

OpenCVで遊ぼう！

Agenda

ひたすらデモとその解説

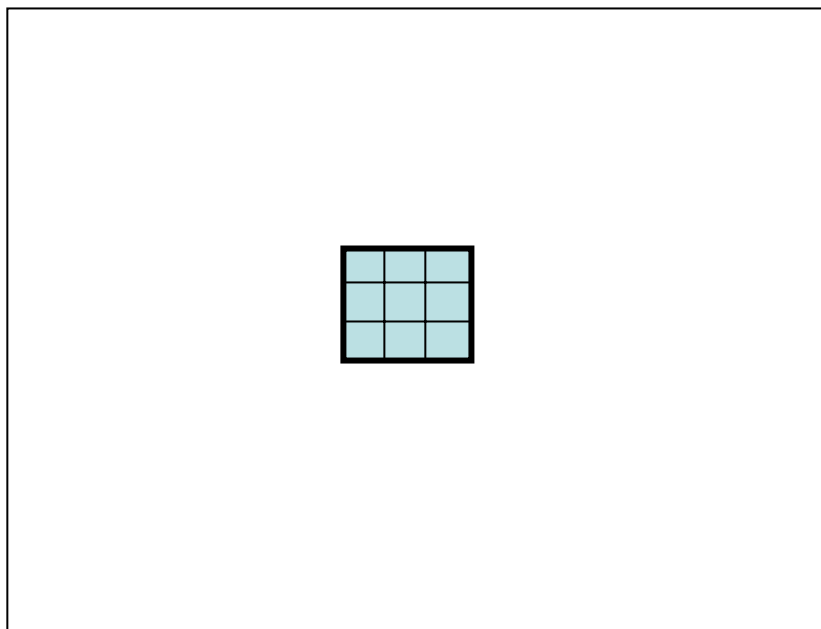
デモはOpenCVに始めからはいっているものプラス自作のものです。

iSightを使ったただのカメラ

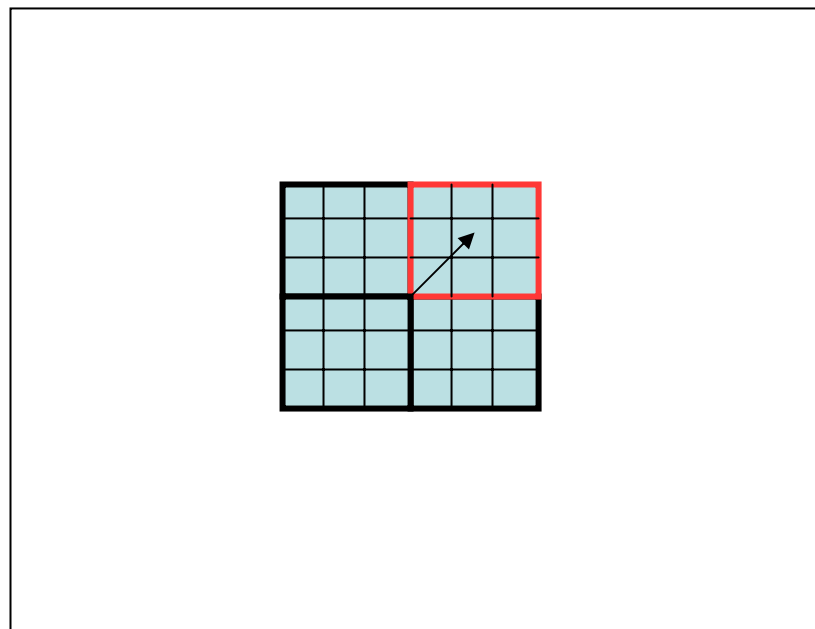
点の追跡 (Optical Flow)

Lucas-Kanade アルゴリズム

Frame 1



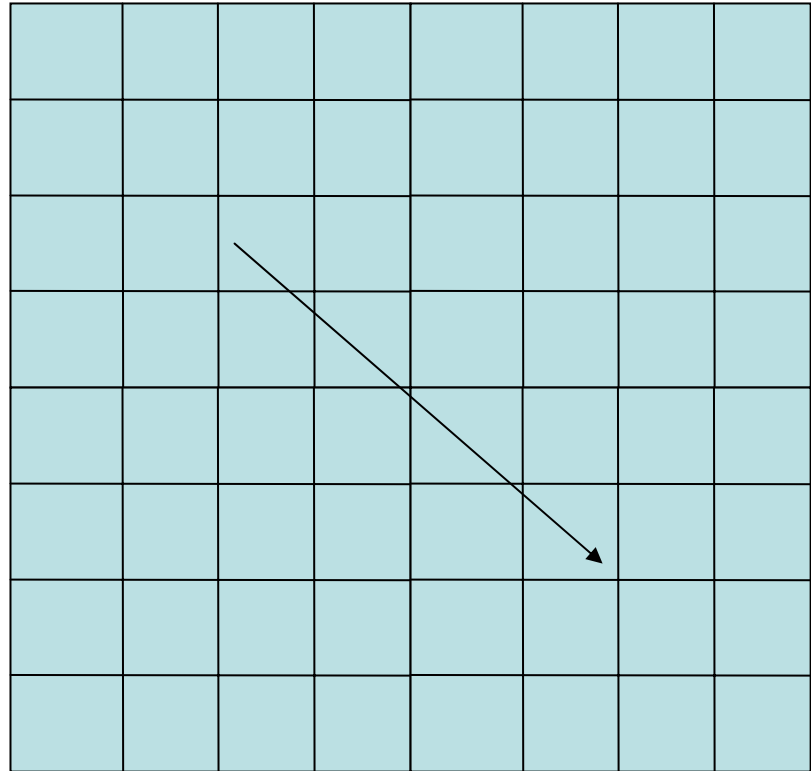
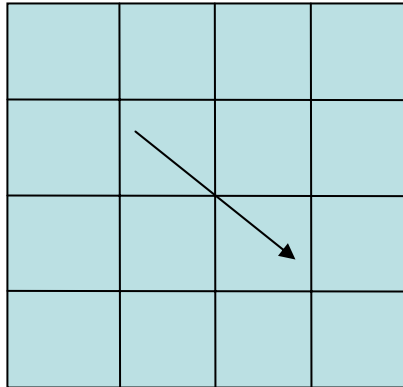
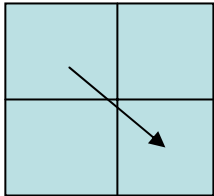
Frame 2



Frame間で、Window毎に比較して、最も近いwindowを探す。
もとのwindowがそこに移動したと解釈する。

	精密さ	ノイズに強い	局所的な動き	大域的な動き
大きいWindow	x	o	x	o
小さいWindow	o	x	o	x

Pyramid Iteration

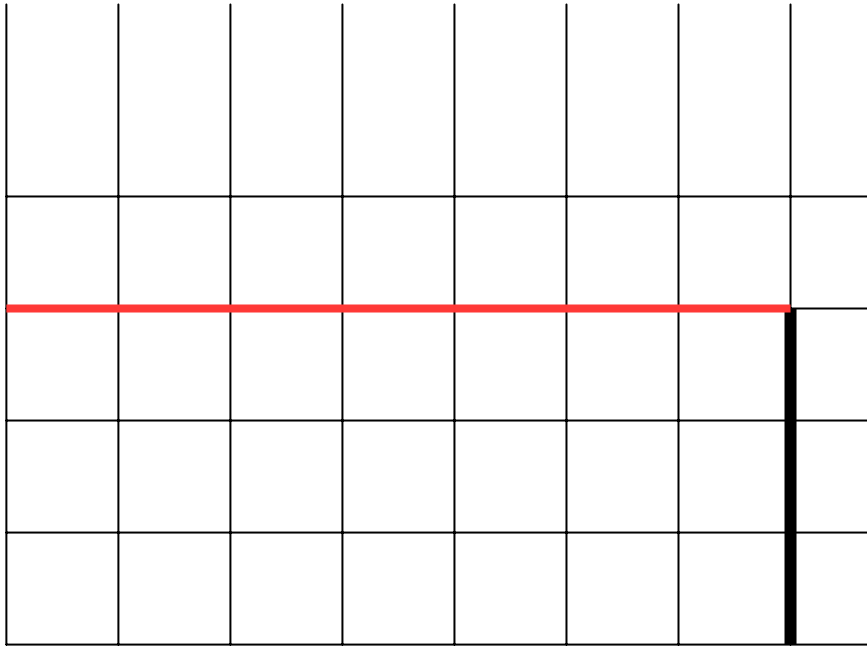


ジェスチャーによる描画

四角形の検出

- Mirror
- 赤成分のみ取り出す
- Equalize Histogram
- 2値化 (demo)
- 境界追跡
- 閉曲線のapproximation (demo->bookmark)
- 4角形を選択
 - 頂点4つ
 - マージンにかからない
 - 面積大きすぎず、小さすぎず
 - 凸包

図形の定着



エッジの検出

エッジの検出

- ピクセル値が急に変化するところはエッジの可能性が高い。
- 微分してみればよい。
- Laplacian $\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$

Canny Edge Detector

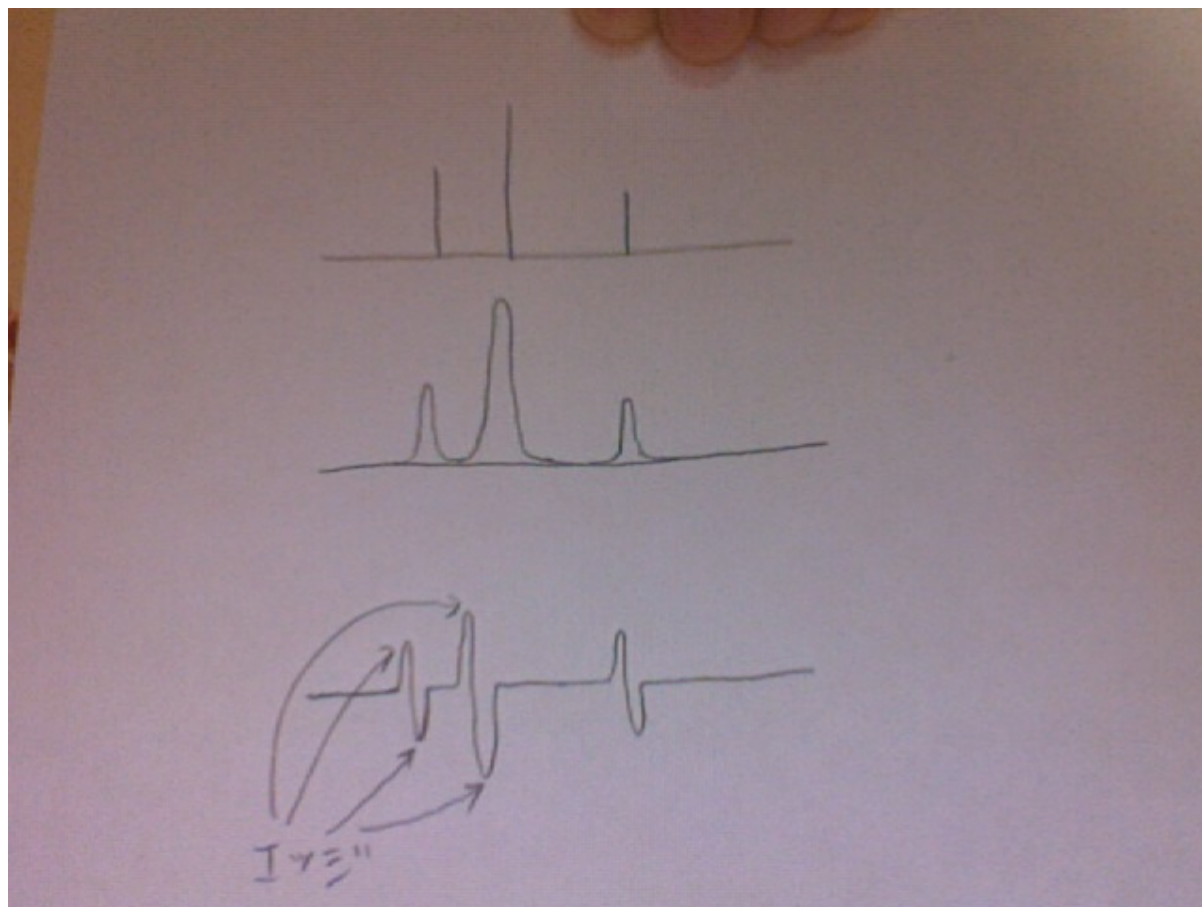
よいedge detectorとは？

- False positive/false negativeがない。
- 正しい位置にエッジを検出する。
- 一つのエッジを複数のエッジとして検出しない。

Cannyによれば、以上から導き出される最高のedge detectorは、

「Gaussianをたたみ込んでから微分」

「Gaussianをたたみ込んで微分」の 一次元的イメージ(実際は2次元)



いかにして速く計算するか

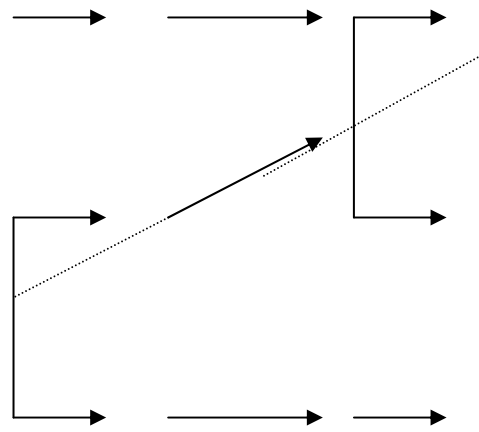
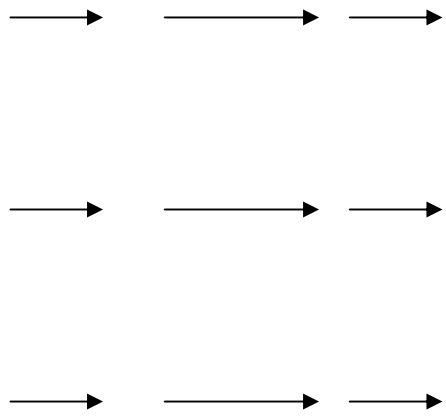
$$C(I) = D * (G * I) = (D * G) * I$$

さらに、2次元のメモリを避けて、1次元のメモリに2回にわたる。

$$C_x(I) = G'_x * (G_y * I)$$

$$C_y(I) = G'_y * (G_x * I)$$

Non-max suppression

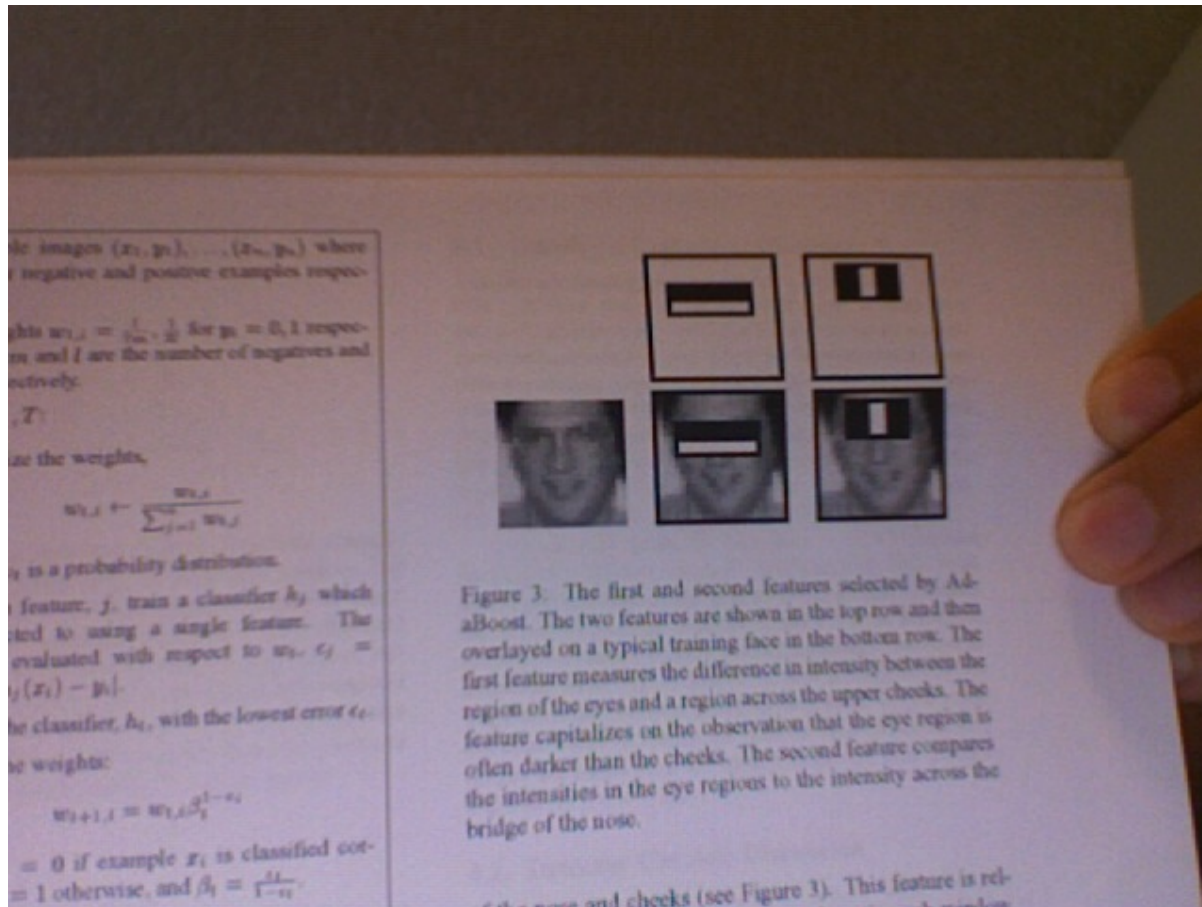


Hysteresis thresholding

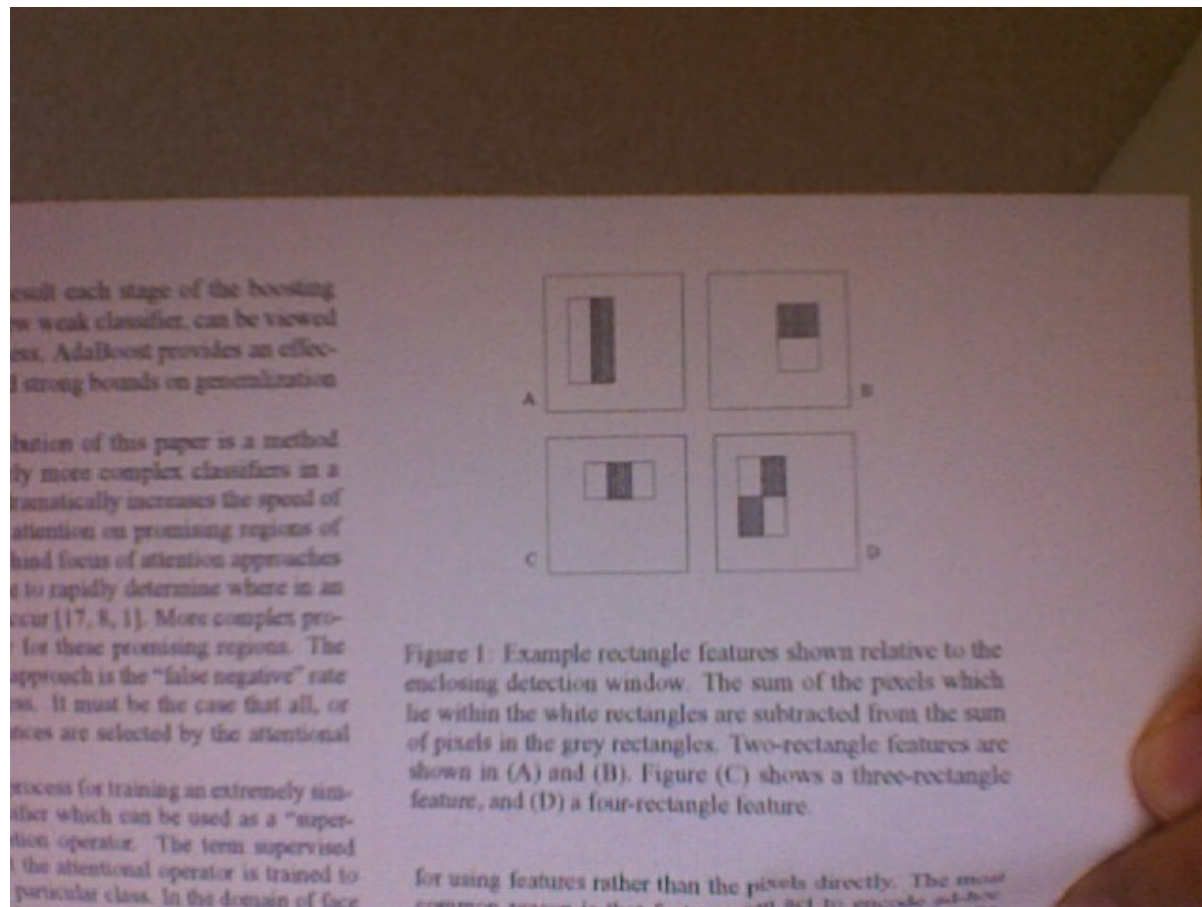
- T_h 以上は無条件でedge pixelと認める
- T_l 以上はedge pixelが隣にあれば、edge pixelと認める

顔認識

Haar-like feature

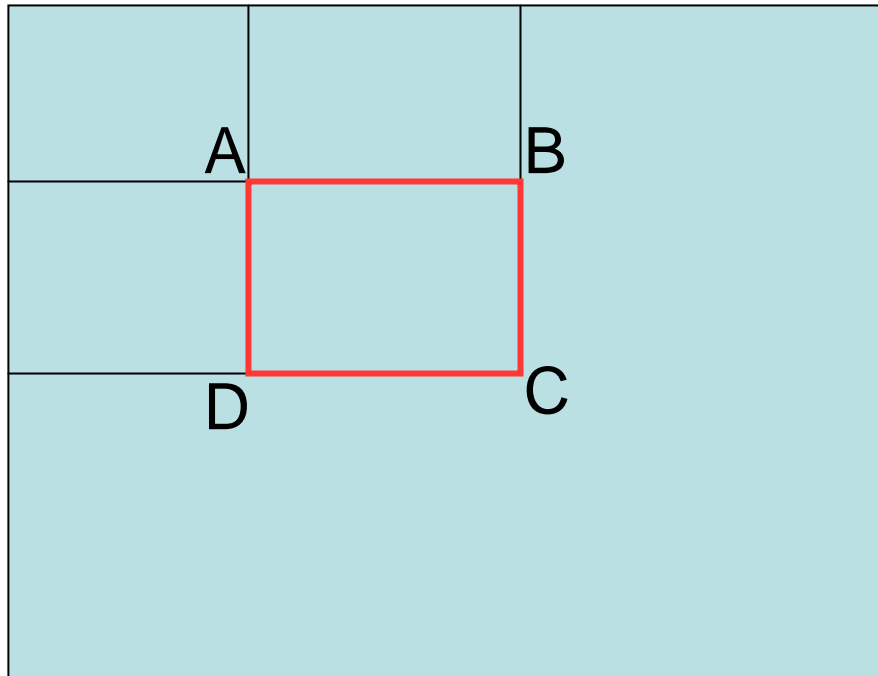


Featureの種類



いかにして速くfeatureを計算するか

- Integral Image



赤い四角の中の積分は、 $C - B - D + A$

問題は、どのfeatureを使うか

Featureの数は数え切れないほどある

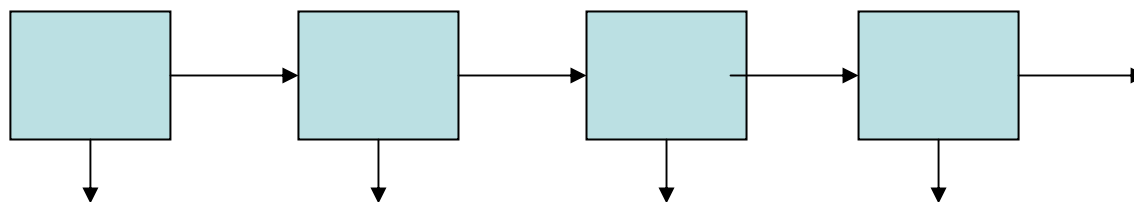
顔のサンプルと顔じゃないイメージのサンプルを多数用意して学習アルゴリズムで有用なfeatureを自動選択する。

Adaboost学習アルゴリズム

- 全てのサンプルを全てのfeatureでclassifyしてみる
 - 1番成績のよかったfeatureを選ぶ
 - そのfeatureで正しくclassifyできなかったサンプルの重要度を上げる
 - 1番のfeatureは抜ける
 - 残りのfeatureで繰り返し
-
- 最終的にfeatureの列とそれらの成績が得られる。
 - 成績のよかったfeatureは多くの票をもつ原則にしたがって、classifierが構成できる。

Cascade of classifier

- 実際の使用時には、圧倒的多数がハズレ
- 少ない計算量(少ないfeature)でハズレをはじくことが重要



- 当たりまではじかないよう、thresholdを調節しておく

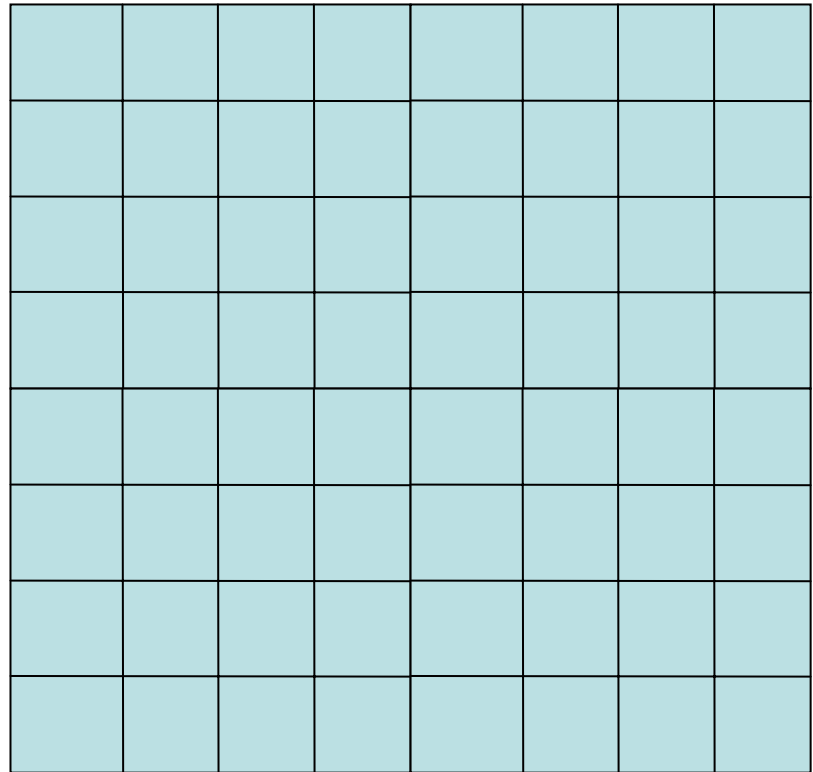
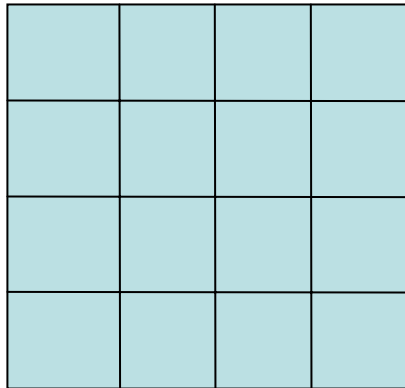
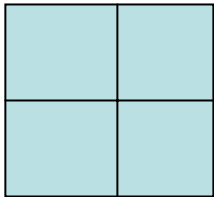
領域認識

領域認識のアイデア

- 似たような色のピクセルどうしを結合、成長させて領域とする
- 問題：領域の核となるピクセルをどこにとるべきか？

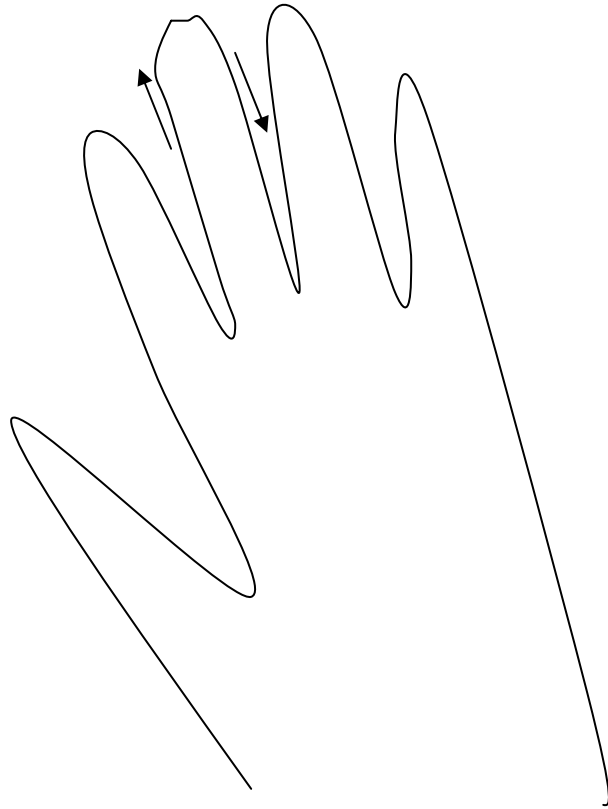
Pyramid Segmentation

領域をピラミッドの上の段から成長させる



指認識

アイデア



手認識

アイデア

- これは手に違いない、という色のみひろって核を作る
- Erosion(タマネギの皮むき)でノイズをとばす
- CannyでEdgeをとる
- Conditional dilationでEdgeにぶつかるまで成長させる

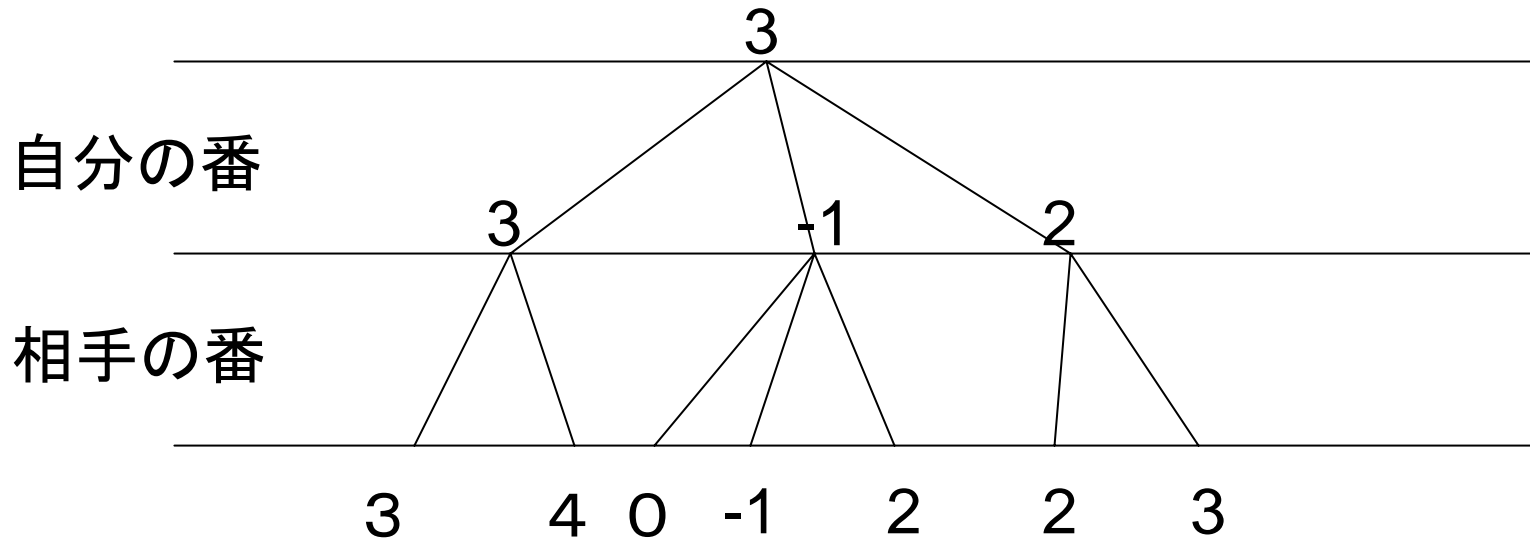
Kalman filter

Kalman filterとは

リアルタイムのシミュレーション
対象物の力学特性を考慮して少し過
去の測定データから現在位置を割り
出す。

応用: ロボットの制御
飛行機を高射砲で狙う

ゲームの木



アルファカット、ベータカット

一つでも返し技のある技は他の返し技を考えるまでもなく使えない。

