法政大学学術機関リポジトリ
HOSEI UNIVERSITY REPOSITORY

PDF issue: 2025-07-16

顔の動きを利用したユーザインタフェース

寺井, 大樹 / TERAI, Daiki

(発行年 / Year)
2012-03-24
(学位授与年月日 / Date of Granted)
2012-03-24
(学位名 / Degree Name)
修士(工学)
(学位授与機関 / Degree Grantor)
法政大学 (Hosei University)

２０１２年度　修士論文

顔の動きを利用した
ユーザインタフェース

指導教員 平原　誠
システム工学専攻修士課程
寺井　大樹

概要

近年、デジタル化に伴い、様々なユーザインタフェースが登場するようになった。３次元コンピュータビジョン技術の発展に伴い、画像解析による非接触型のユーザインタフェースを導入した製品が多く存在する。例えば、Kinect、Air Swing などが挙げられる。本研究では、動画像から顔の平行移動・回転情報を取得し、それをユーザインタフェースに用いる。まず、Haar-like 特徴を用いて作成した弱識別器から AdaBoost に基づいて強識別器を構成し、それを利用して、顔および目、鼻領域の検出を行う。次に、目、鼻の各領域に対し、Harris コーナー検出を行い、特徴点を算出する。最後に、特徴点の追跡情報を用いて顔の平行移動・回転情報を取得する。手を使わずに顔の動きを利用してコンピュータを制御することができ、３D アプリケーション、あるいはドライブシミュレータの視点操作などに応用できる。

Abstract

In recent years, various user interfaces came to appear with digitization. With development of three-dimensional computer vision technology, many products which introduced the noncontact user interface by image analysis exist. For example, Kinect, Air Swing, etc. are mentioned. In this research, the parallel translation and the rotation information on a face are acquired from video, and it is used for a user interface.First, based on AdaBoost, a strong classifier is constituted from a weak classifier created using the Haar-like feature, and detection of a face, eyes, and a nose region is performed using it. Next, to each region of eyes and a nose, Harris corner detection is performed and features are computed.Finally, the tracking information on features is used and the parallel translation and the rotation information on a face are acquired. This system can control a computer using motion of a face, without using hands, and can apply it to viewpoint operation of 3D application or a drive simulator etc.

目次

１．はじめに
２．手順
３．領域検出
3.1 顔領域の検出
3.2 目、鼻領域の検出
3.3 AdaBoost の性質
４．特徴点検出
4.1 目、鼻領域の特徴点検出
4.2 Harris コーナー検出の性質
５．特徴点追跡
６．3 軸の設定
７．平行移動情報の取得
7.1 X 軸、Y 軸平行移動情報の取得
7.2 Z 軸平行移動情報の取得
８．回転情報の取得
8.1 X 軸、Y 軸回転情報の取得
8.2 Z 軸回転情報の取得
９．ユーザインタフェースの実装
１０．考察
１１．ユーザインタフェースの改良
１２．終わりに
１３．参考文献

１．はじめに

近年、デジタル化に伴い、画像処理を用いたユーザインタフェースが登場するようになった。代表的な製品としては、画面に触れることなくコンピュータ上のボタンをタッチできる Air Swing、カメラから体の姿勢情報を読み取り、操作に利用する Kinect が挙げられる。本研究では、動画像から顔を追跡し、顔の動きを利用したユーザインタフェースを開発する。手を使わずにアプリケーションを操作できる直観的なユーザインタフェースであり、３D アプリケーションや、ドライブシミュレータなどの操作などに応用できる。

２．手順

本研究は顔の動きの追跡のため、次の６つの手順を踏んでいる。
STEP1：顔領域の検出
STEP2：目、鼻領域の検出
STEP3：目、鼻領域の特徴点検出
STEP4：特徴点の追跡
STEP5：平行移動情報の取得
STEP6：回転情報の取得

３．領域検出

３．１ 顔領域の検出

顔領域の検出には Haar-like 特徴と AdaBoost を用いた[1][2]。
（１） AdaBoost
AdaBoost はパターン識別のための学習アルゴリズムである。実装が容易で高い識別性能が得られる利点がある。逐次的に学習サンプルの重みを変化させながら異なる識別器を作り、これら複数の識別器の重み付き多数決によって最終的な識別関数を与える。個々の識別器は弱識別器(weak classifier)あるいは弱仮説(weak hypothesis)と呼ばれ、それらを組み合わせたものは強識別器(strong classifier)あるいは最終仮設(final classifier)などと呼ばれる。単純で弱い識別器を逐次的に学習し、識別器の精度を増強(Boost)する方法を総称して Boosting と呼ぶ。AdaBoost は適応的(adaptive)にサンプルの重みを更新し、学習を行う Boosting アルゴリズムの一種である。本研究では顔画像、非顔画像のサンプルを用いて 2 クラス分類（顔かそうでないかの判定）の強識別器$\mathrm{H}(x)$を作成する。入力xに対し、判別精度が低い２クラス識別器$h_m(x)$を選別し、信頼度α_mで重みづけし、M 個線形結合する。重み付け線形結合した関数$F(x)$は式(1)で表される。

$$F(x) = \sum_{m=1}^{M} \alpha_m h_m(x) \qquad (1)$$

この$F(x)$の符号$\mathrm{H}(x)$でクラスを決定する。

$$\mathrm{H}(x) = \mathrm{sign}\,[F(x)] \qquad (2)$$

ここで sign は符号関数であり、2 値の出力を出す。結果が正ならば顔、負ならば非顔と判定する。AdaBoost の学習の目的はα_mと$h_m(x)$を決定することである。実際には、$h_m(x)$が決まればα_mも決まるので$h_m(x)$をいかに決定するかに帰着される。基本的に$h_m(x)$は自由に設計してよく、満たすべき条件は以下のとおりである。

(i)　　$h_m(x)$は、$\mathrm{H}(x)$と同様に+1 もしくは-1 の２値を出力する関数とする。
(ii)　　$h_m(x)$は、誤り率が 0.5 より小さくなければならない
条件(i)は２クラス分類のためのものであり、満たすことは容易である。条件(ii)は２クラス識別器の場合は、容易に満たすことができる。２クラス識別を行う場合、でたらめでも識別結果が間違いとなる確率（誤り率）はたかだか 0.5 である。誤り率が 0.5 を超えた場合は$h_m(x)$の判定を反転すればよい。弱識別器$h_m(x)$の作成には Haar-like 特徴を用いた。
（２） Haar-like 特徴
Haar-like 特徴は白領域と黒領域を組み合わせた特徴であり、図１のようなエッジ検出用、ライン検出用、コーナー検出用に分類できる。

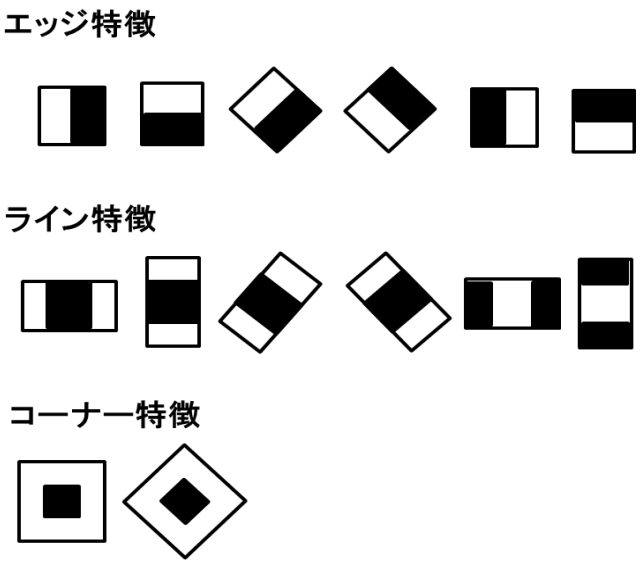


図１　基準 Haar-like 特徴

これらの基準 Haar-like 特徴を元に位置と大きさを変化させることで多数の特徴が生成できる。図２のように検索対象の画像に対し、特徴を重ね合わせ、重ね合わせた領域に対して輝度値に基づく特徴量を算出する。

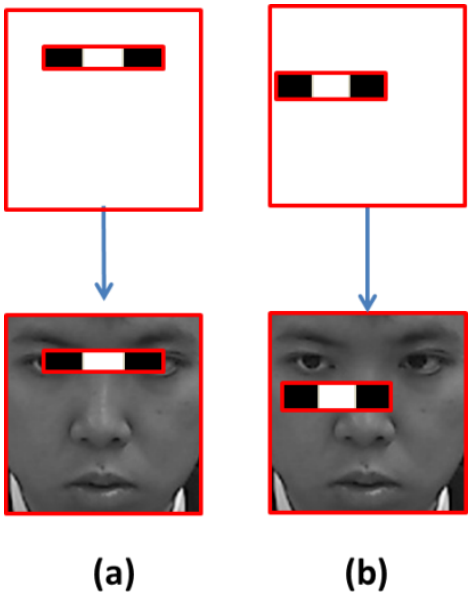


図２　Haar-like 特徴の適用

特徴の白領域における画像の輝度値の平均をS_1、黒領域における画像の輝度値の平均をS_2として特徴量xは次式で与えられる。

$$x = S_1 - S_2$$

例えば、図２**(a)**のように Haar-like 特徴を適用したとき、目の領域をうまく捉えているので特徴量は大きくなる。それに対し、図２**(b)**のように Haar-like 特徴を適用したとき、目の領域ではないので特徴量は小さくなることがわかる。各特徴に対し、算出した特徴量xが閾値θを超えていれば検索対象画像に顔が存在すると判定し、超えていなければ存在しないと判定する２クラス分類の弱識別器$h(x)$を作成する。弱識別器$h(x)$は次式で与えられる。

$$h(x) = \begin{cases} 1 & if\ \ px > p\theta \\ -1 & otherwise \end{cases}$$

ここで、符号pは＋１か－１の２値を取り、特徴量xと閾値θの不等号を調整するための値である。顔画像、非顔画像の全サンプルに対し、最も正解率が高くなるように閾値θと符号pを決定する。例えば、閾値θを全サンプルの最小値から始め、１ずつ値を大きくしていき、正解数の増加と減少が切り替わるところを閾値θとする。画像の輝度値の和は計算量が大きいため、積分画像(Integral Image あるいは Summed Area Table)と呼ばれる画像を作成し、計算に利用する。積分画像は下図のように着目画像座標よりも左上の輝度値を全て足し合わせることで作成する。

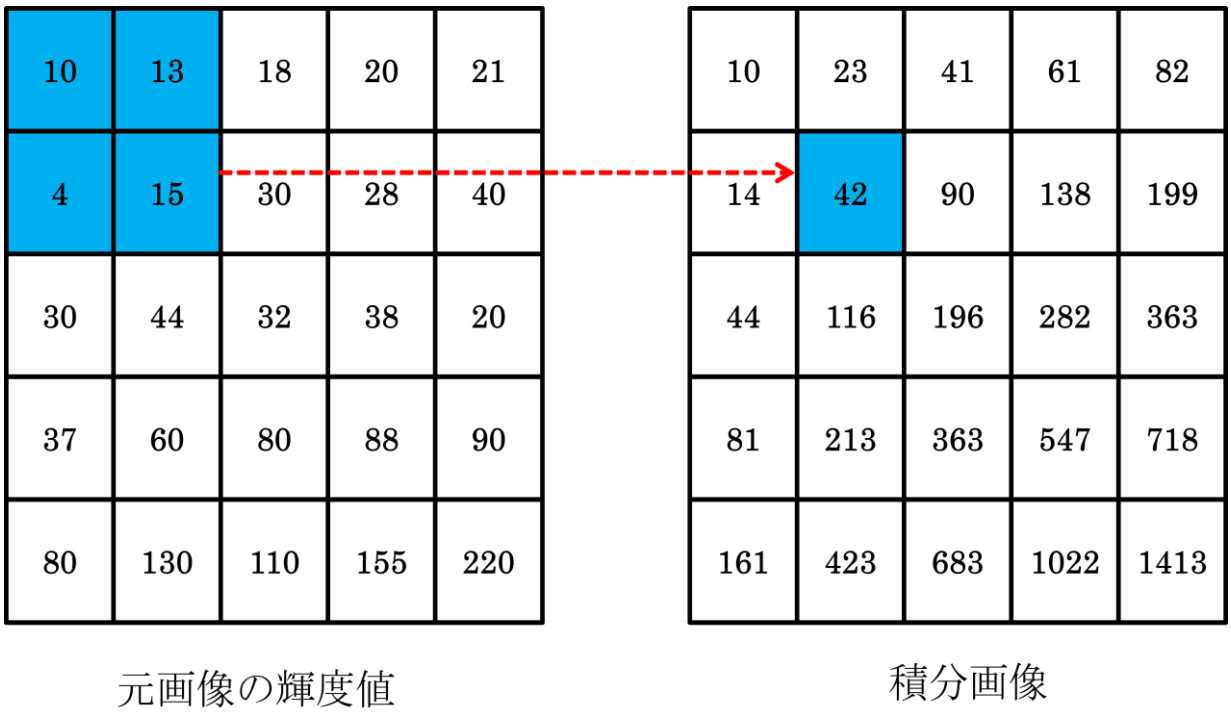


図３　積分画像

具体的には、積分画像の画像座標(x, y)における値$\mathrm{ii}(x, y)$は輝度値$\mathrm{I}(x, y)$を用いて次式で与えられる。

$$\mathrm{ii}(x, y) = \sum_{i=0}^{x} \sum_{j=0}^{y} \mathrm{I}(i, j)$$

下図のような領域Ψが与えられた場合、

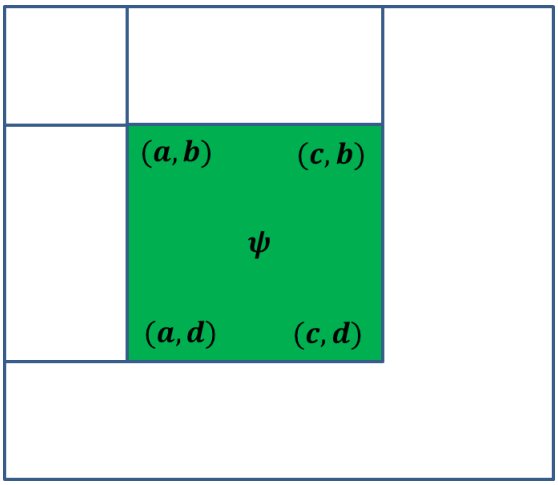


図４　領域Ψ

領域Ψの左上の座標(a,b)、右下の座標(c,d)とする。
下図のような位置の輝度和 A,B,C,D が与えられた場合

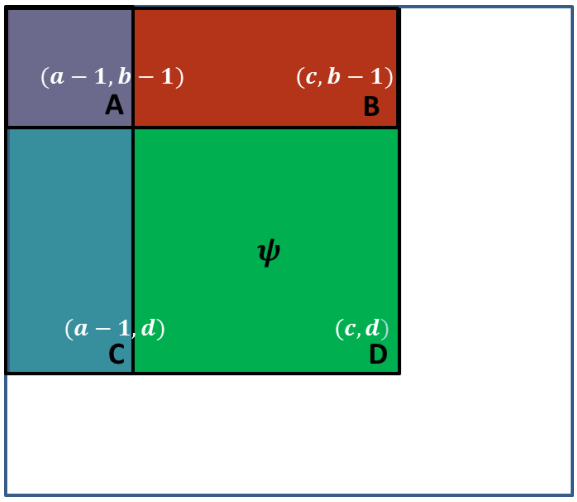


図５　輝度和

領域Ψの輝度和Sは

$$\mathrm{S} = \mathrm{D} - \mathrm{B} - \mathrm{C} + \mathrm{A}$$

となり、これは、積分画像$\mathrm{ii}(x,y)$を用いて次のように計算できる。

$$\mathrm{S} = \mathrm{ii}(c,d) - \mathrm{ii}(c,b-1) - \mathrm{ii}(a-1,d) + \mathrm{ii}(a,b)$$

このように、一度積分画像を作成してしまえば、対応する積分画像の 4 箇所の位置から輝度和を算出することができ、たかだか 4 回の参照で済むため大幅に計算量を削減できる。
AdaBoost を用いて、多数の弱識別器から顔検出に有効な弱識別器を選別し、その信頼度α_mを学習する。N 個の学習サンプル$(\mathrm{x}_1,\mathrm{y}_1),\dots,(\mathrm{x}_N,\mathrm{y}_N)$が与えられているとする。
全ての学習サンプルは予め画像サイズを正規化しておく。$\mathrm{y}_i \in \{-1,+1\}$は i 番目のサンプル$\mathrm{x}_i$が属するクラスのラベルである。

(A) 添え字 m=1 とする。

サンプルの重み分布$D_m(i)$ $(i = 1, \dots, N)$を$\frac{1}{N}$で均等に初期化する。

(B) D_m において、弱識別器$h_s(x)$ $(s = 1, \dots, M)$に対し、次式で与えられる誤り率ε_sを求める。

$$\varepsilon_s = \sum_{i:y_i \neq h_s(x_i)} D_m(i)$$

ε_sが 0.5 を超えた場合、$h_s(x)$の判定符号pを逆にすることでε_sが 0.5 を超えることはない。mにおいてε_sが最小となる s を導く$\delta(m)$を次式で定義する。

$$\delta(m) = \arg\min_s \varepsilon_s$$

これは最小の誤り率$\boldsymbol{\varepsilon}_{\delta(m)}$となる$h_{\delta(m)}(x)$を選択したことを意味する。

(C) 誤り率$\boldsymbol{\varepsilon}_{\boldsymbol{\delta(m)}}$から式(3)で与えられる$h_{\delta(m)}(x)$に対する信頼度$\alpha_m$を算出する。

$$\alpha_m = \frac{1}{2}\log\left(\frac{1 - \boldsymbol{\varepsilon}_{\delta(m)}}{\boldsymbol{\varepsilon}_{\delta(m)}}\right) \qquad (3)$$

なお、$\varepsilon_{\delta(m)} \leq 0.5$なので $\alpha_m > 0$となる

(D) 全サンプルの重みを次式に基づき更新する。

$$D_m'(\iota) \leftarrow D_m(i) e^{-\alpha_m y_i h_{\delta(m)}(x_i)}$$

選択した弱識別器$h_{\delta(m)}$が正しく識別できたサンプルについては$y_i h_{\delta(m)}(x_i) = 1$ であるから、重みが小さくなる。間違えたサンプルについては$y_i h_{\delta(m)}(x_i) = -1$ であるから、重みが大きくなる。つまり、$h_{\delta(m)}(x)$の誤り率が高くなるような重み分布に更新される。$\boldsymbol{\varepsilon}_{\delta(m)}$= 0.5 の場合、$\alpha_m$= 0 となり、分布は更新されず、(A)で同じ弱識別器が選ばれてしまうので学習を継続する意味がない。従って、学習を終了する。

(E) 次式のようにサンプルの重みの和が 1 になるように重み分布$D_m'(\iota)$を正規化する。

$$D_{m+1}(i) = \frac{D_m'(\iota)}{Z_m}$$

ここでサンプルの重み和Z_mは次式で与えられる。

$$Z_m = \sum_{i=1}^{N} D_m(i)\, e^{-\alpha_m y_i h_{\delta(m)}(x_i)}$$

m = m+1 として、m < M ならば(B)に戻る。m = M ならば学習を終了する。

なお、更新後の分布D_{m+1}における弱識別器$h_{\delta(m)}$の誤り率は 0.5 となることが知られている。以下はその証明である。Z_mは次のように変形できる

$$Z_m = \sum_{i=1}^{N} D_m(i)\, e^{-\alpha_m y_i h_{\delta(m)}(x_i)} = \sum_{i:y_i = h_{\delta(m)}(x_i)}^{N} D_m(i)\, e^{-\alpha_m} + \sum_{i:y_i \neq h_{\delta(m)}(x_i)}^{N} D_m(i)\, e^{\alpha_m}$$

ここで

$$\sum_{\mathrm{i}:y_i=h_{\delta(m)}(x_i)}^{\mathrm{N}} \mathrm{D_m(i)} = 1 - \boldsymbol{\varepsilon}_{\delta(m)}$$

$$\sum_{\mathrm{i}:y_i\neq h_{\delta(m)}(x_i)}^{\mathrm{N}} \mathrm{D_m(i)} = \boldsymbol{\varepsilon}_{\delta(m)} \qquad (4)$$

$$\alpha_m = \frac{1}{2}\log\left(\frac{1-\boldsymbol{\varepsilon}_{\delta(m)}}{\boldsymbol{\varepsilon}_{\delta(m)}}\right) \qquad (5)$$

であるので式(6)となる

$$\mathrm{Z_m} = \left(1-\boldsymbol{\varepsilon}_{\delta(m)}\right)e^{-\alpha_m} + \boldsymbol{\varepsilon}_{\delta(m)}e^{\alpha_m}$$

$$= \left(1-\boldsymbol{\varepsilon}_{\delta(m)}\right)e^{-\frac{1}{2}\log\left(\frac{1-\boldsymbol{\varepsilon}_{\delta(m)}}{\boldsymbol{\varepsilon}_{\delta(m)}}\right)} + \boldsymbol{\varepsilon}_{\delta(m)}e^{\frac{1}{2}\log\left(\frac{1-\boldsymbol{\varepsilon}_{\delta(m)}}{\boldsymbol{\varepsilon}_{\delta(m)}}\right)} \qquad (6)$$

ここで、

$$V = e^{\frac{1}{2}\log\left(\frac{1-\boldsymbol{\varepsilon}_{\delta(m)}}{\boldsymbol{\varepsilon}_{\delta(m)}}\right)}$$

とおいてこれの両辺対数を取ると

$$\log V = \frac{1}{2}\log\left(\frac{1-\boldsymbol{\varepsilon}_{\delta(m)}}{\boldsymbol{\varepsilon}_{\delta(m)}}\right)$$

$$\mathrm{V} = \sqrt{\frac{1-\boldsymbol{\varepsilon}_{\delta(m)}}{\boldsymbol{\varepsilon}_{\delta(m)}}}$$

より

$$e^{\frac{1}{2}\log\left(\frac{1-\boldsymbol{\varepsilon}_{\delta(m)}}{\boldsymbol{\varepsilon}_{\delta(m)}}\right)} = \sqrt{\frac{1-\boldsymbol{\varepsilon}_{\delta(m)}}{\boldsymbol{\varepsilon}_{\delta(m)}}} \qquad (7)$$

同様に

$$e^{-\frac{1}{2}\log\left(\frac{1-\boldsymbol{\varepsilon}_{\delta(m)}}{\boldsymbol{\varepsilon}_{\delta(m)}}\right)} = \sqrt{\frac{\boldsymbol{\varepsilon}_{\delta(m)}}{1-\boldsymbol{\varepsilon}_{\delta(m)}}}$$

より$\mathrm{Z_m}$は式(8)のように変形できる。

$$\mathrm{Z_m} = \left(1-\boldsymbol{\varepsilon}_{\delta(m)}\right)\sqrt{\frac{\boldsymbol{\varepsilon}_{\delta(m)}}{1-\boldsymbol{\varepsilon}_{\delta(m)}}} + \boldsymbol{\varepsilon}_{\delta(m)}\sqrt{\frac{1-\boldsymbol{\varepsilon}_{\delta(m)}}{\boldsymbol{\varepsilon}_{\delta(m)}}} = 2\sqrt{(1-\varepsilon_{\delta(m)})\varepsilon_{\delta(m)}} \qquad (8)$$

したがって分布D_{m+1}下での$h_{\delta(m)}(x)$の誤り率$\boldsymbol{\varepsilon}_{\delta'(m)}$は(4)(5)(7)(8)より次式で与えられる。

$$\boldsymbol{\varepsilon}_{\delta'(m)} = \sum_{\mathrm{i}:y_i\neq h_{\delta(m)}(x_i)}^{\mathrm{N}} \mathrm{D_{m+1}(i)} = \frac{\sum_{\mathrm{i}:y_i\neq h_{\delta(m)}(x_i)}^{\mathrm{N}} \mathrm{D_m(i)}e^{\alpha_m}}{\mathrm{Z_m}} = \frac{\sqrt{(\mathbf{1}-\boldsymbol{\varepsilon}_{\delta(m)})\boldsymbol{\varepsilon}_{\delta(m)}}}{2\sqrt{(\mathbf{1}-\boldsymbol{\varepsilon}_{\delta(m)})\boldsymbol{\varepsilon}_{\delta(m)}}} = 0.5$$

これは、更新後の分布$\mathrm{D_{m+1}}$は$h_{\delta(m)}(x)$が最も判別能力が低くなる分布であり、次に選択される弱識別器$h_{\delta(m+1)}(x)$は$\mathrm{D_{m+1}}$に対し、最も識別能力が高いものになることを意味する。つまり、AdaBoost は最も強い弱識別器を最初に選び、それが苦手なサンプルを補う形で別の弱識別器を追加していき、強識別器を作成する学習アルゴリズムである。このようにして作成された強識別器を用いて顔領域の判定を式(2)に基づいて行う。

２．２　目、鼻領域の検出

（１）右目の画像サンプルに対し Haar-like 特徴及び、AdaBoost で右目用の強識別器の作成を行う。同様に左目、鼻に対してもそれぞれの画像サンプルを用いて強識別器の作成を行う。
（２）検出した顔領域に対し、右目検出用の強識別器を用いて右目の検出を行う。同様に左目、鼻の強識別器を用いてそれぞれの領域の検出を行う（図６）。

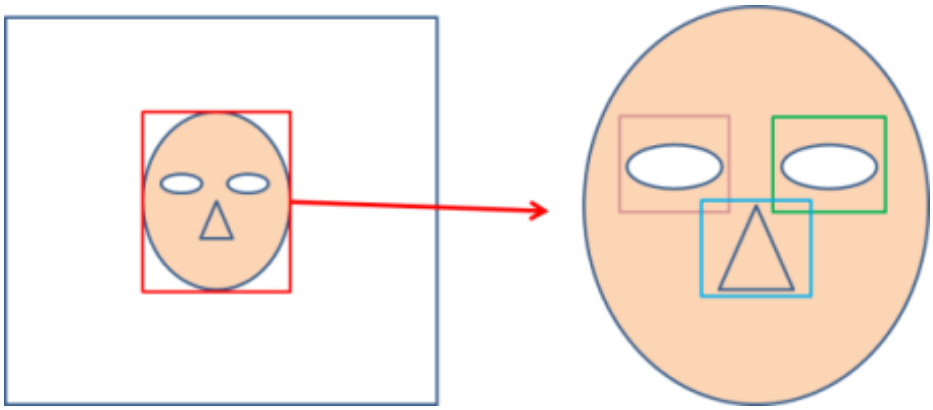


図６　目、鼻領域の検出

２．３　AdaBoost の性質[3]

ここから、AdaBoost の導出について説明する。AdaBoost は指数損失の最小化という概念に基づいて導出されている。指数損失$\mathrm{l}(y, f(x))$は次式で与えられる。

$$\mathrm{l}(y, f(x)) = \mathrm{e}^{-y\mathrm{f}(x)}$$

ここで、$f(x)$は一般の実数関数である。ここで、指数損失の$f(x)$を$\mathrm{F}(x)$とし、$\mathrm{F}(x)$は$k-1$個$(k \leq M)$の重み付けをした弱識別器を線形結合した次式で与えられるものとする。

$$F(x) = \sum_{m=1}^{k-1} \alpha_m h_{\delta(m)}(x)$$

N 個のサンプルの生起確率は一様であると仮定すると、この関数の期待損失$\mathrm{L(F)}$は

$$\mathrm{L(F)} = \sum_{i=1}^{N} \frac{1}{N} e^{-y_i F(x_i)} = \frac{1}{N} \sum_{i=1}^{N} e^{-y_i F(x_i)}$$

と書ける。今、$\mathrm{F}(x)$にk個目の弱識別器$h_k(x)$を重み$\alpha_k(\geq 0)$で１つ加えて

$$\mathrm{F}(x) + \alpha_k h_k(x)$$

とし、次の期待損失$\mathrm{L}(\mathrm{F} + \alpha_k h_k)$を小さくすることを考える。

$$\mathrm{L}(\mathrm{F}+\alpha_k h_k)=\frac{1}{N}\sum_{i=1}^{N} e^{-y_i(F(x_i)+\alpha_k h_k(x_i))}$$
$$=\frac{1}{N}\sum_{i=1}^{N} e^{-y_i F(x_i)} e^{-\alpha_k y_i h_k(x_i)}$$

正解$y_i h_k(x_i)=1$と不正解$y_i h_k(x_i)=-1$に分離できるので

$$=\frac{1}{N}\sum_{i:y_i=h_k(x_i)}^{N} e^{-y_i F(x_i)} e^{-\alpha_k}+\frac{1}{N}\sum_{i:y_i\neq h_k(x_i)}^{N} e^{-y_i F(x_i)} e^{\alpha_k}$$

と変形でき、正解 ＝ 全体 － 不正解なので第一項を書き換えると

$$=\frac{e^{-\alpha_k}}{N}\sum_{i=1}^{N} e^{-y_i F(x_i)}-\frac{e^{-\alpha_k}}{N}\sum_{i:y_i\neq h_k(x_i)}^{N} e^{-y_i F(x_i)}+\frac{e^{\alpha_k}}{N}\sum_{i:y_i\neq h_k(x_i)}^{N} e^{-y_i F(x_i)}$$

となり、

$$=\frac{e^{-\alpha_k}}{N}\sum_{i=1}^{N} e^{-y_i F(x_i)}+\frac{e^{\alpha_k}-e^{-\alpha_k}}{N}\sum_{i:y_i\neq h_k(x_i)}^{N} e^{-y_i F(x_i)} \qquad (9)$$

と変形できる。$\alpha_k(\geq 0)$は自由に決めてよい値なのでここでは定数とみなし、h_kに依存する第２項を考える。ここで、分布$\mathrm{D}_k(\mathrm{i})$はサンプルx_iにおける指数損失を正規化したものであり、$k-1$個目の弱識別器を追加した後の分布であるから

$$\mathrm{D}_k(\mathrm{i})=\frac{\frac{1}{N}e^{-y_i F(x_i)}}{Z_{k-1}^{\prime}}$$

で表される。ただし、全サンプルの指数損失の総和Z_{k-1}は期待損失L(F)と一致し、

$$Z_{k-1}^{\prime}=\mathrm{L}(\mathrm{F})=\frac{1}{N}\sum_{i=1}^{N} e^{-y_i F(x_i)}$$

である。ここで、分布D_kにおける s 番目の弱識別器h_s(s = 1,…,M)の誤り率

$$\varepsilon_s=\sum_{i:y_i\neq h_s(x_i)} D_k(i)=\frac{\sum_{\mathrm{i}:y_i\neq h_s(x_i)}^{\mathrm{N}}\frac{1}{N}e^{-y_i F(x_i)}}{Z_{k-1}^{\prime}}$$

より、

$$Z_{k-1}^{'}\varepsilon_s = \frac{1}{N}\sum_{\mathrm{i}:y_i \neq h_s(x_i)}^{\mathrm{N}} e^{-y_i F(x_i)}$$

となるから、式(9)の第 2 項は

$$\frac{e^{\alpha_k} - e^{-\alpha_k}}{N}\sum_{i:y_i \neq h_k(x_i)}^{N} e^{-y_i F(x_i)} = \{e^{\alpha_k} - e^{-\alpha_k}\}Z_{k-1}^{'}\varepsilon_k = \{e^{\alpha_k} - e^{-\alpha_k}\}\mathrm{L(F)}\varepsilon_k$$

と変形できる。分布D_kにおいて損失$\mathrm{L(F}+\alpha_k h_k)$を小さくするためには、図 7 のように$\alpha_k(\geq 0)$のとき$e^{\alpha_k} - e^{-\alpha_k} > 0$となるから

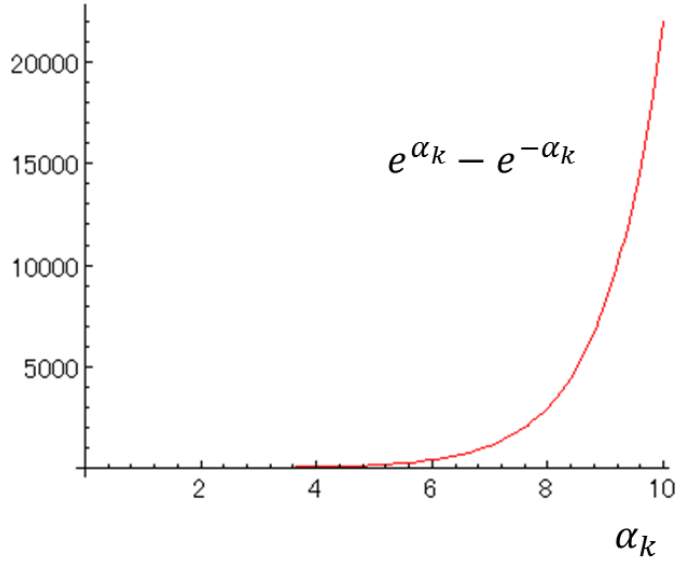


図 7　$e^{\alpha_k} - e^{-\alpha_k}$

最小の誤り率$\varepsilon_{\delta(k)}$となる弱識別器$h_{\delta(k)}$を探せばよいということになる。
同様に式(9)の第 1 項は$Z_{k-1}^{'}$を用いて次式のように書ける。

$$\frac{e^{-\alpha_k}}{N}\sum_{i=1}^{N} e^{-y_i F(x_i)} = Z_{k-1}^{'}e^{-\alpha_k} = \mathrm{L(F)}e^{-\alpha_k}$$

従って損失関数は次のように書き直せる

$$\begin{aligned}\mathrm{L(F}+\alpha_k h_k) &= Z_{k-1}^{'}[e^{-\alpha_k} + \{e^{\alpha_k} - e^{-\alpha_k}\}\varepsilon_s] \\ &= \mathrm{L(F)}[e^{-\alpha_k} + \{e^{\alpha_k} - e^{-\alpha_k}\}\varepsilon_s]\end{aligned}$$

次に、$h_{\delta(k)}$が決まった後で損失関数$\mathrm{L}\left(\mathrm{F}+\alpha_k h_{\delta(k)}\right)$

$$\mathrm{L}\left(\mathrm{F}+\alpha_k h_{\delta(k)}\right) = \mathrm{L(F)}[e^{-\alpha_k} + \{e^{\alpha_k} - e^{-\alpha_k}\}\varepsilon_{\delta(k)}]$$

を最小にするα_kを考える。L(F)、$\varepsilon_{\delta(k)}$は決定済みなので、定数であることに注意して、
α_kで微分して 0 とおく、

$$\frac{d\mathrm{L}\left(\mathrm{F}+\alpha_k h_{\delta(k)}\right)}{d\alpha_k}=\mathrm{L}(\mathrm{F})\left[-e^{-\alpha_k}+\{e^{\alpha_k}+e^{-\alpha_k}\}\varepsilon_{\delta(k)}\right]=0 \qquad (10)$$

また、$\mathrm{L}\left(\mathrm{F}+\alpha_k h_{\delta(k)}\right)$の 2 階微分は

$$\frac{d\mathrm{L}\left(\mathrm{F}+\alpha_k h_{\delta(k)}\right)^2}{d{\alpha_k}^2}=\mathrm{L}(\mathrm{F})\left[e^{-\alpha_k}+\{e^{\alpha_k}-e^{-\alpha_k}\}\varepsilon_{\delta(k)}\right]$$

となる。ここで、α_kがどのような値を取っても$\mathrm{L}(\mathrm{F})>0$、$e^{-\alpha_k}>0$、$\{e^{\alpha_k}-e^{-\alpha_k}\}\varepsilon_{\delta(k)}>0$となり、
２階微分は常に正となるので$\mathrm{L}\left(\mathrm{F}+\alpha_k h_{\delta(k)}\right)$は下に凸な関数となる。
式(10)の両辺$L(F)$で割り、変形すると

$$-(1-\varepsilon_{\delta(k)})+e^{2\alpha_k}=0$$

となり、移項して、両辺対数を取ると

$$\alpha_k=\frac{1}{2}log\left(\frac{1-\boldsymbol{\varepsilon}_{\delta(k)}}{\boldsymbol{\varepsilon}_{\delta(k)}}\right)$$

で最小値を取ることがわかる。これが式(3)となる。式(3)のグラフを下図に示す。

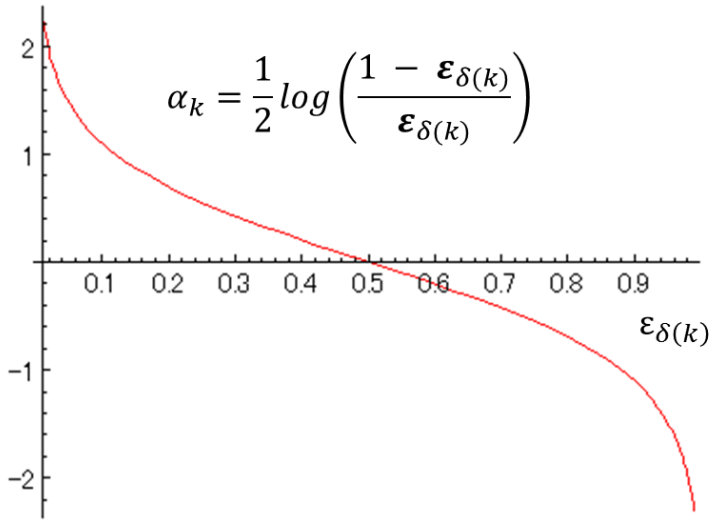


図８　α_k

$\alpha_k(\geq 0)$となるためには誤り率$\boldsymbol{\varepsilon}_{\delta(k)}$は 0～0.5 の値しか取れないことに注意する。
このα_k下での$\mathrm{L}\left(\mathrm{F}+\alpha_k h_{\delta(k)}\right)$は式(6)、式(8)より

$$\mathrm{L}\left(\mathrm{F}+\alpha_k h_{\delta(k)}\right)=\mathrm{L}(\mathrm{F})\left[\left(1-\varepsilon_{\delta(k)}\right)e^{-\alpha_k}+\varepsilon_{\delta(k)}e^{\alpha_k}\right]$$

$$=\mathrm{L}(\mathrm{F})\cdot Z_k$$

$$=\mathrm{L}(\mathrm{F})\cdot 2\sqrt{(1-\varepsilon_{\delta(k)})\varepsilon_{\delta(k)}}$$

となり、Z_kは下図のようなグラフになるので

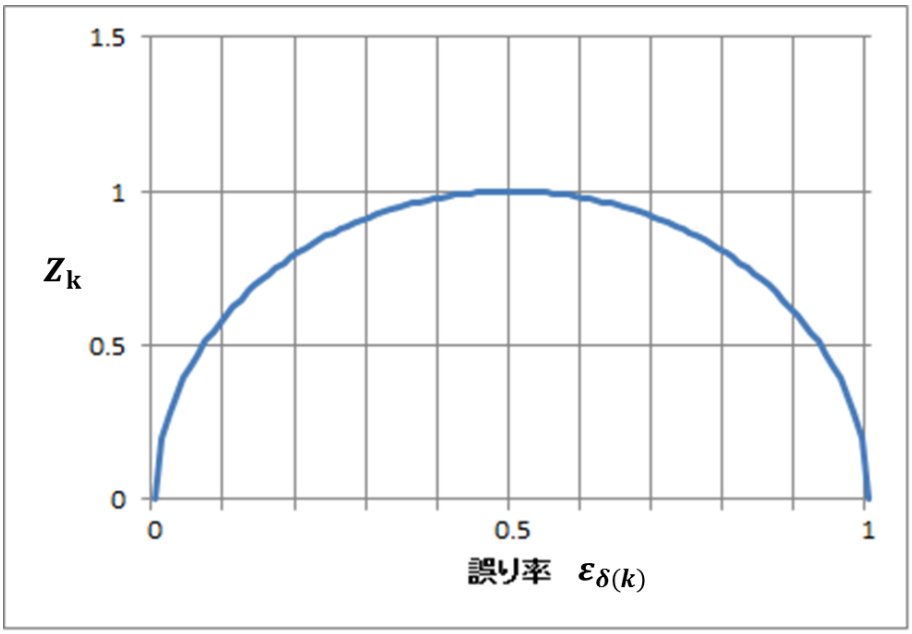


図 9　Z_k

$\varepsilon_{\delta(k)} \leq 0.5$であれば、常に

$$\mathrm{L}\left(\mathrm{F}+\alpha_k h_{\delta(k)}\right)=\mathrm{L}(\mathrm{F})\cdot 2\sqrt{\left(1-\varepsilon_{\delta(k)}\right)\varepsilon_{\delta(k)}}\leq \mathrm{L}(\mathrm{F})$$

($\varepsilon_{\delta(k)}=0.5$ で等号成立)となる。したがって、

$$\mathrm{F}(x)+\frac{1}{2}log\left(\frac{1-\boldsymbol{\varepsilon}_{\delta(k)}}{\boldsymbol{\varepsilon}_{\delta(k)}}\right)h_{\delta(k)}(x)$$

のように誤り率最小の弱識別器$h_{\delta(k)}$を追加することで AdaBoost は指数損失の期待損失を最小化できる。$h_{\delta(k)}$を選択することは期待損失が最も減少する方向を探索していることを意味する。また、信頼度αの決定は、損失関数を最も減少させる方向へ移動させる操作にあたる。

４．特徴点検出

４．１　目、鼻領域の特徴点検出

下図のように検出した目、鼻領域から Harris-Corner 検出[4][5]により、特徴点を算出する。

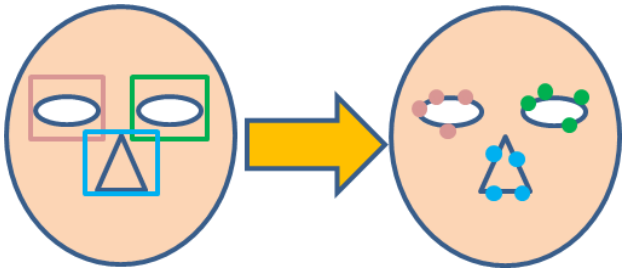


図１０　特徴点の算出

具体的には、式(11)で与えられる Harris 作用素$\mathrm{O}(x,y)$から特徴点を算出する。

$$\mathrm{O}(x,y) = \det\hat{\mathcal{C}}(x,y) - \kappa(\mathrm{tr}\hat{\mathcal{C}}(x,y)) \qquad (11)$$

κは定数であり、[5]によると 0.06～0.08 くらいの値が適当と考えられている。
ここで、$\hat{\mathcal{C}}(x,y)$は式(12)で与えられる。

$$\hat{\mathcal{C}}(x,y) = \begin{bmatrix} \overline{\mathrm{I}_x{}^2(x,y)} & \overline{\mathrm{I}_x(x,y)\mathrm{I}_y(x,y)} \\ \overline{\mathrm{I}_x(x,y)\mathrm{I}_y(x,y)} & \overline{\mathrm{I}_y{}^2(x,y)} \end{bmatrix} \qquad (12)$$

ここで、領域(u,v)における２次元正規化済み正規分布を$\mathrm{G}_\sigma(u,v)$、座標(x,y)における輝度値$\mathrm{I}(x,y)$のx方向の微分を$\mathrm{I}_x(x,y)$、y方向の微分を$\mathrm{I}_y(x,y)$で表すと、x方向の微分の２乗平均$\overline{\mathrm{I}_x{}^2(x,y)}$、$y$方向の微分の２乗平均$\overline{\mathrm{I}_y{}^2(x,y)}$、$x$方向の微分と$y$方向の微分の積の平均$\overline{I_x(x,y)I_y(x,y)}$は次式で与えられる。

$$\overline{\mathrm{I}_x{}^2(x,y)} = \sum_u \sum_v \mathrm{G}_\sigma(u,v)\mathrm{I}_x(x+u,y+v)^2 ,$$

$$\overline{\mathrm{I}_y{}^2(x,y)} = \sum_u \sum_v \mathrm{G}_\sigma(u,v)\,\mathrm{I}_y(x+u,y+v)^2 ,$$

$$\overline{\mathrm{I}_x(x,y)\mathrm{I}_y(x,y)} = \sum_u \sum_v \mathrm{G}_\sigma(u,v)\mathrm{I}_x(x+u,y+v)\,\mathrm{I}_y\,(x+u,y+v)$$

ここで、$\mathrm{I}_x(x,y)$ 、$\mathrm{I}_y(x,y)$は次式で表せる。

$$\mathrm{I}_x(x,y) = \frac{\partial I(x,y)}{\partial x} = I(x+1,y) - I(x-1,y) \ ,$$

$$\mathrm{I}_y(x,y) = \frac{\partial I(x,y)}{\partial y} = I(x,y+1) - I(x,y-1)$$

また、$G_\sigma(u,v)$はガウシアンフィルタとも呼ばれ、領域(u,v)の中心座標から円対称の重み分布を掛ける。これは、領域(u,v)の中央から等方的に重み付けし領域の平均を取る（画像をぼかす）操作となる。次図に 5x5 の場合のガウシアンフィルタを示す。

$\frac{1}{256}$	$\frac{4}{256}$	$\frac{6}{256}$	$\frac{4}{256}$	$\frac{1}{256}$
$\frac{4}{256}$	$\frac{16}{256}$	$\frac{24}{256}$	$\frac{16}{256}$	$\frac{4}{256}$
$\frac{6}{256}$	$\frac{24}{256}$	$\frac{36}{256}$	$\frac{24}{256}$	$\frac{6}{256}$
$\frac{4}{256}$	$\frac{16}{256}$	$\frac{24}{256}$	$\frac{16}{256}$	$\frac{4}{256}$
$\frac{1}{256}$	$\frac{4}{256}$	$\frac{6}{256}$	$\frac{4}{256}$	$\frac{1}{256}$

図１１　5x5 ガウシアンフィルタ

$O(x,y)$は、一様な輝度変化、回転に対して不変量であり、$O(x,y)$の値が大きくなるような座標は下図のような緩やかな傾斜の輝度勾配の極値となる。これは、急激な傾斜の輝度勾配の極値はノイズである可能性があるため、そのような点を避けている。$O(x,y)$の値が大きい座標の順に抽出を行い、特徴点とする。これを目、鼻領域に対し、それぞれ行う。

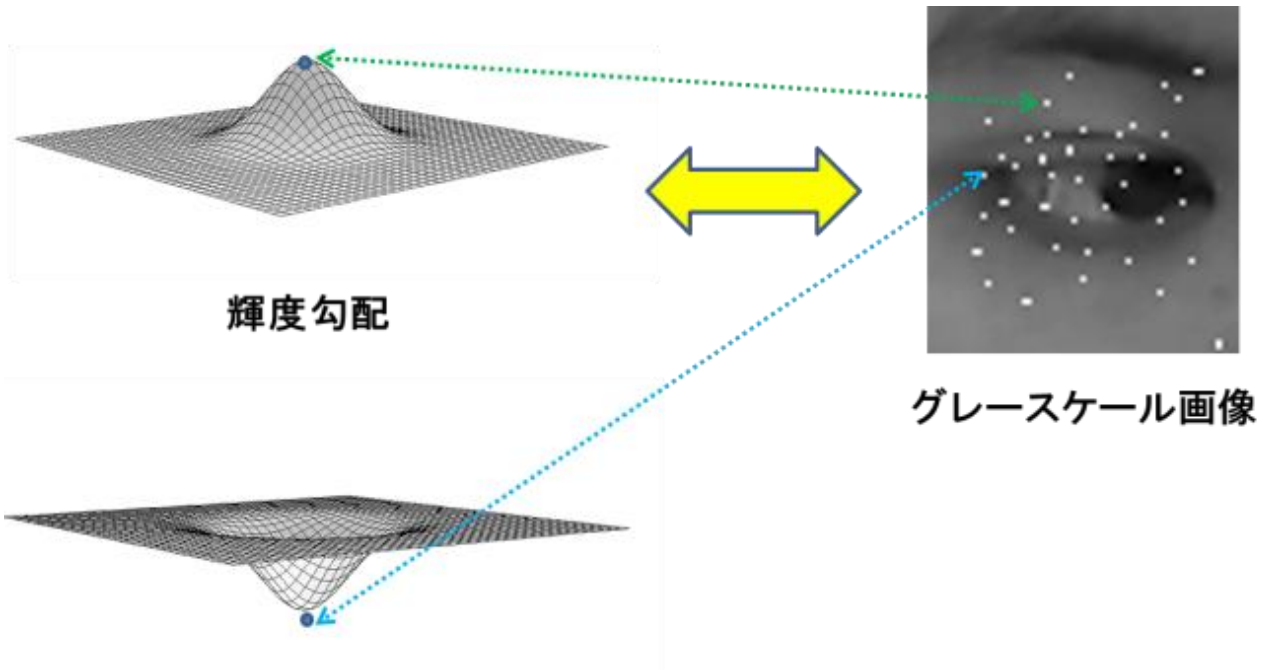


図１２　特徴点の検出

４．２ Harris コーナー検出の性質[5]

（１） 追跡を行うには？

追跡には画像間で点と点の対応を決定する必要がある。対応の取りやすい点は特徴点、関心点と呼ばれる。このような点には対象物の角や線の交わりが選ばれることからコーナーとも呼ばれる。

（２）追跡しやすい特徴点の検出

輝度勾配における平面（グラデーション）やエッジから特徴点を取り出すのは望ましくない。なぜならば下図に示すように、そのような特徴点が左図から右図のように位置変化をした場合、局所的に画像の一部を切り取った場合、対応する候補が複数出現し、移動元との対応関係が一意に取れなくなってしまう。

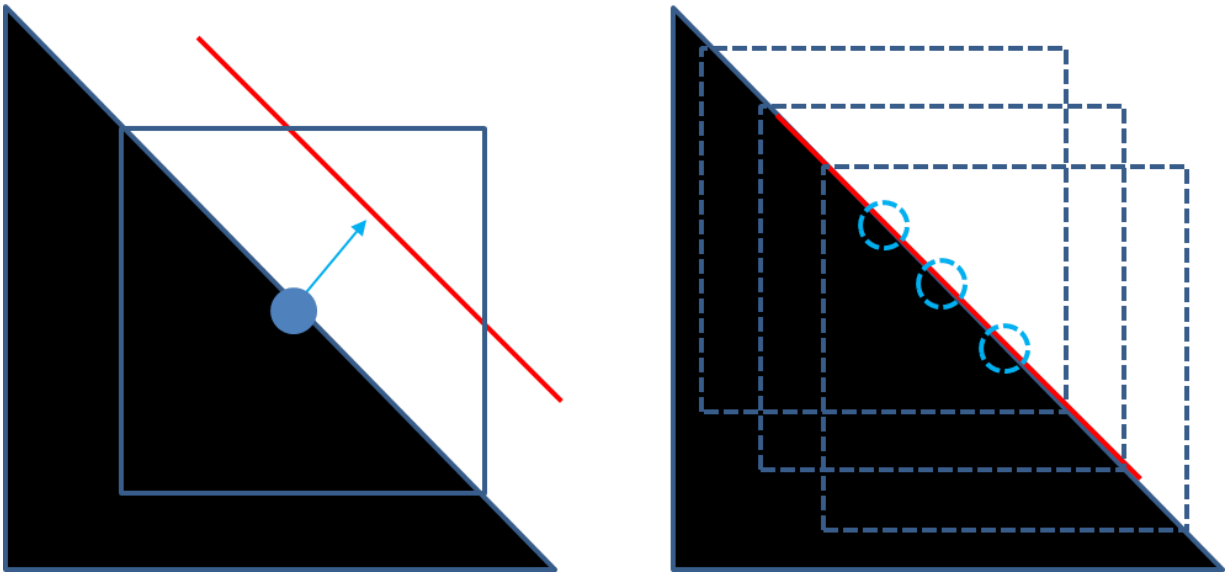


図１３　窓問題

切り取った局所領域を窓と呼ぶことから、このような問題は窓問題と呼ばれる。このことから、精度の良い対応点を求めるには、エッジや平面を避けなければならない。また、追跡対象は画像間で回転する可能性があるので特徴点の抽出は座標回転に関する不変量に基づいて行わなければならない。例えば曲面の形状を記述する量であれば、不変量となり得る。曲面の形状を記述するためには、２次微分以上の評価が必要となる。画像座標(x, y)の輝度値$I(x, y)$を高さと解釈し、x, yを連続変数とみなせば画像は輝度の曲面とみなせる。領域(u, v)を座標(c_x, c_y)動かした領域$(\acute{u}, \acute{v})$を考える。ただし、

$$\acute{u} = u + c_x$$
$$\acute{v} = v + c_y$$

とする。領域$(\acute{u}, \acute{v})$に対し２次元正規化済み正規分布$\mathrm{G}_\sigma(\acute{u} - c_y, \acute{v} - c_y)$で各$I(x, y)$を重みづけし、微小距離$(d_x, d_y)$ずらすことを考える。ここで、２つの窓の差分の重み付き２乗和$\mathrm{W}(d_x, d_y)$は

$$\mathrm{W}(d_x, d_y) = \sum_{\acute{u}} \sum_{\acute{v}} \mathrm{G}_\sigma(\acute{u} - c_x, \acute{v} - c_y)(I(\acute{u} + d_x, \acute{v} + d_y) - I(\acute{u}, \acute{v}))^2 \qquad (13)$$

で与えられる。$I(\acute{u} + d_x, \acute{v} + d_y)$は$(d_x, d_y)$まわりのテイラー展開により次式のように近似できる。

$$I(\acute{u}+d_x,\acute{v}+d_y)=I(\acute{u},\acute{v})+\frac{\partial I(\acute{u},\acute{v})}{\partial x}d_x+\frac{\partial I(\acute{u},\acute{v})}{\partial y}d_y+\cdots$$

$$\approx I(\acute{u},\acute{v})+\mathrm{I}_x d_x+\mathrm{I}_y d_y \qquad (14)$$

２次項以降は十分微小なものとして無視する。
ただし、

$$\mathrm{I}_x(\acute{u},\acute{v})=\frac{\partial I(\acute{u},\acute{v})}{\partial x},\mathrm{I}_y(\acute{u},\acute{v})=\frac{\partial I(\acute{u},\acute{v})}{\partial y}$$

とする。式(13)に式(14)を代入すると

$$\mathrm{W(x,y)}\approx\sum_{\acute{u}}\sum_{\acute{v}}\mathrm{G}_\sigma(\acute{u}-c_x,\acute{v}-c_y)(\mathrm{I}_x d_x+\mathrm{I}_y d_y)^2$$

$$=\overline{\mathrm{I}_x{}^2}d_x{}^2+\overline{\mathrm{I}_x\mathrm{I}_y}d_x d_y+\overline{\mathrm{I}_y{}^2}d_y{}^2 \qquad (15)$$

$$=(d_x \quad d_y)\begin{bmatrix}\overline{\mathrm{I}_x{}^2} & \overline{\mathrm{I}_x\mathrm{I}_y}\\ \overline{\mathrm{I}_x\mathrm{I}_y} & \overline{\mathrm{I}_y{}^2}\end{bmatrix}\begin{pmatrix}d_x\\ d_y\end{pmatrix} \qquad (16)$$

のように変形できる。ここで、式(12)の固有値λを求める固有方程式は

$$\begin{vmatrix}\overline{\mathrm{I}_x{}^2}-\lambda & \overline{\mathrm{I}_x\mathrm{I}_y}\\ \overline{\mathrm{I}_x\mathrm{I}_y} & \overline{\mathrm{I}_y{}^2}-\lambda\end{vmatrix}=\lambda^2-\mathrm{tr}\hat{\mathcal{C}}(d_x,d_y)\lambda+\mathrm{det}\hat{\mathcal{C}}(d_x,d_y)=0 \qquad (17)$$

となり、
一方、2 つの固有値をλ_1 ,λ_2とすると$(\lambda_1<\lambda_2)$ 、解と係数の関係より

$$(\lambda-\lambda_1)(\lambda-\lambda_2)=\lambda^2-(\lambda_1+\lambda_2)\lambda+\lambda_1\lambda_2=0 \qquad (18)$$

とも書けるので、式(17)(18)の係数比較より式(19)が導ける。

$$\begin{aligned}\mathrm{tr}\hat{\mathcal{C}}(d_x,d_y)&=\lambda_1+\lambda_2=\overline{\mathrm{I}_x{}^2}+\overline{\mathrm{I}_y{}^2}\ ,\\ \mathrm{det}\hat{\mathcal{C}}(d_x,d_y)&=\lambda_1\lambda_2=\overline{\mathrm{I}_x{}^2}\,\overline{\mathrm{I}_y{}^2}-(\overline{\mathrm{I}_x\mathrm{I}_y})^2\end{aligned} \qquad (19)$$

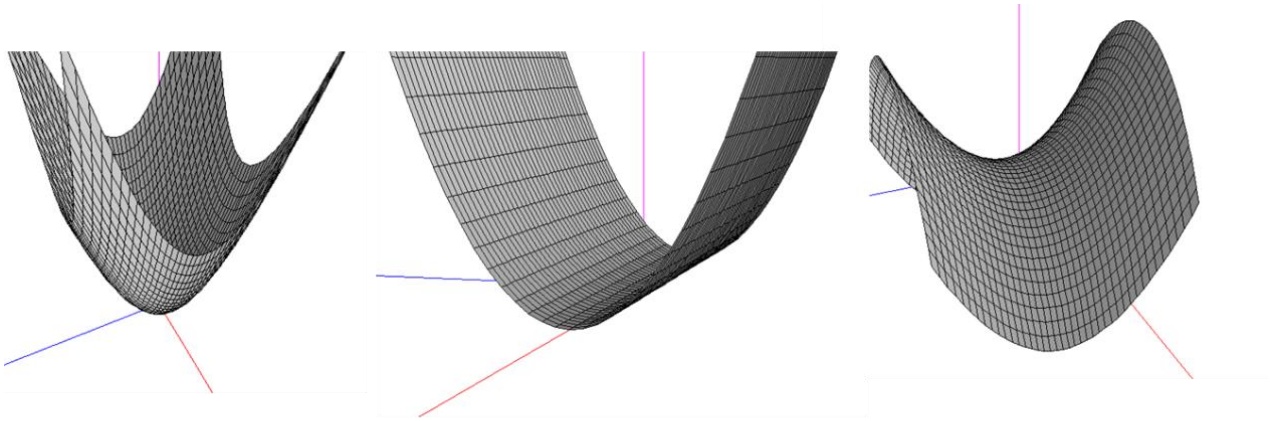
$\mathrm{tr}\hat{\mathcal{C}}(d_x,d_y)$は重みづけされているが輝度勾配の大きさを示している。一方、$\mathrm{det}\hat{\mathcal{C}}(d_x,d_y)$は$\mathrm{W}(d_x,d_y)$の勾配の形状を示している。2 変数関数$f(x,y)=ax^2+bxy+cy^2$の判別式$\Delta$とすると、

$$\Delta=4\mathrm{ac}-\mathrm{b}^2$$

特にbが 2 の倍数のとき$\acute{\mathrm{b}}=2\mathrm{b}$として判別式$\Delta$は判別式$\acute{\Delta}$のように書き直せる

$$\acute{\Delta}=\mathrm{ac}-\acute{\mathrm{b}}^2$$

曲面の形状は次図のようにおおまかに３種類に分類できる。



$(i) f(x,y) = x^2 + y^2$　　$(ii) f(x,y) = x^2$　　$(iii) f(x,y) = x^2 - y^2$

図１４　曲面の形状

式(15)の係数から $\mathrm{W}(d_x, d_y)$ の判別式 $\acute{\Delta}$ は $\det\hat{\mathcal{C}}(d_x, d_y)$ に一致する。$\mathrm{W}(d_x, d_y)$ の輝度勾配は以下のように議論ができる。ただし、$\mathrm{W}(d_x, d_y)$ が輝度勾配同士の差を取っていることに注意する。

$(i)\ \det\hat{\mathcal{C}}(d_x, d_y) = \lambda_1\lambda_2 = \overline{\mathrm{I}_x{}^2}\,\overline{\mathrm{I}_y{}^2} - (\overline{\mathrm{I}_x\mathrm{I}_y})^2 \gg 0 \Rightarrow$ コーナー

$(ii)\ \det\hat{\mathcal{C}}(d_x, d_y) = \lambda_1\lambda_2 = \overline{\mathrm{I}_x{}^2}\,\overline{\mathrm{I}_y{}^2} - (\overline{\mathrm{I}_x\mathrm{I}_y})^2 \approx 0 \Rightarrow$ エッジ、もしくは平面

なお、

$(iii)\ \det\hat{\mathcal{C}}(d_x, d_y) < 0 \Rightarrow$ ありえない

$\mathrm{W}(x,y) \geq 0$ より式(16)も非負となり、このとき $\hat{\mathcal{C}}(x,y)$ は半正定値行列なので $\hat{\mathcal{C}}(x,y)$ の固有値は非負となる。そのため、固有値の積は負となることはないので(iii)はあり得ないことがわかる。(ii)となる場合、輝度勾配の大きさ $\mathrm{tr}\hat{\mathcal{C}}(d_x, d_y)$ の情報を用いるとさらに次のように議論ができる。

$(a)\ \det\hat{\mathcal{C}}(d_x, d_y) = \lambda_1\lambda_2 = \overline{\mathrm{I}_x{}^2}\,\overline{\mathrm{I}_y{}^2} - \left(\overline{\mathrm{I}_x\mathrm{I}_y}\right)^2 \approx 0$ かつ $\mathrm{tr}\hat{\mathcal{C}}(d_x, d_y) = \lambda_1 + \lambda_2 \gg 0 \Rightarrow$ エッジ

$(b)\ \det\hat{\mathcal{C}}(d_x, d_y) = \lambda_1\lambda_2 = \overline{\mathrm{I}_x{}^2}\,\overline{\mathrm{I}_y{}^2} - \left(\overline{\mathrm{I}_x\mathrm{I}_y}\right)^2 \approx 0$ かつ $\mathrm{tr}\hat{\mathcal{C}}(d_x, d_y) = \lambda_1 + \lambda_2 \approx 0 \Rightarrow$ 平面

また、判別式の値によって窓の中心座標 (c_x, c_y) における輝度勾配の形状も推測できる。(i) の場合、(d_x, d_y) をどのような方向にずらしても $\mathrm{W}(d_x, d_y)$ の値が増えることから、窓の中心座標 (c_x, c_y) を極値とした輝度勾配のコーナーであると推測できる。(ii) の場合、特定方向に対しては動かしても $\mathrm{W}(d_x, d_y)$ が増えないことからその方向に伸びるエッジである推測できる。このことから、式(19)から固有値の和と積の関係を見るだけで輝度勾配の形状がわかることになる。以上から固有値と輝度勾配の形状の関係をまとめると次図のようになる。

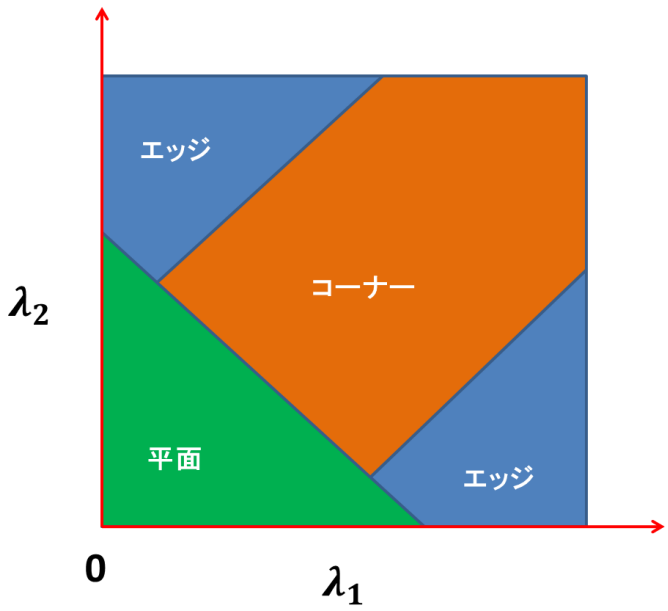


図１５　固有値と輝度勾配の形状の関係

λ_1とλ_2の値が互いに大きい場合はコーナーとなり、片方の固有値がもう片方の固有値より十分大きい場合はエッジとなる。λ_1とλ_2がともに小さい場合は平面となる。それを踏まえて式(11)は輝度勾配の極値となる点でかつ緩やかな勾配となるような点が最も値の高くなる点となることがわかる。勾配の大きさで減算しているのは、急勾配となりえるノイズを防ぐためである。ところで、曲面の形状を記述するためには、２次微分以上の評価が必要となると述べたが、$\hat{C}(d_x, d_y)$は一次微分までしか行っていない。詳細を省くが、実はガウシアンフィルタ$G_\sigma(u, v)$が２次微分以上の役割を果たしているため（[5]によると３次微分まで評価される）、$\det\hat{C}(d_x, d_y)$により曲面の形状を評価することができる。

５．特徴点追跡[6][7]

特徴点の追跡には、Lucas-Kanade 法[6][7]（以下 LK 法）を用いる。時刻$\boldsymbol{t}$における特徴点を中心とした領域ω内の座標$\boldsymbol{u}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{t})$ の輝度値を$\boldsymbol{I}(\boldsymbol{u})$とする。図１６のように時刻$\boldsymbol{t+1}$に座標$\boldsymbol{\acute{u}}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{t+1})$から移動ベクトル$\boldsymbol{d}(\boldsymbol{\delta x}, \boldsymbol{\delta y}, \boldsymbol{0})$ずれたときの輝度値を$\boldsymbol{I}(\boldsymbol{\acute{u}+d})$とすると、輝度値の差の二乗和 **SSD** は式(20)で表せる。

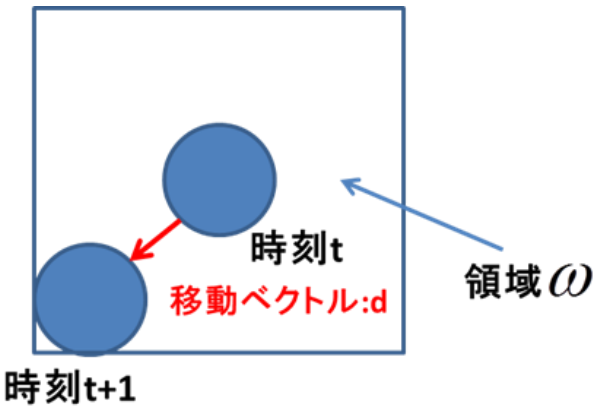


図１６　移動ベクトル$\boldsymbol{d}$

$$\mathbf{SSD}: \boldsymbol{\varepsilon}(\boldsymbol{d}) = \sum_{u\in\omega}(\boldsymbol{I}(\acute{\boldsymbol{u}}+\boldsymbol{d}) - \boldsymbol{I}(\boldsymbol{u}))^2 \qquad (20)$$

次式のように、この SSD が最も小さくなる、移動ベクトル$\boldsymbol{d}$を時刻$\boldsymbol{t}$から時刻$\boldsymbol{t}+\boldsymbol{\delta t}$の間に動いた移動ベクトル$\widehat{\boldsymbol{d}}$と推定する。

$$\widehat{\boldsymbol{d}} = \mathbf{arg\,min}\ \boldsymbol{\varepsilon}(\boldsymbol{d})$$

LK 法は、この$\widehat{\boldsymbol{d}}$を解析的に求める方法であり、処理時間は常に一定となる。式(20)の第 1 項をテイラー展開し、次のように近似する。

$$\begin{aligned}\boldsymbol{I}(\acute{\boldsymbol{u}}+\boldsymbol{d}) &= \boldsymbol{I}(\boldsymbol{x}+\boldsymbol{\delta x}, \boldsymbol{y}+\boldsymbol{\delta y}, \boldsymbol{t}+\mathbf{1}) \\ &= \boldsymbol{I}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{t}) + \boldsymbol{\delta x}\boldsymbol{I_x} + \boldsymbol{\delta y}\boldsymbol{I_y} + \boldsymbol{I_t} + \cdots \\ &\approx \boldsymbol{I}(\boldsymbol{x},\boldsymbol{y},\boldsymbol{t}) + \boldsymbol{\delta x}\boldsymbol{I_x} + \boldsymbol{\delta y}\boldsymbol{I_y} + \boldsymbol{I_t} \qquad (21)\end{aligned}$$

２次項以降は十分微小なものとして無視する。
ただし、

$$\mathrm{I}_x(x,y,t) = \frac{\partial I(x,y,t)}{\partial x}, \mathrm{I}_y(x,y,t) = \frac{\partial I(x,y,t)}{\partial y}, \mathrm{I}_t(x,y,t) = \frac{\partial I(x,y,t)}{\partial t}$$

とする。
$\boldsymbol{\delta x} = \boldsymbol{d_x}, \boldsymbol{\delta y} = \boldsymbol{d_y}$として式(21)を式(20)に代入すると

$$\boldsymbol{\varepsilon}(\mathbf{d}) = \sum_{u\in\omega}(\boldsymbol{I}(\acute{\boldsymbol{u}}+\boldsymbol{d}) - \boldsymbol{I}(\boldsymbol{u}))^2 = \sum_{(x,y)\in\omega}(\boldsymbol{d_x}\mathrm{I}_x + \boldsymbol{d_y}\mathrm{I}_y + \mathrm{I}_t)^2$$

となり、$\widehat{\boldsymbol{d}}$をもとめるには、$\frac{\partial\boldsymbol{\varepsilon}}{\partial \boldsymbol{d_x}} = \mathbf{0}, \frac{\partial\boldsymbol{\varepsilon}}{\partial \boldsymbol{d_y}} = \mathbf{0}$となる$\boldsymbol{d}$を求めればよいので

$$\frac{\partial\boldsymbol{\varepsilon}}{\partial \boldsymbol{d_x}} = \mathbf{2}\sum_{(x,y)\in\boldsymbol{\omega}}\mathrm{I}_x(\boldsymbol{d_x}\mathrm{I}_x + \boldsymbol{d_y}\mathrm{I}_y + \mathrm{I}_t) = \mathbf{0}$$
$$\frac{\partial\boldsymbol{\varepsilon}}{\partial \boldsymbol{d_y}} = \mathbf{2}\sum_{(x,y)\in\boldsymbol{\omega}}\mathrm{I}_y(\boldsymbol{d_x}\mathrm{I}_x + \boldsymbol{d_y}\mathrm{I}_y + \mathrm{I}_t) = \mathbf{0}$$

より

$$\begin{pmatrix}\frac{\partial\boldsymbol{\varepsilon}}{\partial \boldsymbol{d_x}} \\ \frac{\partial\boldsymbol{\varepsilon}}{\partial \boldsymbol{d_y}}\end{pmatrix} = \begin{pmatrix}\sum_{(x,y)\in\boldsymbol{\omega}}(\boldsymbol{d_x}{\mathrm{I}_x}^2 + \boldsymbol{d_y}\mathrm{I}_x\mathrm{I}_y + \mathrm{I}_x\mathrm{I}_t) \\ \sum_{(x,y)\in\boldsymbol{\omega}}(\boldsymbol{d_x}\mathrm{I}_x\mathrm{I}_y + \boldsymbol{d_y}{\mathrm{I}_y}^2 + \mathrm{I}_y\mathrm{I}_t)\end{pmatrix} = \mathbf{0} \qquad (\mathbf{22})$$

ここで、領域ω内の画像座標$\boldsymbol{u_1}$~$\boldsymbol{u_n}$とし、行列$\boldsymbol{A}$、ベクトル$\boldsymbol{b}$を定義する。
$\boldsymbol{T}$は行列、ベクトルの転置とする

$$\mathbf{A} = \begin{bmatrix} \boldsymbol{I_x(u_1,t)} & \cdots & \boldsymbol{I_x(u_n,t)} \\ \boldsymbol{I_y(u_1,t)} & \dots & \boldsymbol{I_y(u_n,t)} \end{bmatrix}^{\boldsymbol{T}}, \qquad (23)$$

$$\boldsymbol{b} = [\boldsymbol{-I_t(u_1,t)} \quad \dots \quad \boldsymbol{-I_t(u_n,t)}]^{\boldsymbol{T}}$$

式(23)を用いて式(22)を変形する。ここで、

$$(\boldsymbol{A^TA})\boldsymbol{d} = \begin{bmatrix} \boldsymbol{I_x(u_1,t)} & \cdots & \boldsymbol{I_x(u_n,t)} \\ \boldsymbol{I_y(u_1,t)} & \dots & \boldsymbol{I_y(u_n,t)} \end{bmatrix} \begin{bmatrix} \boldsymbol{I_x(u_1,t)} & \boldsymbol{I_y(u_1,t)} \\ \vdots & \vdots \\ \boldsymbol{I_x(u_n,t)} & \boldsymbol{I_y(u_n,t)} \end{bmatrix} \begin{bmatrix} \boldsymbol{d_x} \\ \boldsymbol{d_y} \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{(\boldsymbol{x},\boldsymbol{y})\in\boldsymbol{\omega}} \mathrm{I_x}^2 & \sum_{(\boldsymbol{x},\boldsymbol{y})\in\boldsymbol{\omega}} \mathrm{I_x I_y} \\ \sum_{(\boldsymbol{x},\boldsymbol{y})\in\boldsymbol{\omega}} \mathrm{I_x I_y} & \sum_{(\boldsymbol{x},\boldsymbol{y})\in\boldsymbol{\omega}} \mathrm{I_y}^2 \end{bmatrix} \begin{bmatrix} \boldsymbol{d_x} \\ \boldsymbol{d_y} \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{(\boldsymbol{x},\boldsymbol{y})\in\boldsymbol{\omega}} (\mathrm{d_x I_x}^2 + \mathrm{d_y I_x I_y}) \\ \sum_{(\boldsymbol{x},\boldsymbol{y})\in\boldsymbol{\omega}} (\mathrm{d_x I_x I_y} + \mathrm{d_y I_y}^2) \end{bmatrix}$$

$$\boldsymbol{A^Tb} = \begin{bmatrix} \boldsymbol{I_x(u_1,t)} & \cdots & \boldsymbol{I_x(u_n,t)} \\ \boldsymbol{I_y(u_1,t)} & \dots & \boldsymbol{I_y(u_n,t)} \end{bmatrix} \begin{bmatrix} \boldsymbol{-I_t(u_1,t)} \\ \vdots \\ \boldsymbol{-I_t(u_n,t)} \end{bmatrix}$$

$$= \begin{bmatrix} -\sum_{(\boldsymbol{x},\boldsymbol{y})\in\boldsymbol{\omega}} \mathrm{I_x I_t} \\ -\sum_{(\boldsymbol{x},\boldsymbol{y})\in\boldsymbol{\omega}} \mathrm{I_y I_t} \end{bmatrix}$$

となるので式(22)は

$$\begin{pmatrix} \boldsymbol{\frac{\partial \varepsilon}{\partial d_x}} \\ \boldsymbol{\frac{\partial \varepsilon}{\partial d_y}} \end{pmatrix} = \begin{pmatrix} \sum_{(\boldsymbol{x},\boldsymbol{y})\in\boldsymbol{\omega}} (\boldsymbol{d_x}\mathrm{I}_x{}^2 + \boldsymbol{d_y}\mathrm{I}_x\mathrm{I}_y + \mathrm{I}_x\mathrm{I}_t) \\ \sum_{(\boldsymbol{x},\boldsymbol{y})\in\boldsymbol{\omega}} (\boldsymbol{d_x}\mathrm{I}_x\mathrm{I}_y + \boldsymbol{d_y}\mathrm{I}_y{}^2 + \mathrm{I}_y\mathrm{I}_t) \end{pmatrix} = \left(\boldsymbol{A^TA}\right)\boldsymbol{d} - \boldsymbol{A^Tb} = \mathbf{0}$$

と変形できる。ここで、
行列$\boldsymbol{A^TA}$が正則な時、逆行列$(\boldsymbol{A^TA})^{\boldsymbol{-1}}$を持ち、$\boldsymbol{\hat{d}}$が求まる。

$$\boldsymbol{\hat{d}} = (\boldsymbol{A^TA})^{\boldsymbol{-1}}\boldsymbol{A^Tb}$$

図１３のように$\boldsymbol{I_x}$は$\boldsymbol{x}$方向の微分であり、横方向のエッジ画像から求められる。$\boldsymbol{I_y}$は$\boldsymbol{y}$方向の微分であり、縦方向のエッジ画像から求められる。$\boldsymbol{I_t}$は時間微分であり、時間差分画像から求められる。

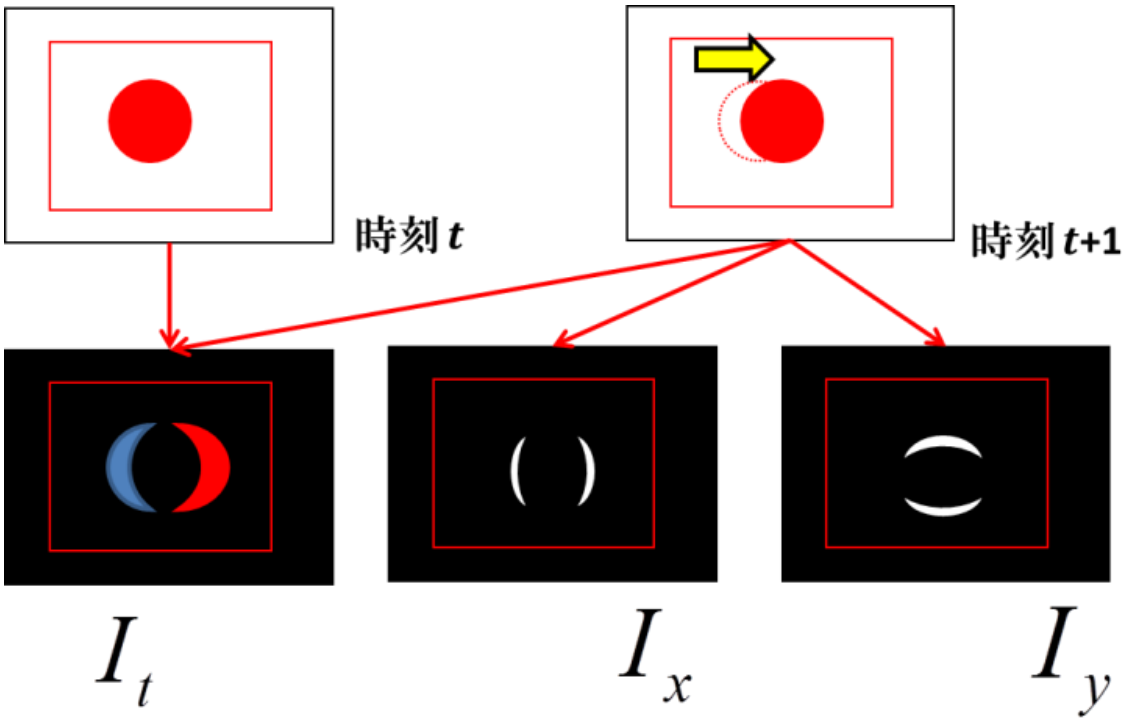


図１７　移動ベクトル算出に必要な差分画像

全ての特徴点に対し、移動ベクトル$\hat{\mathbf{d}}$を算出し、特徴点の追跡を行う。

６．３軸の設定

下図のように画像の横軸を X 軸、縦軸を Y 軸とし、X 軸と Y 軸に直交する奥行き方向を Z 軸とする正規直交座標系を定義する。

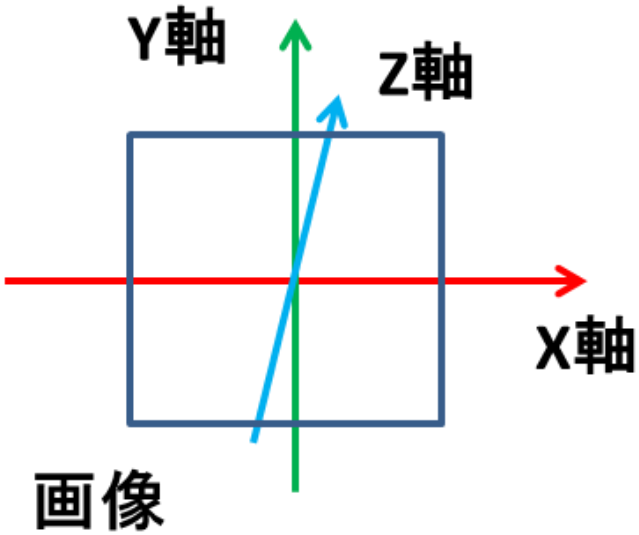


図１８　３軸の設定

７．平行移動情報の取得

7.1 X 軸、Y 軸平行移動情報の取得

まず、右目領域内の各特徴点の平均移動ベクトルを算出する。同様に左目、鼻領域に対し、平均移動ベクトルを算出する（図１９）。最後にこれら３つの平均移動ベクトルの平均を算出し、これを X 軸と Y 軸の平行移動情報とする。

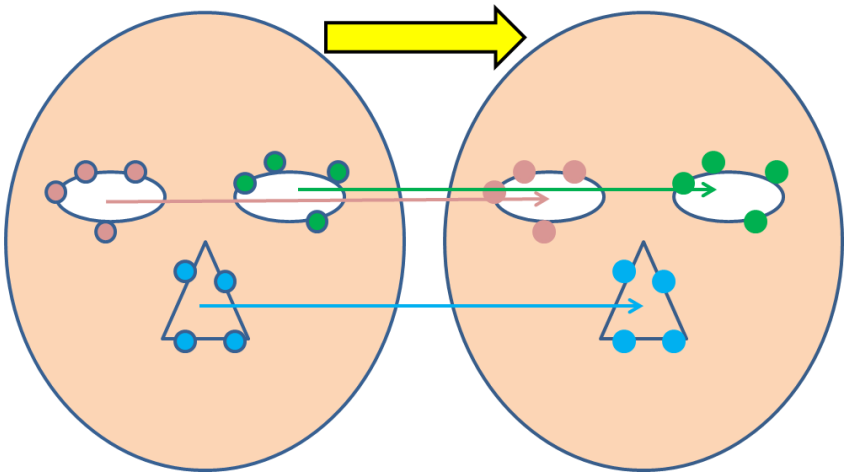


図１９　X 軸、Y 軸平行移動の取得

7.2 Z 軸平行移動情報の取得

まず、右目領域内の各特徴点の平均位置を算出する。次に、平均位置から右目領域内の各特徴点に対し、平均位置からの距離を測り、右目領域内平均距離を算出する。同様に、左目、鼻領域に対してもそれぞれ領域内平均距離を算出する。最後にこれら３つの領域内平均距離の平均距離を取り、この平均距離の変化を Z 軸平行移動情報とする。例えば、図２０の左図から右図のように変化した場合、各領域の中心点から特徴点への距離が長くなるため、奥行き方向に対し手前に動いたと判別できる。逆に距離が短くなった場合は奥に動いたと判別できる。

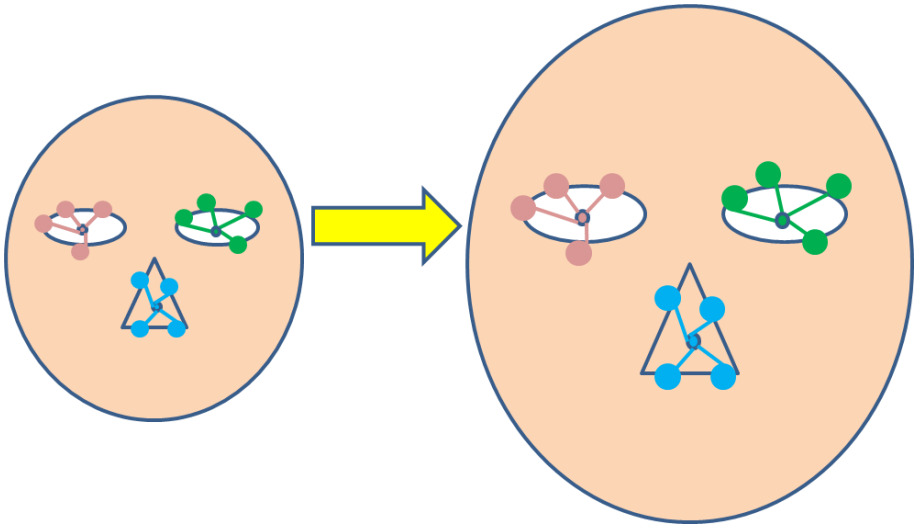


図２０　Z 軸平行移動の取得

８．回転情報の取得

8.1 X 軸、Y 軸回転情報の取得

回転方向と回転量から回転情報を取得する。Y 軸の回転情報の取得には、まず、下図のように右目、左目領域の特徴点の平均点同士を結ぶ線分を作成する。次に、線分の長さ変化で回転量を算出する。最後に、X 軸に関する平行移動情報から回転方向を算出する。

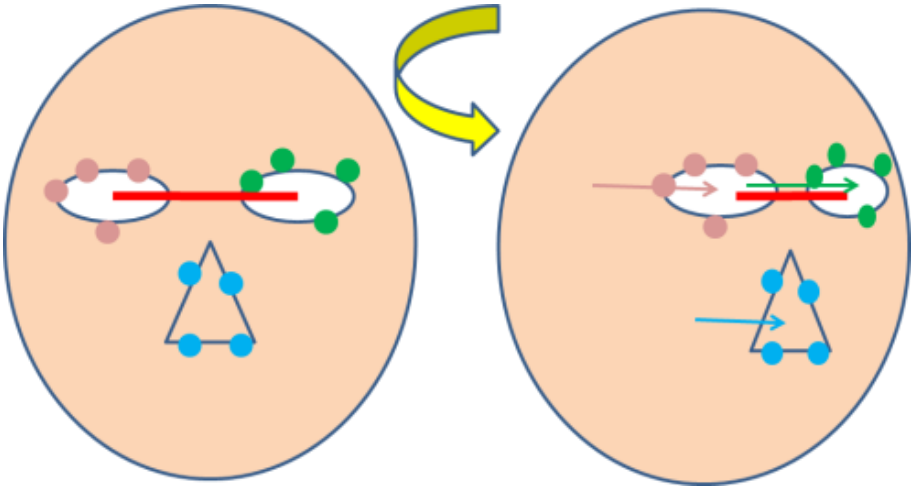


図２１　Y 軸回転情報の取得

同様に X 軸の回転情報の取得には、下図のように目領域間の中点と鼻領域の特徴点の平均点を結ぶ線分を作成し、その線分の長さ変化で回転量を算出し、Y 軸に関する平行移動情報から回転方向を算出する。

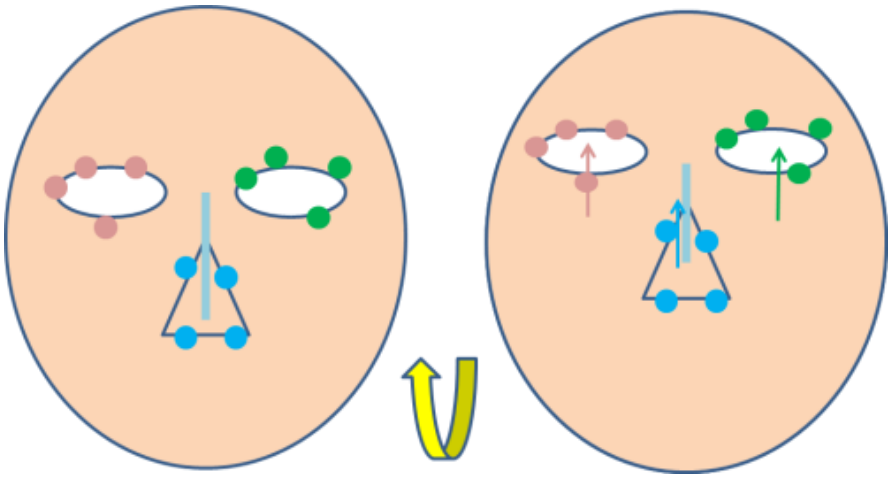


図２２　X 軸回転情報の取得

8.2 Z 軸回転情報の取得

下図のように X 軸に対する両目を結ぶ線の傾きθを算出する。傾きθを Z 軸の回転情報とする。

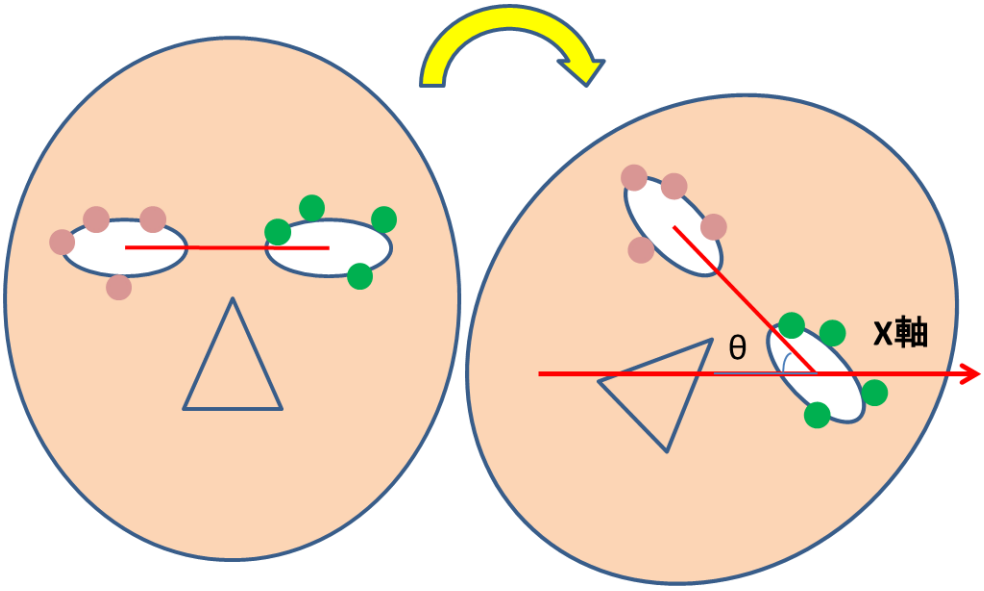


図２３　Z 軸回転情報の取得

９．ユーザインタフェースの実装

下図のように、ウェブカメラからの奥行方向を Z 軸、取得した動画像の横軸を X 軸、縦軸を Y 軸とした。

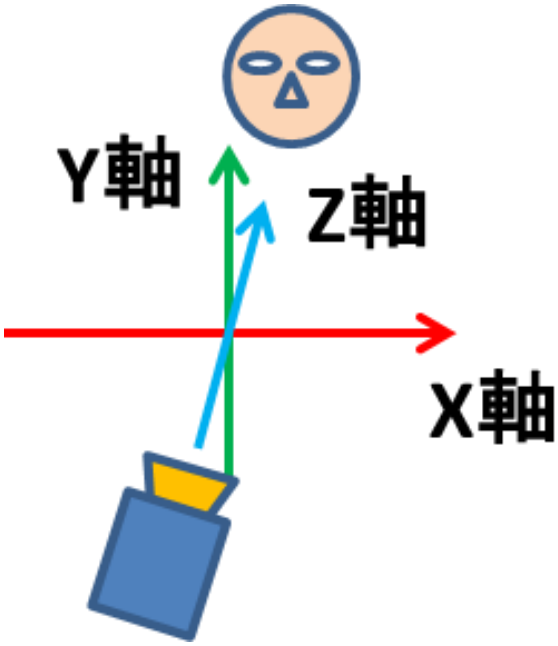


図２４　装置

ウェブカメラで取得した動画像を入力情報としてユーザインタフェースを実装した。
WindowsXP 以降で動作する。本システムのフローチャートは次図となる

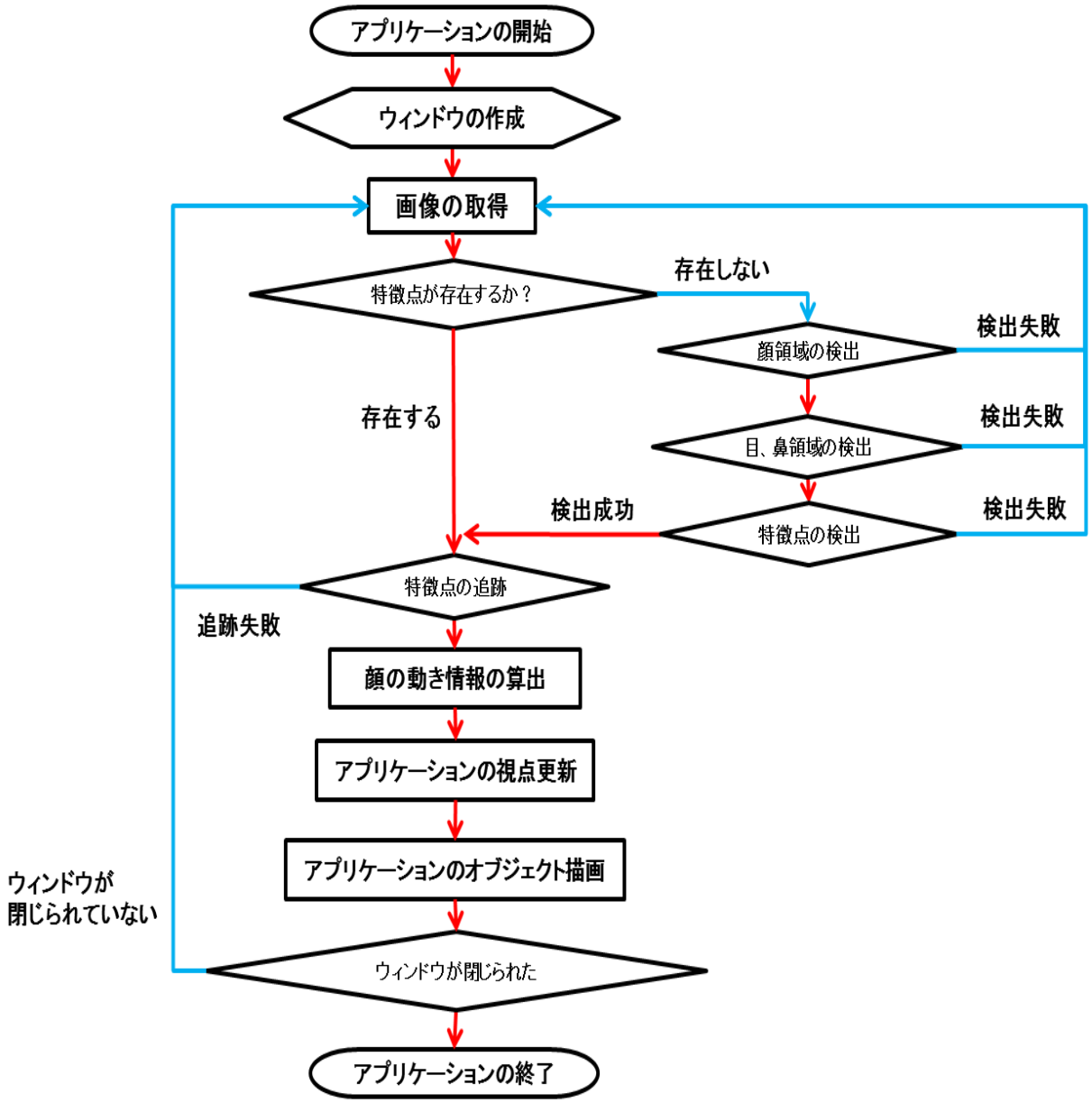
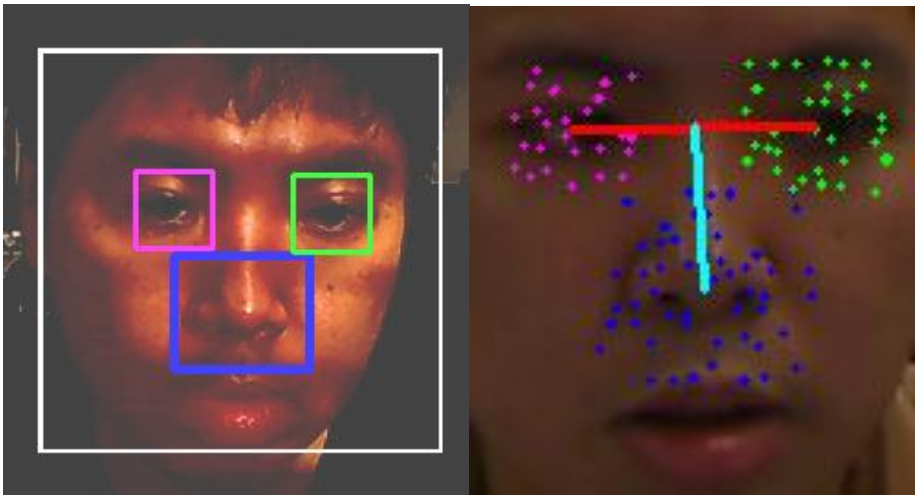


図２５　フローチャート

まず、操作対象となるアプリケーションのウィンドウを作成する。続いて画像の情報を取得する。実験ではウェブカメラから取得しているが、動画ファイルにも適応することができる。画像内に特徴点が存在していなければ、顔領域の検出を行う。顔領域の検出に成功すれば、その顔から目、鼻領域の検出を行う。そして、目、鼻の各領域からそれぞれ特徴点を検出する。いずれも、検出に失敗した場合は、特徴点の検出ができるまで画像の取得を繰り返す。特徴点の検出に成功した場合、特徴点を追跡し、移動ベクトルを算出する。その移動ベクトルに基づいて顔の動き情報の算出を行い、アプリケーションの視点を更新する。最後にアプリケーションのオブジェクトを描画する。これにより顔の動きに連動してアプリケーションの風景が動く。領域の検出と特徴点の検出は次図のように実装できた。



領域の検出　　**特徴点の検出**

図２６　ユーザインタフェースの実装

ユーザインタフェースから取得した情報を図２７のような３Ｄビューワーの視点操作に利用した。カメラに顔を近づけば、３Ｄビューワーの奥へ進むことができ、逆にカメラから顔を遠ざければ後退することができる。また、首を傾げれば、３Ｄビューワーの画面も回転する。顔を上下左右にスライドさせれば、画面もスライドする。顔を横に向ければ、画面も連動して左右を見ることができる。同様に、顔を上下に向けば画面も連動して上下を見ることができる。以上より顔の動きによる３軸の平行移動・回転情報を取得し、視点操作することができた。

図２７　３Ｄビューワー

１０．考察

１）特徴点のロスト

急激な動きを伴う場合、下図のように追跡対象となる特徴点を中心とした領域ωの外に特徴点が出てしまうため特徴点の追跡が不可能となる。

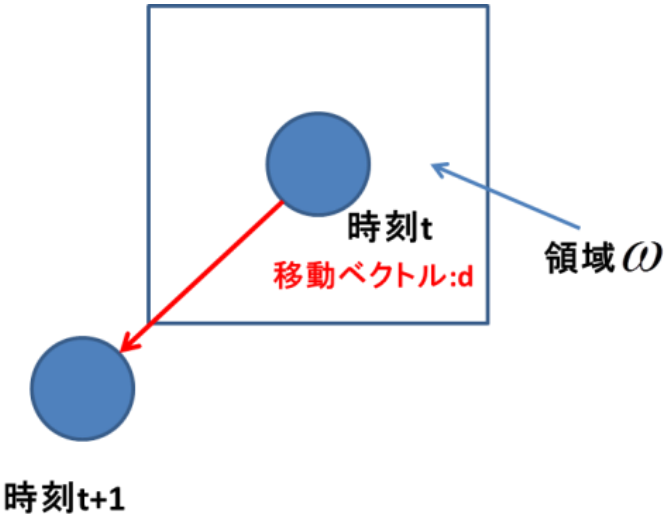


図２８　追跡不可能な場合

このため、図２９の左図のように平均位置が目、鼻領域の中心から外れてしまい、目の間を結ぶ線分や鼻と目の間の中点を結ぶ線分が崩れてしまう。また、図２９の右図のように顔を横向きにするなど目、鼻領域が遮蔽された場合、画像から特徴点の追跡は不可能となる。

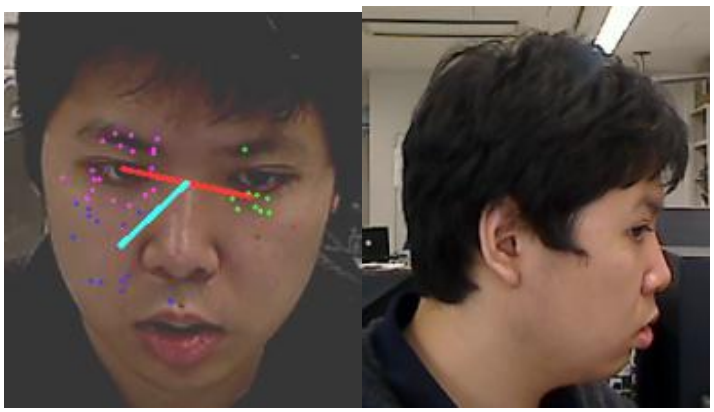


図２９　特徴点のロスト

このような場合、時刻$\boldsymbol{t}$と時刻$\boldsymbol{t+1}$の間で特徴点が減るため、時刻$\boldsymbol{t}$の特徴点の数に対し、時刻$\boldsymbol{t+1}$の特徴点の数が一定数以上減ったときに再び STEP１の顔領域から検出することで対処できる。

２）照明に関する影響

輝度勾配を使った領域検出、特徴点の抽出と追跡を行っているため、照明変化の影響を受ける。特に画像が図３０のように白飛びしている場合は輝度勾配の情報が失われるため、顔、目、鼻領域の検出は不可能となり、特徴点の検出および追跡も不可能となる。この現象は、アプリケーション実行時の環境と画像取得のデバイスに依存するため、本研究ではウェブカメラ付属のドライバソフトで露出や明度の調整を行い対処した。

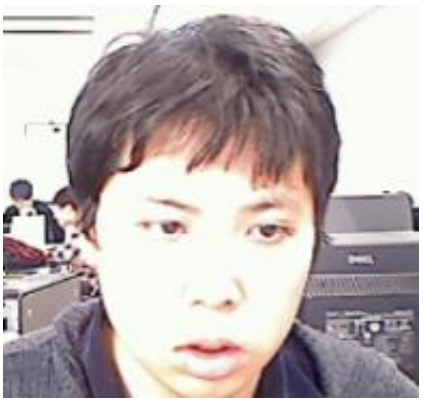


図３０　白飛び

３）目領域の誤検出
図３１のように、顔領域に対し、目領域の検出を行うと目と違う領域を誤検出してしまうことがある。この問題は、この後の特徴点検出などの処理にも影響してしまう。対処に関しては次の章で述べる。

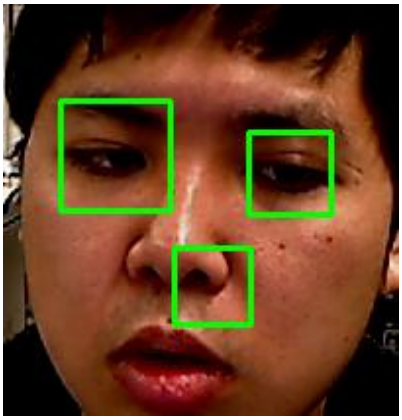


図３１　目の誤検出例

１１．ユーザインタフェースの改良

目領域の誤検出を防ぐために、下図のように両目領域を検出し、その中から左目、右目領域の検出を行うことで対応した。両目領域の検出には両目領域用に AdaBoost で学習済み強識別器を用いている。

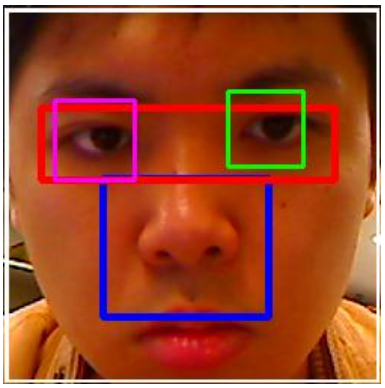


図３２　両目領域の検出

さらに、顔の動きを取得する処理を並列化することにより、処理速度を大幅に向上させることができた。改良後のフローチャートを次図に示す。

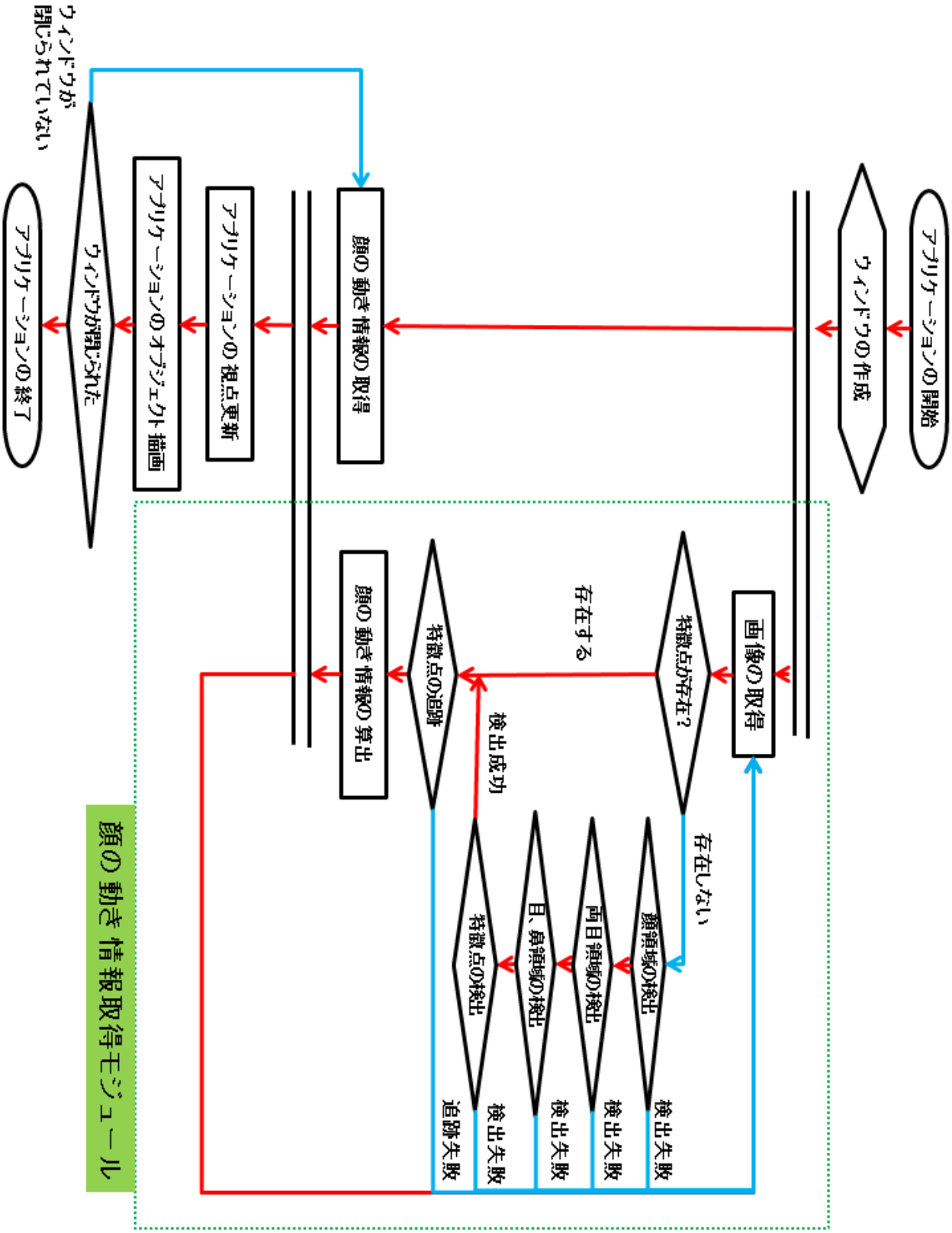


図３３　改良後のフローチャート

２重線は並列処理を意味している。ウィンドウの作成後、顔の動きを取得する処理を並列化し、顔が動いた時の情報取得のみをアプリケーション側本体の処理と同期を取ることで、処理の高速化と本システムを完全に独立なモジュールとして分離することに成功した。

１２．終わりに

本研究では、特徴点追跡に LK 法を用いたが、それを改良した KLT 法[8][9]を用いることでより追跡精度の高いユーザインタフェースに改良できる。また、本研究では、両目と鼻の領域を使ったが、鼻孔の大きさを利用した研究などもある[10]。

１３．参考文献

[１]Paul Viola and Michael Jones: Rapid Object Detection using a Boosted Cascade of Simple Features. 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), pp.511-518, 2001

[２]Rainer Lienhart and Jochen Maydt: An Extended Set of Haar-like Features for Rapid Object Detection. IEEE The International Conference on Image Processing (ICIP 2002), Vol. 1, pp. 900-903, 2002

[３]八木康史、斎藤英雄編：コンピュータビジョン最先端ガイド１、アドコム・メディア株式会社、2008.

[４]Chris Harris and Mike Stephens: A combined corner and edge detector.Proceedings of the 4th Alvey Vision Conference (pp.147-151),1988.

[５]金澤靖、金谷健一：コンピュータビジョンのための画像の特徴点の抽出.電子情報通信学会誌,Vol.87, No.12,pp.1043-1048,2004.

[６]Bruce D. Lucas, Takeo Kanade: An Iterative Image Registration Technique with an Application to Stereo Vision.Proceedings of the 1981 DARPA Imaging Understanding Workshop, pp. 121-130, 1981.

[７]Jean-Yves Bouguet: Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm.
Intel Corporetion Microprocessor Research Labs, 2003.

[８]Carlo Tomasi and Takeo Kanade: Detection and Tracking of Point Features. Carnegie Mellon University Technical Report CMU-CS-91-132, April 1991.

[９] Jianbo Shi and Carlo Tomasi: Good Features to Track. IEEE Conference on Computer Vision and Pattern Recognition, pp.593-600, 1994.

[１０]中沢信明:画像処理を利用した非接触型インタフェースの開発.
“http://www.ccr.gunma-u.ac.jp/4u/Events/2010/Documents/NewTechnologyNo9/NewTechnologyNo9_05.pdf”,2010