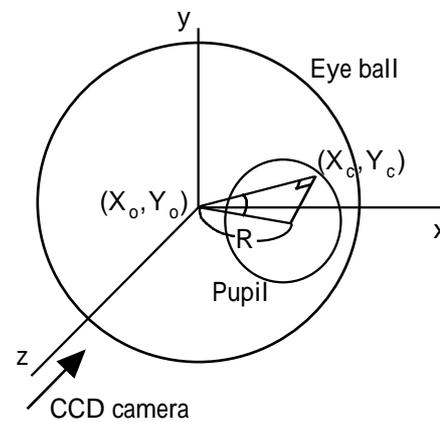


Fig.7 Calculate rotation radius.

実際の測定においては、複数の画像よりRを求め、平均値を回転半径とする。

### 3) 瞳孔中心座標

画像処理により瞳孔を楕円で近似する。楕円の中心座標を計算により求める。次の章で詳細に検討する。

以上述べてきた方法により、瞳孔中心座標よりカメラ座標系における視線ベクトルを求めることが可能となる。しかし、実際に測定したいものは、被験者の見ている場所である。原点が眼球回転中心、視

覚刺激が提示される平面がxy平面となる座標系を対象物座標系とする ( Fig.8)。被験者の見ている場所を知るためには、カメラ座標系ではなく、対象物座標系における視線ベクトルを求める必要がある。それぞれの座標系における視線ベクトルは、Eq.6により変換できる。

$$\begin{pmatrix} O_x \\ O_y \\ O_z \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \begin{pmatrix} C_x \\ C_y \\ C_z \end{pmatrix} \dots \text{Eq.6}$$

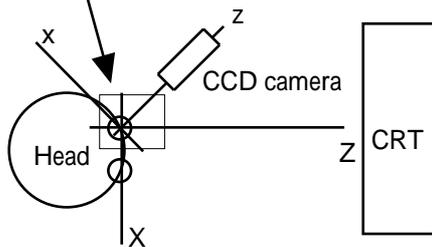
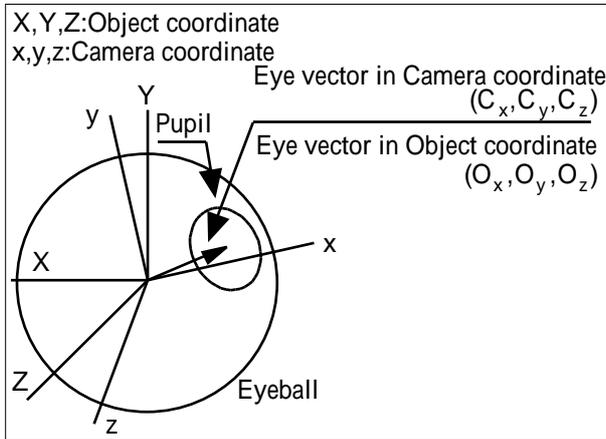


Fig.8 Camera and Object coordinates.

変換行列は、被験者に対象物座標系における位置が計測されている視標を提示することにより求める。例としてFig.9を用いる。Fig.9で示される視標を順に提示する。視標を見ている時の瞳孔画像より、カメラ座標系における視線ベクトルを測定する。Fig.9において、1番目の視標の対象物座標系における長さ1のベクトル( $O_{x1}, O_{y1}, O_{z1}$ )は、次に式で表される。

$$= \left( \frac{-10}{\sqrt{10^2+10^2+50^2}}, \frac{10}{\sqrt{10^2+10^2+50^2}}, \frac{50}{\sqrt{10^2+10^2+50^2}} \right)$$

...Eq.7

1番目の視標を見ている時のカメラ座標系における視線ベクトル( $C_{x1}, C_{y1}, C_{z1}$ )を測定する。他の点に関しても同様にベクトルの組を測定する。3点以上のベクトルの組があれば、Eq.8より最小自乗法を用いて、カメラ座標系から対象物座標系に変換する行列を得ることができる。Fig.9の視標を用いる場合には、 $n=9$ となる。変換に用いる行列は、Eq.8の左辺の行列の転置行列である。この行列を用いて座標変換することにより、カメラ座標系における視線ベクトルを対象物座標系における視線ベクトルに変換することができる。刺激提示面までの距離がLであるとき、刺激提示面における視線位置(X cm, Y cm)は、

Eq.10で表すことができる。

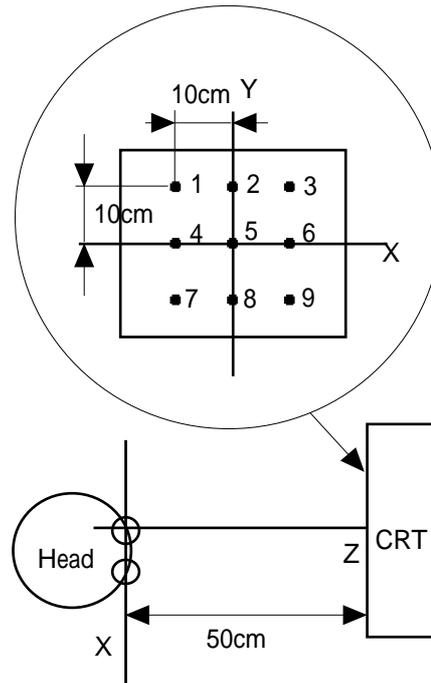


Fig.9 Camera and Object coordinates.

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

...Eq.8

$$(X, Y) = \left( \frac{O_x}{O_z} L, \frac{O_y}{O_z} L \right) \dots \text{Eq.10}$$

今まで述べてきた方法により、あらかじめ校正によって、眼球回転中心、瞳孔回転半径、カメラ座標—対象物座標変換行列を求めておく。それらの値とEq.3、Eq.6、Eq.10を用いることにより、CCDカメラによって撮影された画像の瞳孔中心座標から、そのときの視線位置を算出することができる。

### 3. 画像処理による瞳孔検出法

以下に述べる1)から4)の手順に従って、撮影された画像を処理し、瞳孔を楕円として検出し、カメラ座標系のxy平面における楕円の式 (Eq.11) の係数を求める。5)より係数から必要な値を求めることができる。

$$x^2 + axy + by^2 + cx + dy + e = 0 \dots \text{Eq.11}$$

#### 1) 瞳孔検出のための閾値の決定

実験室等で明るさの変動が無い場合は、手動で設定することもできるが、ここでは、自動で設定する方法について記述する。画像の明るさのヒストグラ

ムを作成する。全画素を用いたヒストグラムを作成すると、時間が掛かりリアルタイム動作に支障が出るため、画素を間引き ヒストグラムを作成する（実装例においては、40x30pixelsの画像より求めている）(Fig.10)。

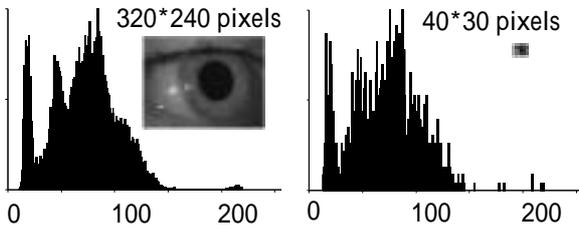


Fig.10 Brightness histogram of a image.

画素間引きによる分布の偏りを補正するため、移動平均を用いて、ヒストグラムを平滑化する場合もある。瞳孔を暗く撮影するため、最も左にあらわれるピークの右側の谷の部分が、最適な閾値となる。また、瞳孔の全体に占める面積は、ある一定値以下になるので、その範囲におけるピークの右の谷を、閾値とすることにより最適な閾値を設定することができる。外部の明るさの変動が急峻でない場合は、時間方向に移動平均をしたヒストグラムを作成することにより、瞳孔が撮影されない場合などに、閾値が大きく動いてしまうといったエラーを回避することができる。

## 2) 瞳孔の検出

閾値を決定するのに使用した間引きによる縮小画像を用い、閾値以下の画素のラベリング（画素の塊毎に番号をつける）を行う。ラベリングにおいて、大きさが1画素のものは無視し、上下方向のみ連結を行う。面積が最大の塊に注目する。塊の外周の点かつ隣接する点が凸に接続される画素の座標 $(x_i, y_i)$ をEq.12に代入し連立方程式を解くことにより、最小自乗法を用いて楕円の式（Eq.11）の係数 $(a, b, c, d, e)$ を求めることができる。

$$\begin{cases} a \sum_{i=1}^n x_i^2 y_i^2 + b \sum_{i=1}^n x_i y_i^3 + c \sum_{i=1}^n x_i^2 y_i + d \sum_{i=1}^n x_i y_i^2 + e \sum_{i=1}^n x_i y_i = - \sum_{i=1}^n x_i^3 y_i \\ a \sum_{i=1}^n x_i y_i^3 + b \sum_{i=1}^n y_i^4 + c \sum_{i=1}^n x_i y_i^2 + d \sum_{i=1}^n y_i^3 + e \sum_{i=1}^n y_i^2 = - \sum_{i=1}^n x_i^2 y_i^2 \\ a \sum_{i=1}^n x_i^2 y_i + b \sum_{i=1}^n x_i y_i^2 + c \sum_{i=1}^n x_i^2 + d \sum_{i=1}^n x_i y_i + e \sum_{i=1}^n x_i = - \sum_{i=1}^n x_i^3 \\ a \sum_{i=1}^n x_i y_i^2 + b \sum_{i=1}^n y_i^3 + c \sum_{i=1}^n x_i y_i + d \sum_{i=1}^n y_i^2 + e \sum_{i=1}^n y_i = - \sum_{i=1}^n x_i^2 y_i \\ a \sum_{i=1}^n x_i y_i + b \sum_{i=1}^n y_i^2 + c \sum_{i=1}^n x_i + d \sum_{i=1}^n y_i + e \sum_{i=1}^n 1 = - \sum_{i=1}^n x_i^2 \end{cases}$$

...Eq.12

## 3) 瞳孔縁の検出

2) で検出した瞳孔は、縮小画像を用いて検出しているので精度が低い。そこで、再度瞳孔の検出を行う。2段階で検出する理由は、画像が蓄積されているメモリに対し、少ないアクセスで測定することができるので、高速に処理できるという利点がある。2) で算出した楕円の近傍を探査し、閾値の近傍で画素の明るさの変化が最大の点を瞳孔の縁として座

標を記録する。

## 4) 楕円近似

3) で記録した座標に楕円の縁で無いもの、具体的には目蓋、睫、照明光の映り込みなど、が入っている場合がある。これらを含んだまま、楕円近似を行うと、正確に瞳孔を検出することができない（Fig.11）。そこで、楕円の幾何学的な性質を利用し、瞳孔の縁以外を除去する。

楕円には、次のような幾何学的な性質がある。楕円と交わる3本の平行な直線 $l, m, n$ を考える。

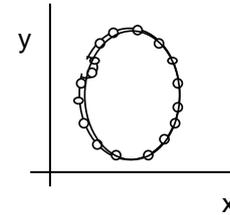


Fig.11 Ellipse fitting include errors.

$l, n$ は $m$ から等距離にあるとする。楕円と直線 $l$ との交点を $a, b$ 、直線 $n$ の交点を $c, d$ とする。楕円と直線 $m$ の交点の中点を $o$ とする。

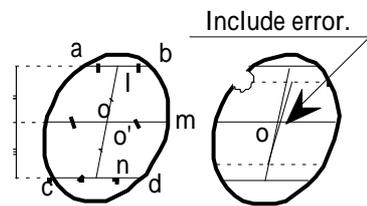


Fig.12 Error reduction method.

$a, b$ の中点と $c, d$ の中点を結ぶ線の中点を $o'$ とすると、 $o'$ は $o$ と重なるという性質がある（Fig.12）。瞳孔の画像について瞳孔画像のほぼ中央に引いた直線 $m$ から等距離にある $N$ 個の平行線の組の $o'_i$  ( $i=1 \sim N$ )に当たる点を求める。求められた点は、直線 $m$ の上に分布する。誤って検出された縁が十分少ないときは、最も多くの点 $o'_i$ が集まった位置が $o$ の位置に相当する。その位置から離れている点は、楕円の軌跡上にはない点を含んでいることになるので、それらを除外する（Fig.12）。どのくらい離れていたら、除外するかは、プログラムで指定することにより、許容範囲の設定が可能になる。4点が相補的に誤差を含んでしまった場合には、除外されなくなるが、確率的にそのようなことは起こりにくい。 $o$ に十分近い $o'_i$ を持つ点 $a_i, b_i, c_i, d_i$ は、楕円の軌跡上の点なので、これらの点の座標をEq.11に代入し正確な楕円近似を行うことができる。

## 5) 各値の算出

瞳孔中心座標 $(X_c, Y_c)$ は、楕円の式の係数より次の式で求めることができる。

$$(X_c, Y_c) = \left( \frac{aY_c + c}{2}, \frac{ac - 2d}{4b - a^2} \right) \dots \text{Eq.13}$$

校正に必要な、短軸の傾き、短軸の長さ、長軸の長さ（瞳孔径）は、次のように表される。 $C1$ を以下の式で定義する。

$$C1 = X_c^2 + aX_c Y_c + bY_c^2 + cX_c + dY_c + e \dots \text{Eq.14}$$

$a = 0$ の場合。

$$a_1 = \frac{b-1}{a} + \sqrt{\left(\frac{b-1}{a}\right)^2 + 1}, a_2 = \frac{b-1}{a} - \sqrt{\left(\frac{b-1}{a}\right)^2 + 1} \dots \text{Eq.15}$$

$$r_1 = \sqrt{\frac{-C_1(a_1^2+1)}{ba_1^2+aa_1+1}}, r_2 = \sqrt{\frac{-C_1(a_2^2+1)}{ba_2^2+aa_2+1}} \dots \text{Eq.16}$$

$r_1 > r_2$ ならば、  
 短軸= $r_2$ 、瞳孔半径= $r_1$ 、短軸の傾き= $a_2$   
 $r_1 < r_2$ ならば、  
 短軸= $r_1$ 、瞳孔半径= $r_2$ 、短軸の傾き= $a_1$   
 $a=0$ の場合、

$$r_1 = \sqrt{-C_1}, r_2 = \sqrt{\frac{-C_1}{b}} \dots \text{Eq.17}$$

$r_1 > r_2$ ならば、  
 短軸= $r_2$ 、瞳孔半径= $r_1$ 、短軸傾き=  
 $r_1 < r_2$ ならば、  
 短軸= $r_1$ 、瞳孔半径= $r_2$ 、短軸傾き= $0$   
 となる。

#### 4. 精度の検証

NTSCの出力を持つCCDカメラから画像をデジタル化するビデオ入力装置を用いると、画像の解像度は、640x480画素になる。この解像度は、フレームにおける解像度である。フレームは、偶数フィールドと奇数フィールドから成り立っている。動きの早い画像の場合、両フィールドの像は、異なりフレーム画像にぶれが生じる。そこで、処理はフィールド毎に行った方が、良い結果が得られる。フィールドの解像度は、フレームの半分の 640x240画素となる。縦横比が3:4であるとき、実際の画像の縦横比と同じになるので、ここでは、320x240画素以下の解像度における精度の検証を行った。

##### 1) 理論的精度の検証

Fig.9の視標を見ている時の瞳孔画像をC.G.により作成した (Fig.13)。実際の使用状況に近くなるようにカメラの位置は正面から45度の位置に設置した。64x48、96x72、128x96、160x120、192x144、224x168、256x192、288x216、320x240の大きさの画

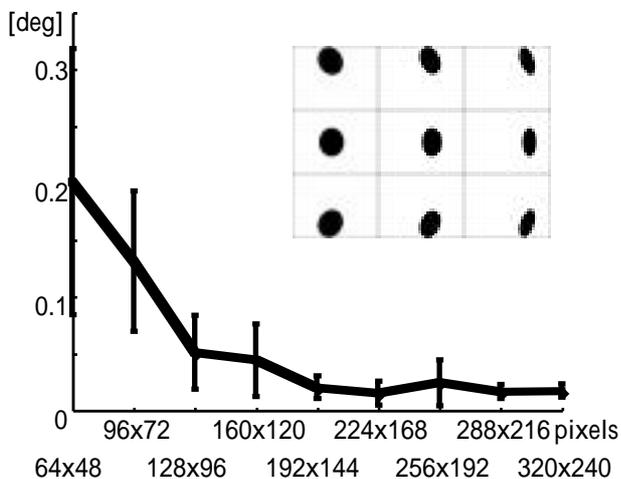


Fig.13 Relation between resolution and accuracy.

像を9枚ずつ作成した。これらの、像を入力として、眼球回転中心、瞳孔回転半径、変換行列を求める。次に、それぞれの像の瞳孔中心から、対象物に対する視線ベクトルを求める。求められた視線ベクトルと、実際の対象物に対するベクトルのなす角を誤差とする。各解像度における誤差と標準偏差を測定した。9点における誤差の平均と標準偏差を示す (Fig.13)。グラフより、128x96画素の解像度があれば、0.1度以下の精度で測定できることがわかる。

##### 2) 実際の場合における精度の検証

被験者に、Fig.9の視標を番号順に見てもらい、その時の眼球映像をビデオに記録する。同一のビデオ画像を用いて、解像度による精度の違いを検証する。画像を取り込む解像度を変更し、それぞれの解像度における、眼球回転中心、瞳孔回転半径、変換行列を求める。求められた値を使い視標を見ている時の瞳孔中心座標から対象物に対する視線ベクトルを求める。視線ベクトルと、実際の対象物に対するベクトルのなす角を誤差とする。各解像度における誤差の平均と標準偏差をFig.14に示す。

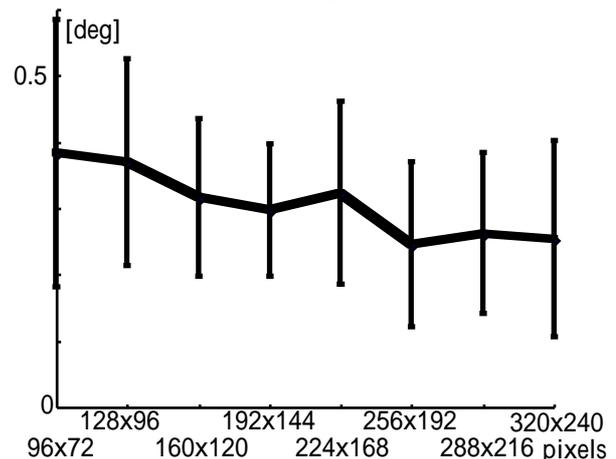


Fig14 Relation between resolution and accuracy.

実際の計測においては、頭部の微動などの要素も入るために、理論的精度よりも誤差が大きい。しかし、160x120画素以上あれば、0.5度以下の精度で測定できることが分かる。

##### 3) 瞳孔の欠損と精度

瞳孔がまぶた隠れてしまった場合、どの程度まで精度が保証できるかをC.G.画像を用いて検証する。Fig.13で用いた画像を、上部から5%ずつ隠していった画像を作成する (Fig.15)。画像の大きさは、320x240画素とする。1)、2)と同様の方法でそれぞれの画像の組について誤差と標準偏差を求めた。結果をFig.15に示す。瞳孔がまぶたに隠れるような場合でも欠損が30%以下の場合、0.1度以下の精度で視線位置が検出できることがわかる。

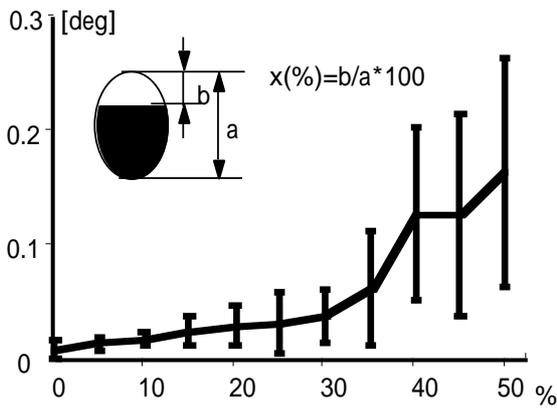


Fig.15 Errors of partly hidden pupil images.

#### 4) 分解能

絶対精度では無く、どのくらい瞳孔が動いたら変化が検出できるか、という分解能について検証を行う。楕円の縁を20点以上用いて計算すると、分解能は、0.05画素以下になる。2)において、平均で1画素の変化が0.5度に相当するので、分解能は、0.025度以下と考えられる。

#### 5. 実装における問題点

これまでの計算においては、実際の光景と入力画像の縦横比が同一であること前提としてた。しかし、レンズ、CCDカメラ、ビデオ入力装置などにより、縦横比が変化し、入力画像が実際より縦長になったり、横長になってしまったりする場合がある。縦横比が異なると、正確な測定ができなくなってしまいう例を次に提示する。訓練を行ったサルに、5つの視標を見せ、その際の眼球画像を撮影する。得られた画像の瞳孔を楕円で近似し、楕円の中心と短軸の傾きを求める。楕円の中心から短軸を延長する。入力する画像の縦横比を変えると正確に眼球回転中心を求めることができないことが分かる (Fig.16)。

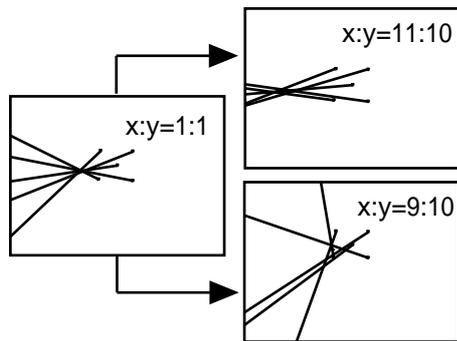


Fig.16 Calculate eyeball center in each ratio.

縦横比を補正する方法は2通りある。実験環境設定時に一度行えば良いので、場合により適切な方法を選ぶ。

1) 実験を行なう前に、真球であると考えられる適度な大きさの物体を撮影し、得られた画像の縦横比から補正值を求める。

2) 縦横比が1:1の場合は、楕円の短軸の延長線が1

点で交わるということを利用し、最適の縦横比を計算する。被験者に、視覚刺激を提示し瞳孔画像を撮影し、瞳孔を楕円で近似する。画像毎に楕円の短軸の傾き、瞳孔中心座標を求める。最小自乗法により眼球回転中心を求める。短軸の延長線と眼球回転中心との距離を誤差とする。縦横比を変化させ誤差をプロットする (Fig.17)。誤差が最小となる縦横比を補正值とする。この方法は、校正と同時に行うことにより、自動的に縦横比の補正を行うことができる。また、瞳孔画像しかない場合にも対応することができる。

Error[pixels]

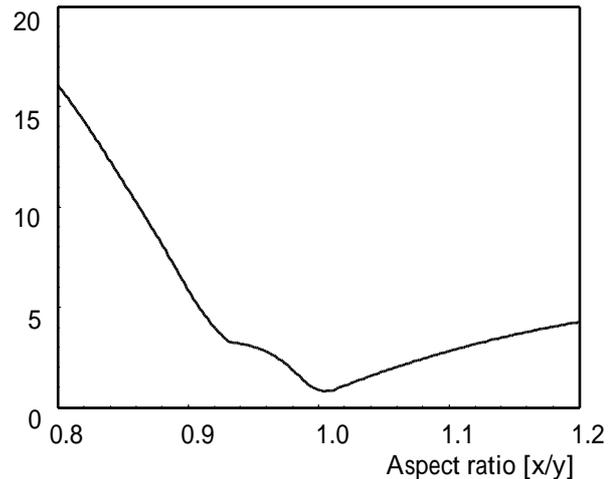


Fig.17 Relation between error and aspect ratio.

#### 6. システム例

##### 1) 光学系

45mmのマクロレンズを使用。目とレンズ間での距離は15cm。CCDカメラには、Sony XC-75を用いた。カメラ内部には、赤外線フィルターがあるためこれを取り除いて使用した。

##### 2) コンピュータの構成

使用したOSの違いによりa),b)二つのシステムがある。

##### a)MS-DOS上のシステム

コンピュータ:PC-9821Xa10/C12 (Pentium 100MHz)

ビデオ入力装置:カノーブスSuperCVI

OS:日本語MS-DOSを使用

シングルタスクOSなので、リアルタイム性が保証できることがメリットであるが、今後ソフトウェア、ハードウェアの入手が困難になることが予想される。オプションとして、デジタルアナログコンバータボードを装着することにより、視線位置をアナログ電圧値の変化として出力することができる。また、デジタルIOボードを装着することにより、視線がある領域に入った時に信号を出すことができる。取込み解像度は、320x240、取込み速度は30Hzである ( Fig.18)。

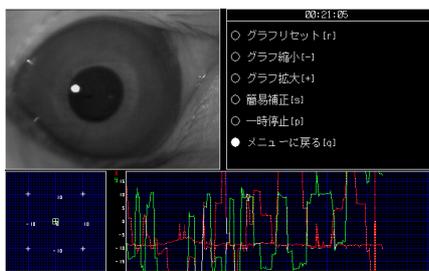


Fig.18 A system for measuring eye positions (MS-DOS)

#### b)Unix互換OS上のシステム

##### ノートブックシステム

コンピュータ：Panasonic CF-A77(celeron300MHz)

ビデオ入力装置：IBM SmartCaptureCardII ( PC card )

OS：FreeBSD3.2R

PCカード用デバイスドライバは

( [http://www.mickey.ai.kyutech.ac.jp/~ohashi/scc/index\\_j.html](http://www.mickey.ai.kyutech.ac.jp/~ohashi/scc/index_j.html) ) を用いた。取込み解像度は、

160x120、取込み速度 < 30Hz。

##### デスクトップシステム

コンピュータ：EpsonDirect VZ-5000(PentiumIII 700MHz)

ビデオ入力装置：Interware Picpom Linux ( PCI )

OS：Linux6.1J(Linux2.2.12)

デバイスドライバーには、カーネル2.2以降に標準で添付されているVideo for Linux を用いた。上記のビデオ入力装置以外でもVideo for Linuxに適合していれば使用可能である。取込み解像度320x240、取込み速度は、30Hzまたは60Hz。DAボードによる視線位置電圧出力、DIOボードによる視線スイッチ信号出力が可能である。

UNIX互換OSにおいては、デバイスドライバにアクセスする部分のみ異なるファイルとなっている。このファイルを変更することで、異なる画像入力方法に対応できる。画像表示にはX-Window systemを用いた ( Fig.19 )。

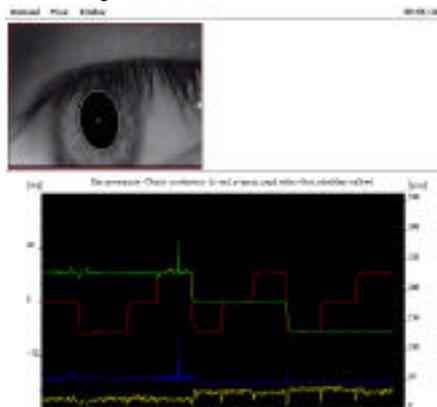


Fig.19 A system for measuring eye positions (Linux)

#### 7. まとめ

我々は、画像処理による無侵襲視線位置計測法を開発した。シミュレーションにより、画素数が160x120以上あれば、0.1度以下の精度があることが

分かった。実際の使用において画素数が160x120以上あれば、0.5度以下の精度で計測できることが分かった。シミュレーションにより、画素数が320x240あれば、瞳孔が目蓋などで30%隠れても、精度が落ちないことを示した。画素数が320x240かつ現在のマクロレンズを使用時に、分解能が0.025度以下であることを示した。現在はNTSC仕様のCCDカメラを利用しているので、時間分解能は60Hzである。しかし、画像処理時間2ms以下であるので、高速な画像入力手段があればさらに、時間分解能をあげることが可能である。Linuxは、リアルタイムOSでないため、処理落ちの可能性はあるが、処理速度が十分早いため現時点では、取りこぼしは発生して無い。視線位置計測以外に、サルにおいてはfixation task ( 凝視訓練 ) を学習する際に使用することができる。人間においては、視線スイッチとして使用することができる。今後は、採取したデータの解析プログラムを作成する予定である。本プログラムは、<http://www.etl.go.jp/~matsuda/eye/>にて、公開している。

#### 参考文献

[1]松田圭司、山根茂：「慢性実験サルの視線位置を計測する簡便なシステム」、第8回生体・生理工学シンポジウム論文集、p5-p10、(1993)

[2]松田圭司、山根茂：「高速楕円近似による視線位置計測システム」、第9回生体・生理工学シンポジウム論文集、p127-p130、(1994)

[3]松田圭司、永見武司、山根茂：「高速楕円近似による汎用視線位置計測システム」、第10回生体・生理工学シンポジウム論文集、p401-404、(1995)

[4]松田圭司、永見武司：「閾値自動検出方式を用いた視線位置計測システム」、第11回生体・生理工学シンポジウム論文集、p161-164、(1996)

[5]松田圭司、永見武司：「汎用視線位置計測システム」、第12回生体・生理工学シンポジウム論文集、p173-p176、(1997)

[6]松田圭司、永見武司：「異機種実装を考慮した視線位置計測システムの開発」、第13回生体・生理工学シンポジウム論文集、p253-p256、(1998)

[7]松田圭司、永見武司：「隔操作に対応した視線位置計測システムの開発」、第14回生体・生理工学シンポジウム論文集、p57-p60、(1999)

[8]松島純一、熊谷雅彦、伊福部達、原田千洋：「固体撮像素子(CCD)イメージセンサを用いた眼球運動記録(CCD-EOG)について」、Equilibrium Res. Vol.47 (2);169-173、(1988)

[9]伴野明：「視線検出のための瞳孔撮影光学系の設計法」、電子情報通信学会論文誌 Vol..J74-D-II No.6 736-747、(1991)

[10]伴野明、岸野文郎：「顔と瞳孔の3次元位置計測に基づく注視点検出アルゴリズム」、電子情報通信学会論文誌 Vol..J75-D-II No.5 861-872、(1992)

[11]伴野明、岸野文郎、小林幸雄：「瞳孔抽出と頭部の動きを許容する視線検出装置の試作」、電子情報通信学会論文誌 Vol..J76-D-II No.3 636-646、(1993)