

仙台市/仙台市産業振興事業団

ロボット博士の基礎からのメカトロニクスセミナー

C10/Rev 1.0

第10回

カメラと画像処理の基礎

仙台市地域連携フェロー

熊谷 正 朗

kumagai@tjcc.tohoku-gakuin.ac.jp

東北学院大学工学部
ロボット開発工学研究室

RDE

今回の目的

○ カメラと画像処理の基礎

テーマ1：画像処理の目的と要素

- ・ 画像計測の利点と欠点
- ・ 画像計測の三要素：カメラ, 処理, 認識

テーマ2：画像の処理、認識の基礎

- ・ 画像処理
- ・ 画像からの情報の抽出

テーマ3：画像処理の実例

- ・ 家庭用？防犯システム

画像計測

○ カメラで得た画像/映像を利用した計測

◇コンピュータビジョン

- ・ ロボットの目をつくりたい
 - ロボット研究開発の大きな分野
 - 機械系に研究者が案外多い
- ・ 開発された技術の産業への適用
 - ・ 技術としての洗練
 - ・ コンピュータの価格性能比向上
 - ・ カメラの性能向上

画像計測

○ カメラの利点と欠点

◇利点：一度に多くの情報を得られる

◇欠点：一度に多くの情報を得てしまう

- ・ 今時のデジカメ：1000万画素当たり前
人間の目の性能を超える
※全てを同時にだと、まだ微妙？
- ・ 得てしまった多数が混じり合った情報から
必要なものを抜き出す作業が困難。

画像計測

○ 三つの主要要素

◇カメラ（照明，カメラ，光学系，入力）

画像、映像を生のデジタルデータとして
取得する。

◇画像処理（加工、下処理）

認識しやすくするように画像の調整を
行う。単純/膨大な数式処理。

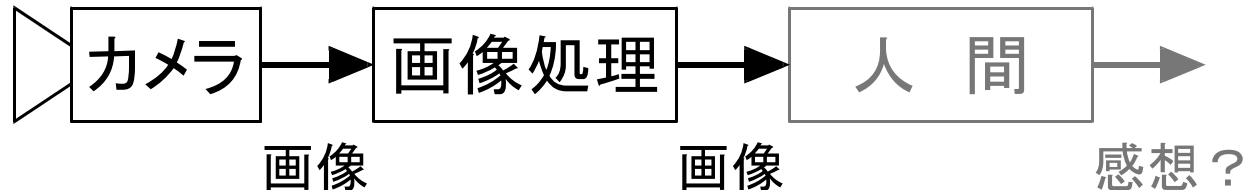
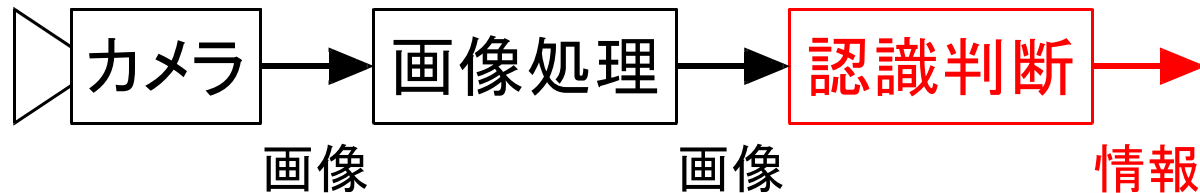
◇画像認識（判断、情報の抽出）

目的に合致した判定、多種多様。

画像計測の構成

○ 認識、判断の重要性

参考：センサ情報処理の基礎

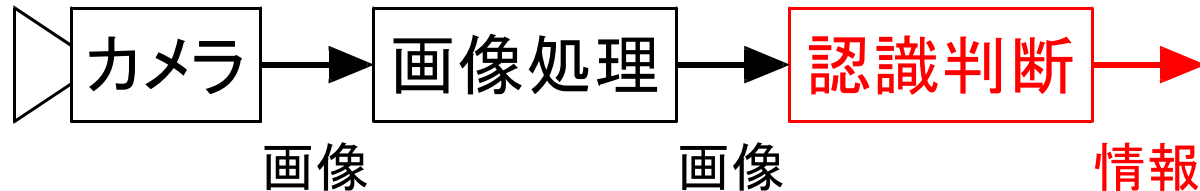


- ・ デジカメ、ビデオカメラも途中までは同類。
- ・ メカトロは「自動で認識、判断」必須。

← 人間に遠く及ばない

画像計測の限界

○ 認識、判断できるかどうか



- ・ 認識判断アルゴリズムの重要性
方式、精度、速度、演算量
「できました」という研究は10年は様子見
- ・ 認識判断をやすくしてあげる：
 - 1: 画像処理段階を工夫する（強調など）
 - 2: 画像に含まれる情報を減らしておく

画像計測の改善

○ 撮像段階の工夫

◇環境

- ・なるべく対象がシンプルに映るように。
- ・撮影環境を一定にするように。

◇カメラを工夫する

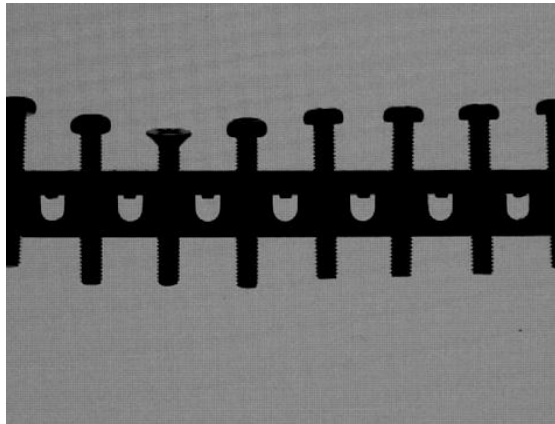
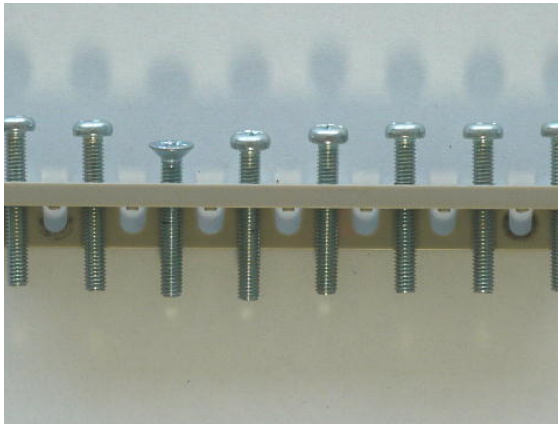
- ・性能の高いカメラ、カメラの設置方法

◇光源を工夫する

- ・光の当て方、光の種類(カラー、赤外など)

画像計測の改善

○ どの画像が見やすいですか？



想定状況：
部品の
品質確認

画像計測の容易化

○ 工場、ラインだからできることが多い

◇環境の一定化

- ・光源の条件、太陽光/環境光の遮断
- ・撮影対象の単純化
- ・撮影対象とカメラの関係の固定

◇一般の環境では...

- ・全部バラバラ
- マージンを広く取る＝判定が甘くなる

画像計測を「自作」するか

○ 低コストの可能性、でも、勧めず

◇物的低コスト

- ・ 単純な処理ならソフトは難しくない
公開されている処理ライブラリ: OpenCV等
画像対応の処理ソフト : LabVIEW等
- ・ パソコンにカメラをつなぐだけ
制御系にPCがあれば追加コスト低い

◇人的高コスト？

- ・ 量産するものでないと「買った方が安い」に

画像計測を「自作」するか

○ 自作の意義

◇オリジナルの処理で付加価値

- ・市販/公開のツールは「汎用のもの」が中心なので、特殊な処理や「汚い手」な処理をできないことがある。
- ・一般化する前に先んじて実現する。

◇人的コスト

- ・画像処理の担当者が必要。
- ・性能評価の手間。

画像の取得

○ 電気信号化と光の取り扱い

◇カメラ

- ・撮像素子
- ・レンズ

◇光源

カメラの基礎

○ 撮像素子、レンズ、出力信号

◇ 撮像素子

- ・ CCD撮像素子, CMOS撮像素子
格子状に並んだ光センサ。
- ・ 取得される1点1点を「画素」と呼ぶ。

※画素＝ピクセル

カラーの場合は、各点、**赤**(R)**緑**(G)**青**(B)

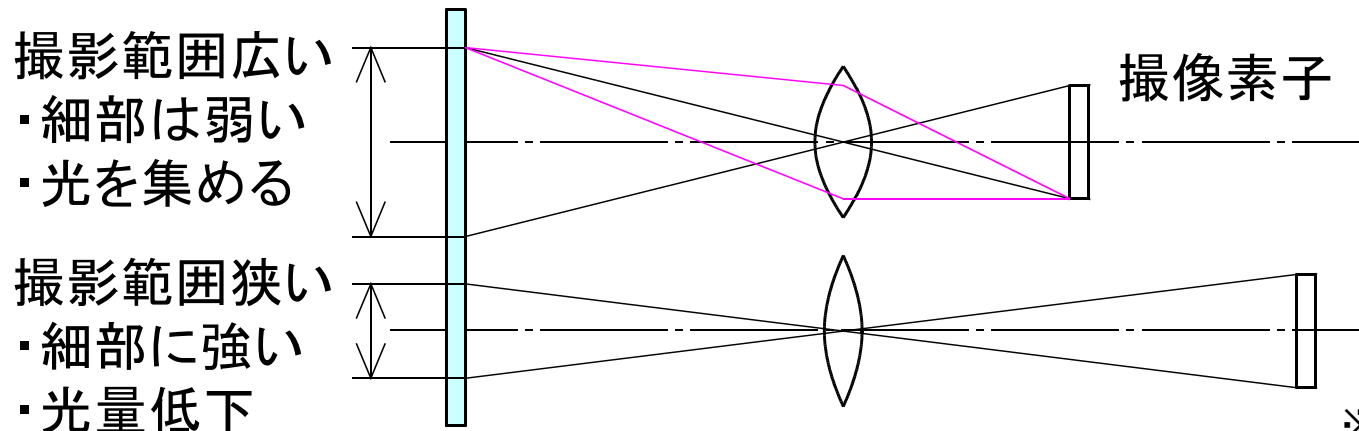
※センサ自体は異なる出力の場合あり

カメラの基礎

○ 撮像素子、レンズ、出力信号

◇レンズ

- ・ 太いほど光量を集められる＝ノイズ低減
（無理に増幅しなくともよい）
- ・ 焦点距離（短いと広角，長いと望遠）



※イメージ図

カメラの基礎

○ 撮像素子、レンズ、出力信号

◇出力信号

- USB
- IEEE1394
- NTSC コンポジット, Y/C分離(S端子)
- その他専用 (アナログ/デジタル)
- オフラインの媒体 (メモリカード)

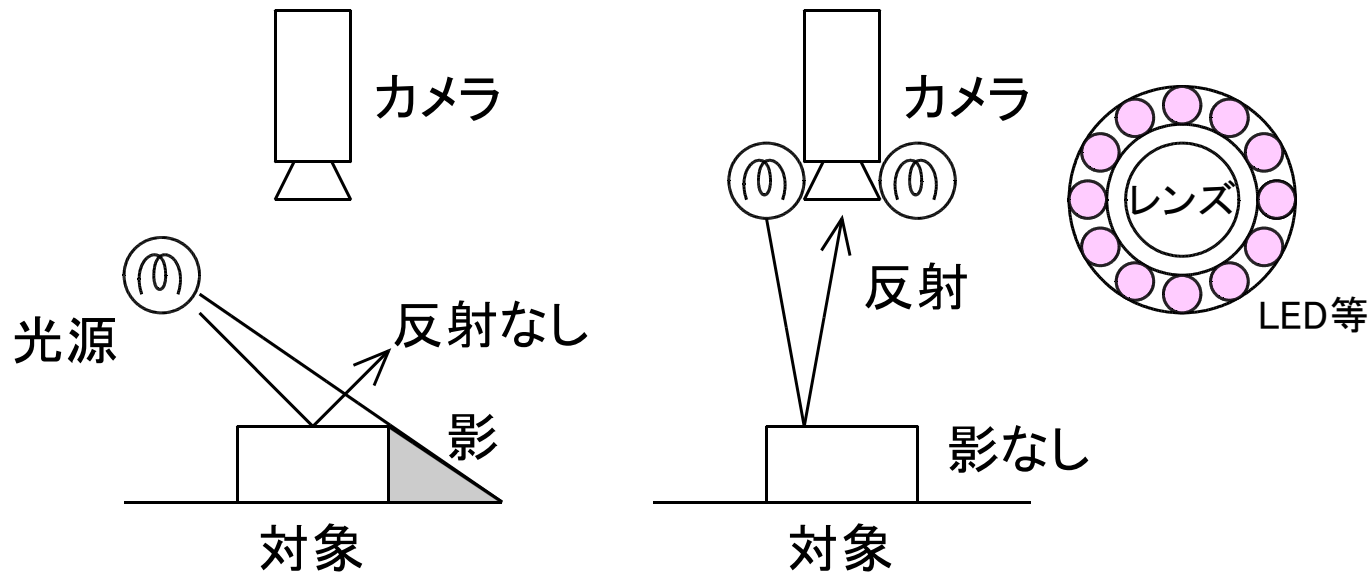
それぞれ、対応する接続方法あり

光源の基礎

○ 撮影するためには光が必要

◇工場ラインでは「望ましい光」を使える。

- ・明るさ、色、波長(例：赤外線)、当て方

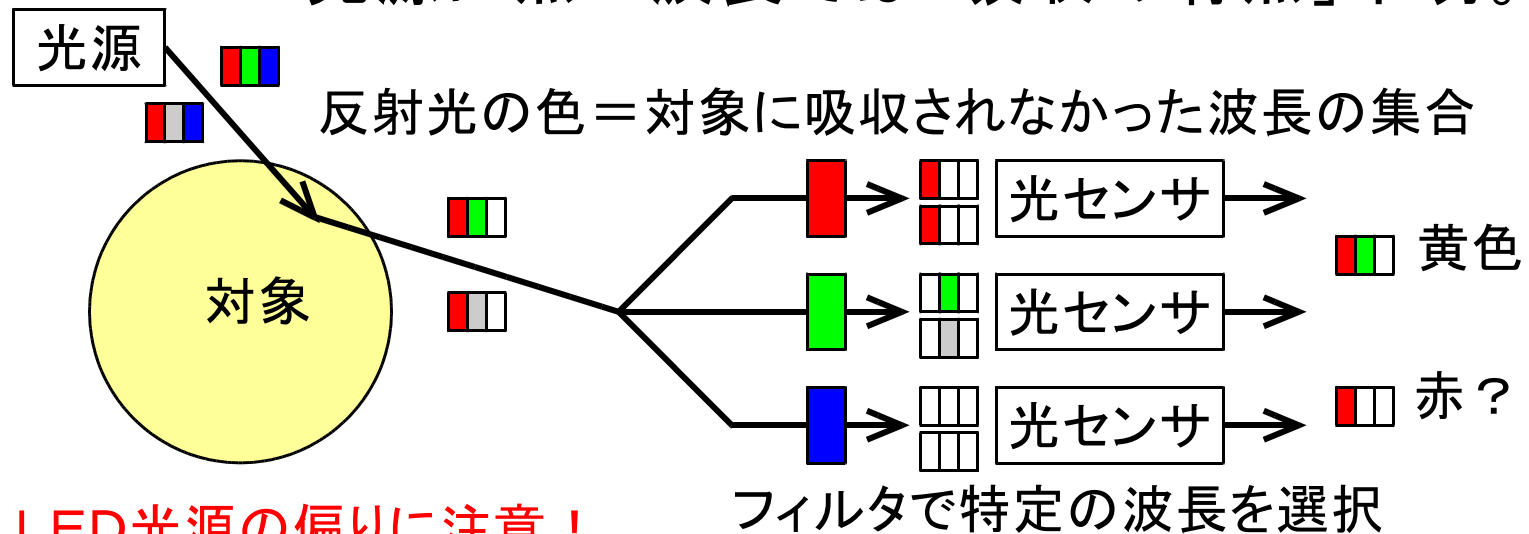


光源の基礎

○ 光源と対象と撮像

◇光源の波長 と 対象の吸光特性

- ・ 色＝「なにが吸収されたか」
- ・ 光源に無い波長では「吸収の有無」不明。



LED光源の偏りに注意！

今回の目的

○ カメラと画像処理の基礎

テーマ1: 画像処理の目的と要素

- ・ 画像計測の利点と欠点
- ・ 画像計測の三要素: カメラ, 処理, 認識

テーマ2: 画像の処理、認識の基礎

- ・ 画像処理
- ・ 画像からの情報の抽出

テーマ3: 画像処理の実例

- ・ 家庭用? 防犯システム

画像処理

○ 画像処理の概要

◇比較的単純 & 数が多い

- ・多くの処理は1画素やその近傍で、
簡単な算術演算のみ行う。
- ・画素数分の処理を行うため、同一作業が
数十万～数百(千)万回になる。
→1処理が1マイクロ秒でも全体で秒単位
→処理速度（秒コマ数, 遅延）

◇認識/判断とは別

画像処理

○ 画像処理の概要

◇ 1画素の処理

- ・フルカラー → モノクロ
- ・フルカラー → HSV（色相，彩度，明度）
- ・2値化

◇ 近傍画素の処理

- ・移動平均フィルタ、メディアンフィルタ
- ・たたみ込みフィルタ（Sobel，ラプラシアン）
- ・膨張、収縮処理

1 画素の処理

○ モノクロ化

◇カラー画像をモノクロにする

- ・人間の目の明るさの特性
- ・輝度 $Y = 0.30R + 0.59G + 0.11B$ (NTSC)
- ・データ量の削減 (1/3)

※最初からモノクロが良い (解像度的に)

補足: Y-Cr-Cb, Y-I-Q

- ・輝度 + (色合い + 濃さの2次元表示)
- ・YCrCbなカメラもある。

1 画素の処理

○ HSV変換

◇RGBカラーから色合いを抽出

- ・ H : 色相 = 色の種類
- ・ S : 彩度 = 色の鮮やかさ
- ・ V : 明度 = 明るさ (輝度Yとは異なる)
- ・ 「色」をターゲットにした処理に便利。

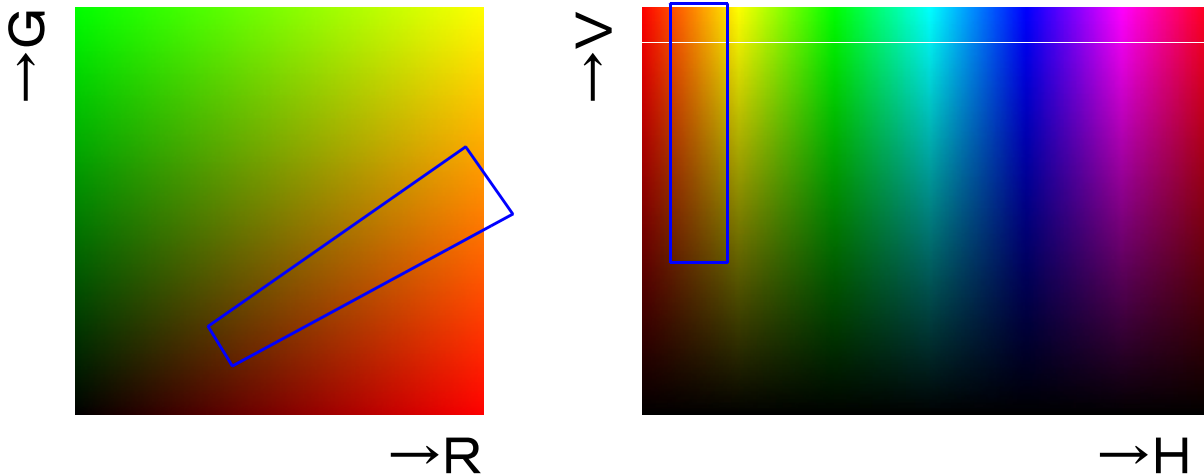
1 画素の処理

○ HSV変換

◇RGBカラーから色合いを抽出

- ・「色」をターゲットにした処理に便利。

例) オレンジ色かどうか

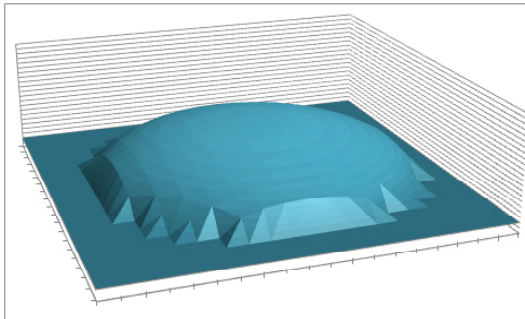


1 画素の処理

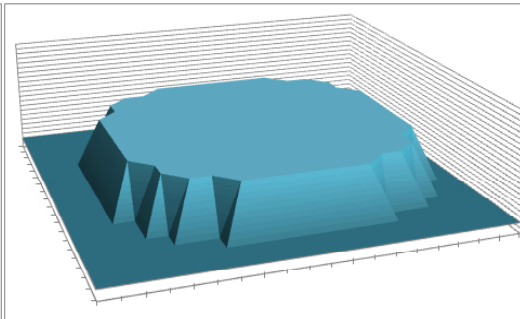
○ 2値化

◇白黒をつける (コンパレータ的)

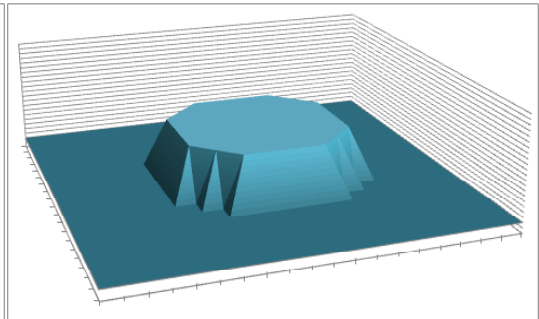
- ・ある評価条件を閾値(しきいち,境界)で判断
 - 閾値より高ければ「白」「1」
 - 閾値より低ければ「黒」「0」



元データ



閾値低



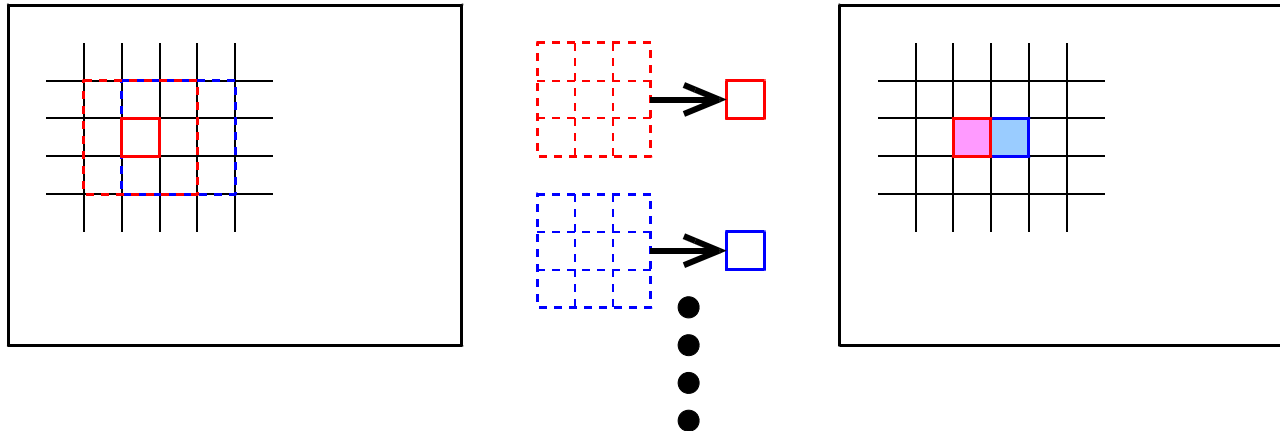
閾値高

近傍画素の処理

○ 3×3 画素 $\rightarrow$ 1画素

◇元画像の1ブロックの画素をもとに処理

- 3×3 が多い（原理そのまま5x5など可）
- 空間的な処理（フィルタなど）

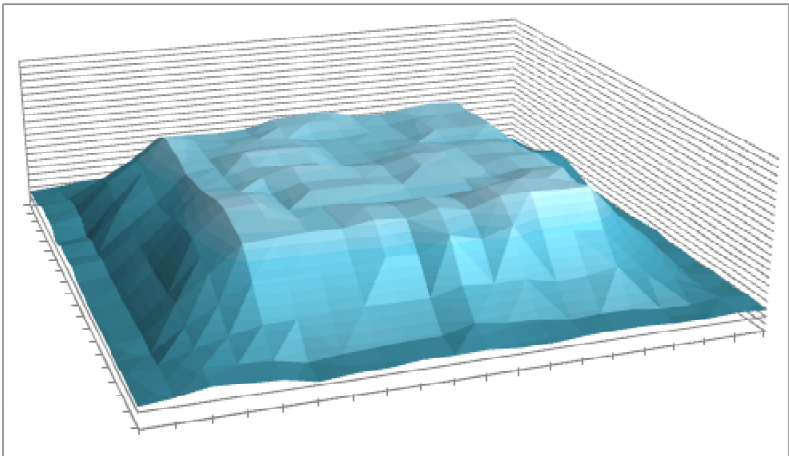
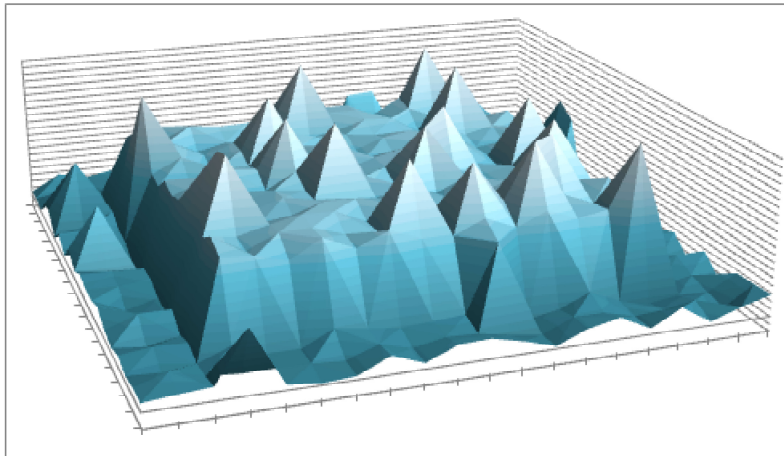
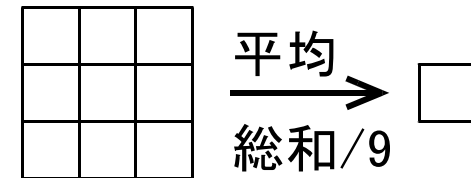


近傍画素の処理

○ 平滑化：移動平均とメディアンフィルタ

◇ 移動平均

- ・ 近傍画素の平均を結果とする。

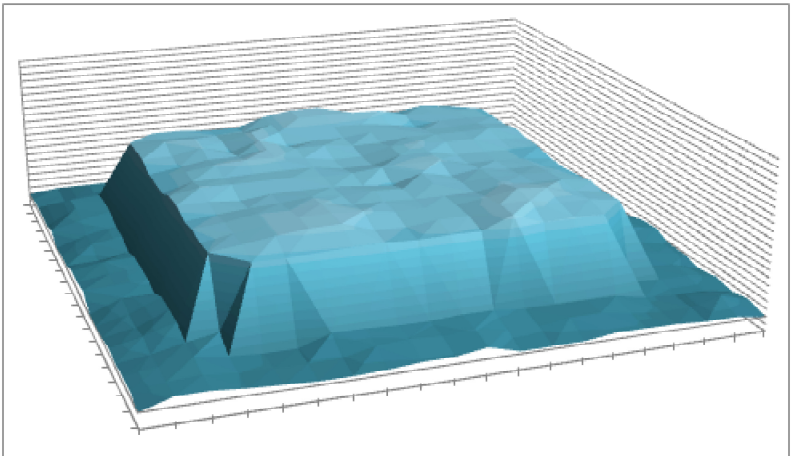
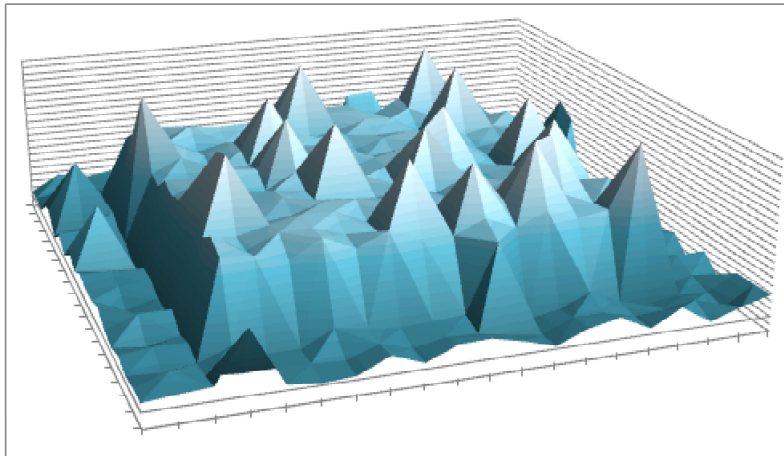
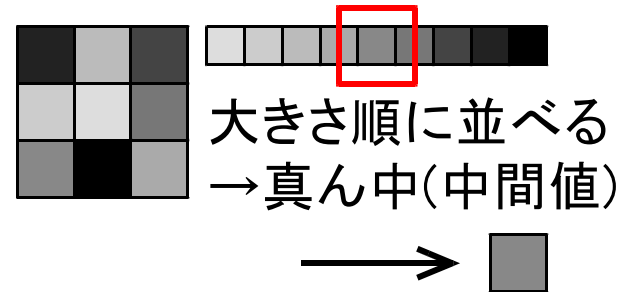


近傍画素の処理

○ 平滑化：移動平均とメディアンフィルタ

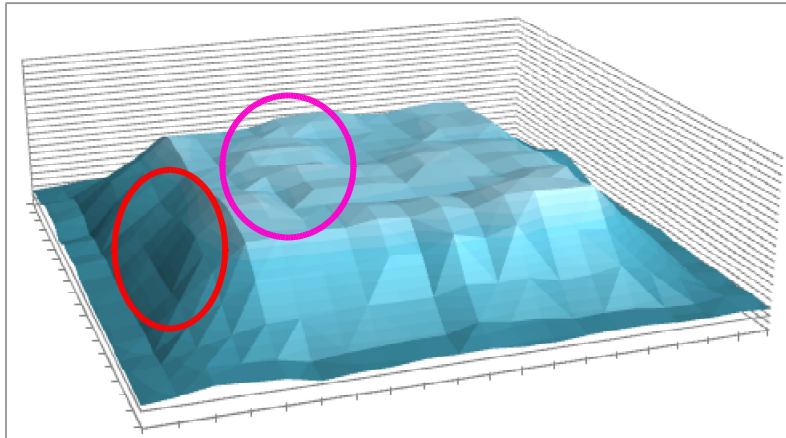
◇メディアンフィルタ

- ・ 近傍画素の中間値を結果とする。



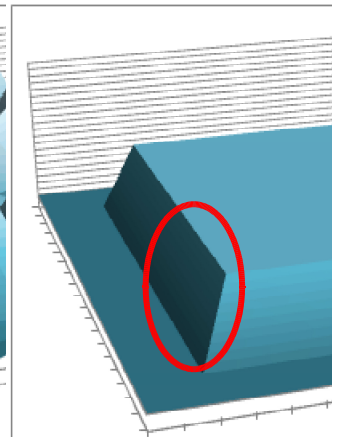
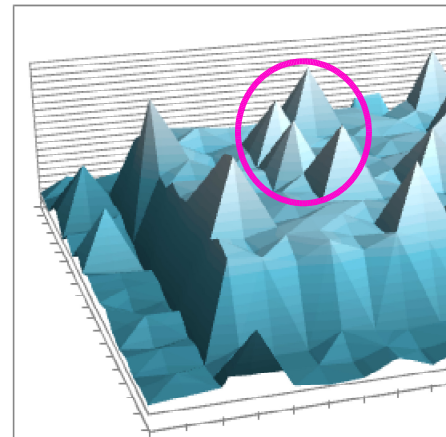
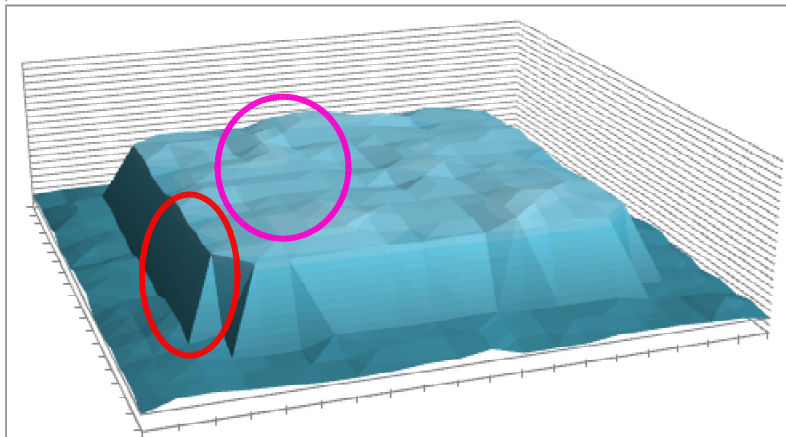
近傍画素の処理

○ 平滑化：移動平均とメディアンフィルタ



メディアンフィルタの特徴

- ・ **角**をなまらせずに、
突発的なノイズを除去。
- ・ 処理量が多い。

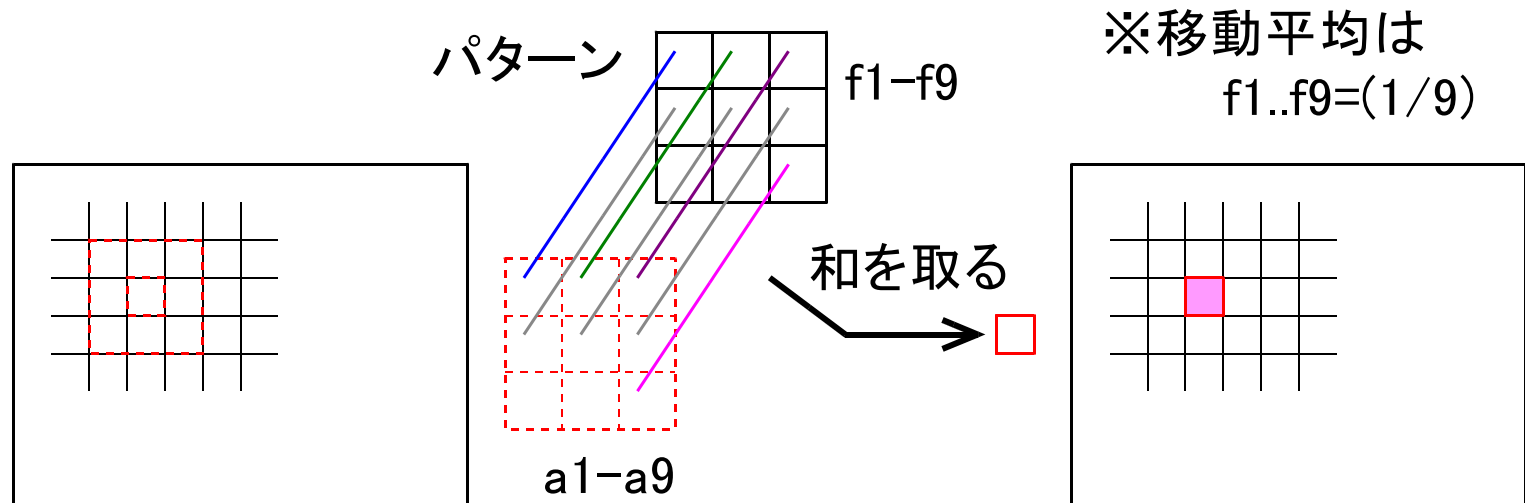


近傍画素の処理：たたみ込みフィルタ

○ たたみ込み演算

◇パターンと積和演算

- $f1*a1 + f2*a2 + f3*a3 + \dots + f9*a9$
- パターンの作り方で種々の演算。



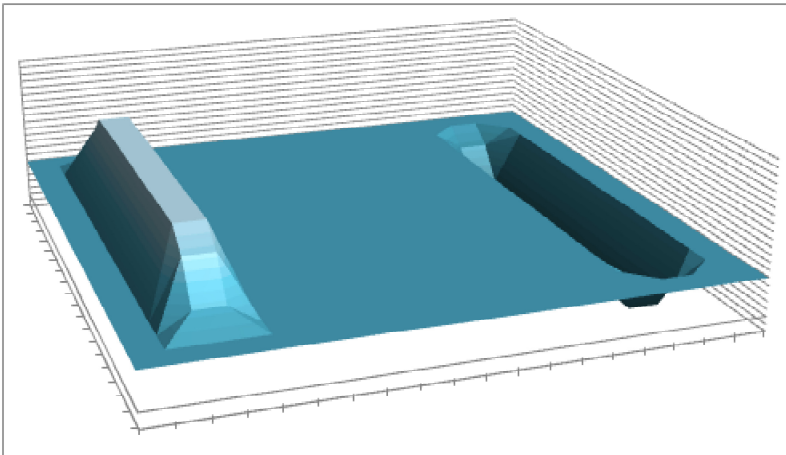
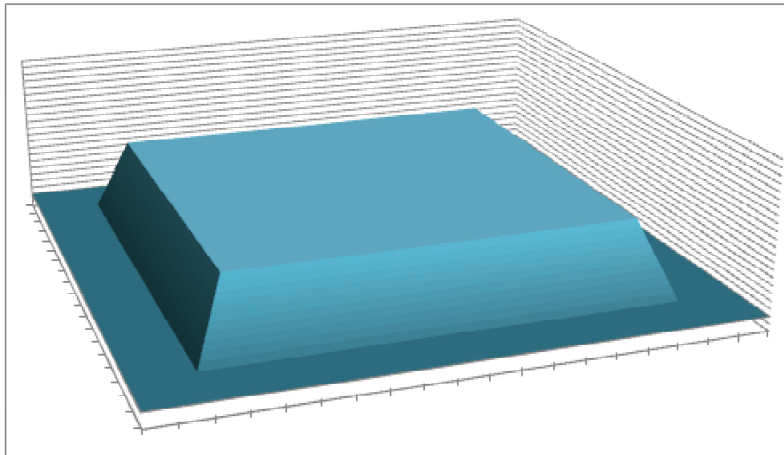
近傍画素の処理：たたみ込みフィルタ

○ Sobelフィルタ（微分型）

◇方向性のあるエッジ検出

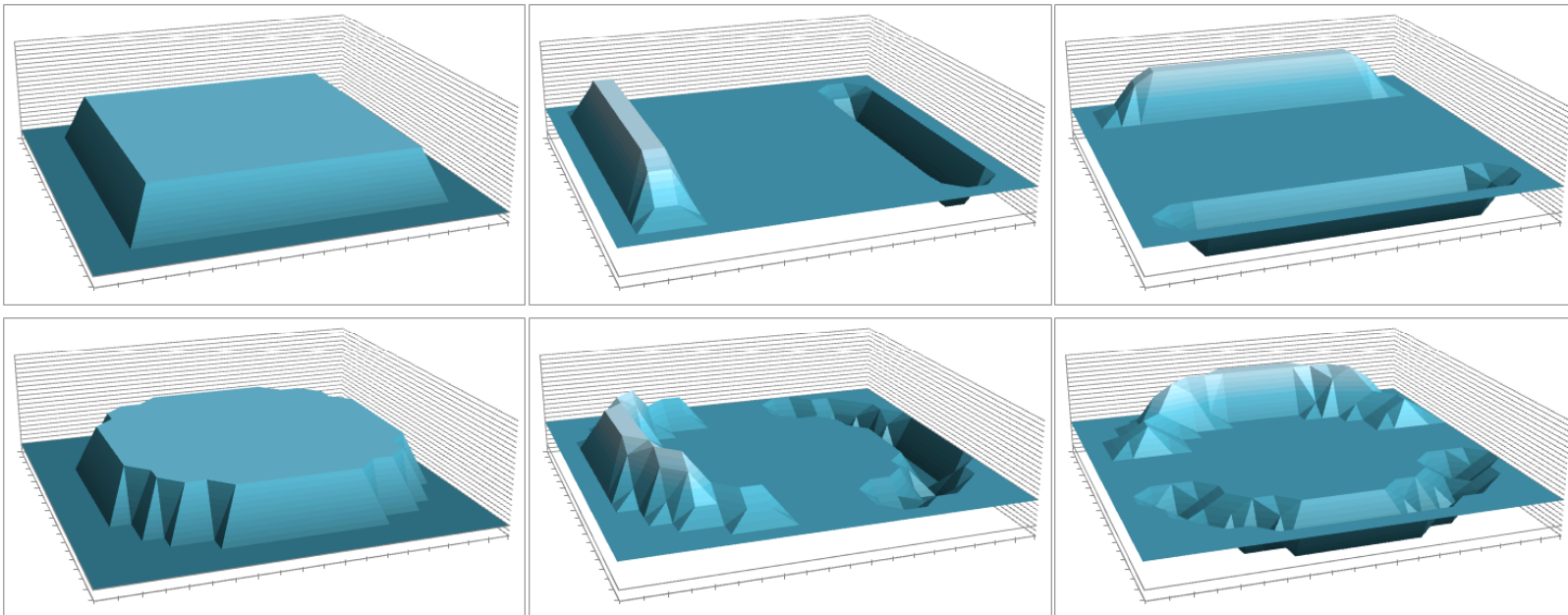
- ・縦方向，横方向のエッジを得られる。

-1	0	1
-2	0	2
-1	0	1



近傍画素の処理：たたみ込みフィルタ

○ Sobelフィルタ（微分型）



元データ

-1	0	1
-2	0	2
-1	0	1

1	2	1
0	0	0
-1	-2	-1

近傍画素の処理：たたみ込みフィルタ

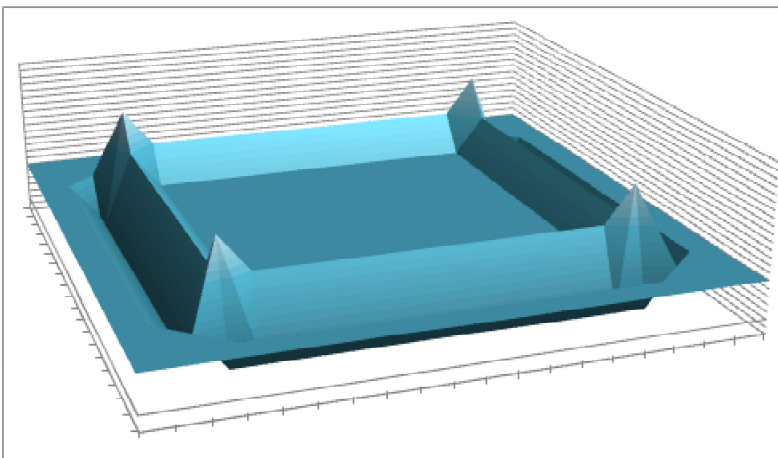
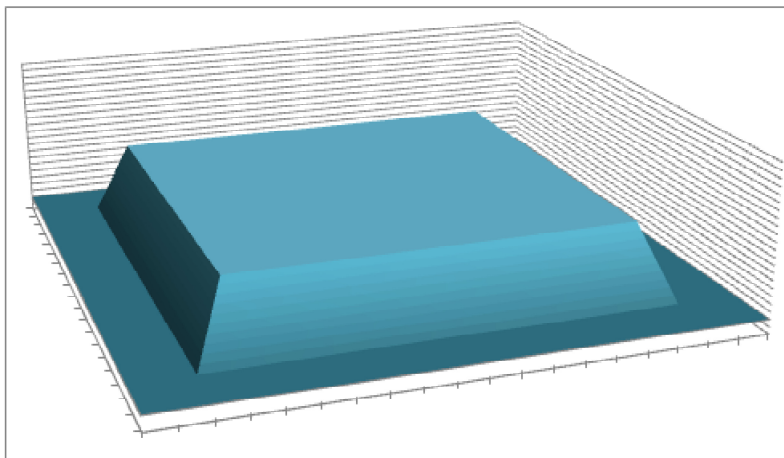
○ ラプラシアン（微分型）

◇方向性のないエッジ検出

- ・ ほぼ方向性なく、エッジを得られる。

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1



近傍画素の処理：たたみ込みフィルタ

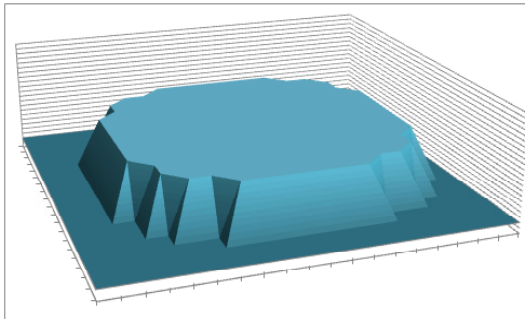
○ ラプラシアン（微分型）

◇方向性のないエッジ検出

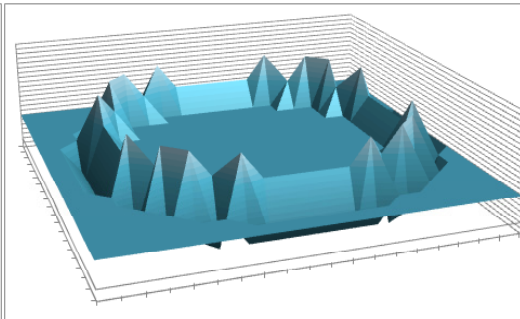
- ・ ほぼ方向性なく、エッジを得られる。

0	-1	0
-1	4	-1
0	-1	0

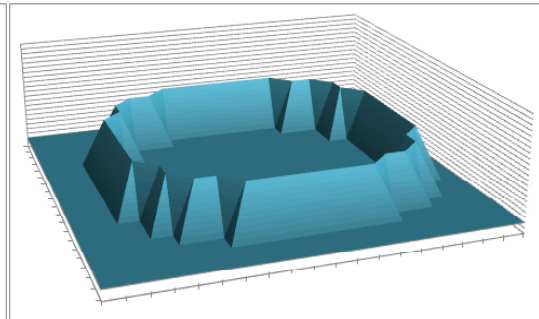
-1	-1	-1
-1	8	-1
-1	-1	-1



元データ



ラプラシアン



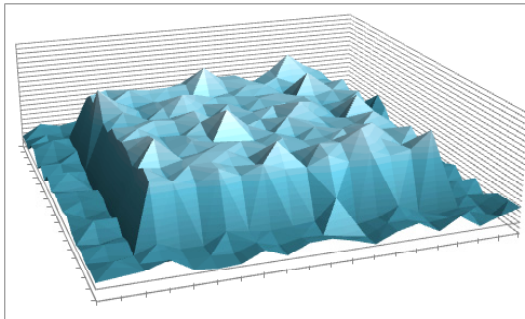
2値化

近傍画素の処理：たたみ込みフィルタ

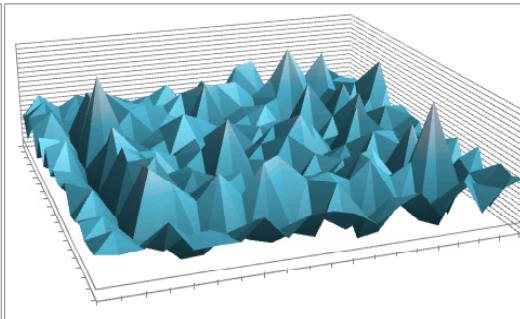
○ ラプラシアン（微分型）

◇微分型(エッジ検出)の注意点

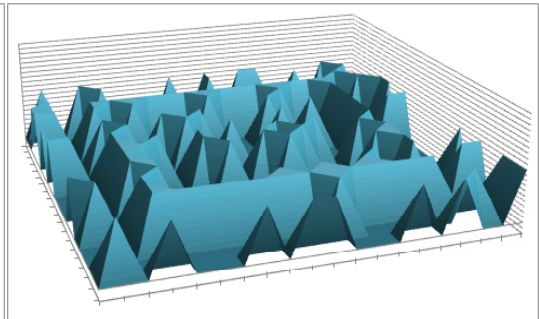
- ・ノイズを拡大する。
- ・ノイズの少ない画像、ノイズ除去(メディアン)



元データ



ラプラシアン

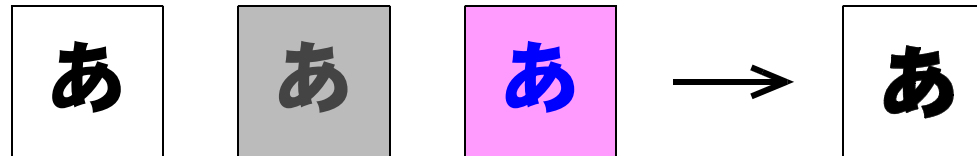


2値化

近傍画素の処理：たたみ込みフィルタ

○ 微分型フィルタの利点

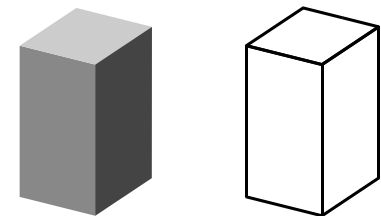
◇形状だけ必要なら、明るさに左右されにくい。



◇エッジの特徴から外形の判断しやすい。

例) 直方体物体の姿勢認識

鉛直な線のための抽出



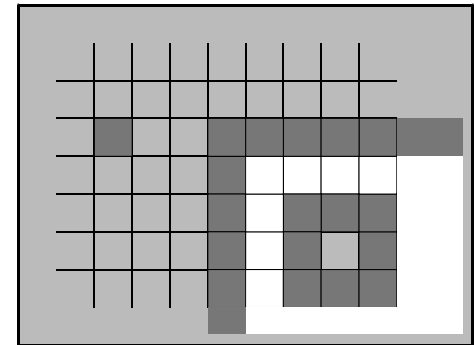
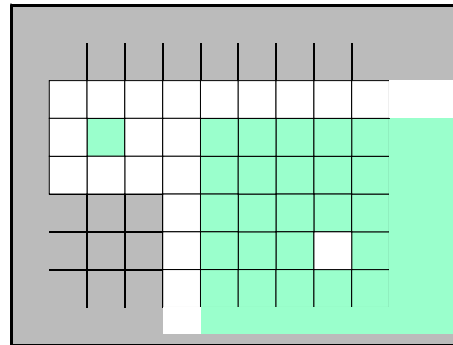
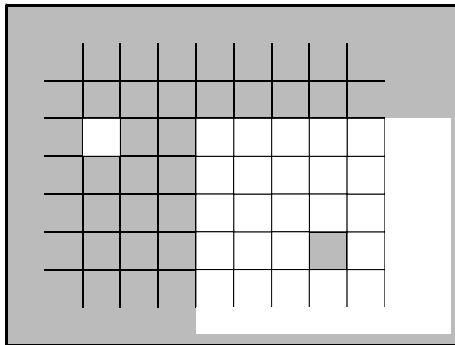
◇前述の弱点に注意。

近傍画素の処理：膨張収縮

○ 2値化後の白点/黒点除去

◇隣に1個でも

- ・ 白があれば白にする = 膨張 → 黒点消える
- ・ 黒があれば黒にする = 収縮 → 白点消える



画像処理

○ 画像処理の概要

◇画像間の処理

- ・差分
- ・テンプレートマッチング

◇画像の変形

- ・歪みの除去、透視変換

◇画像の分析

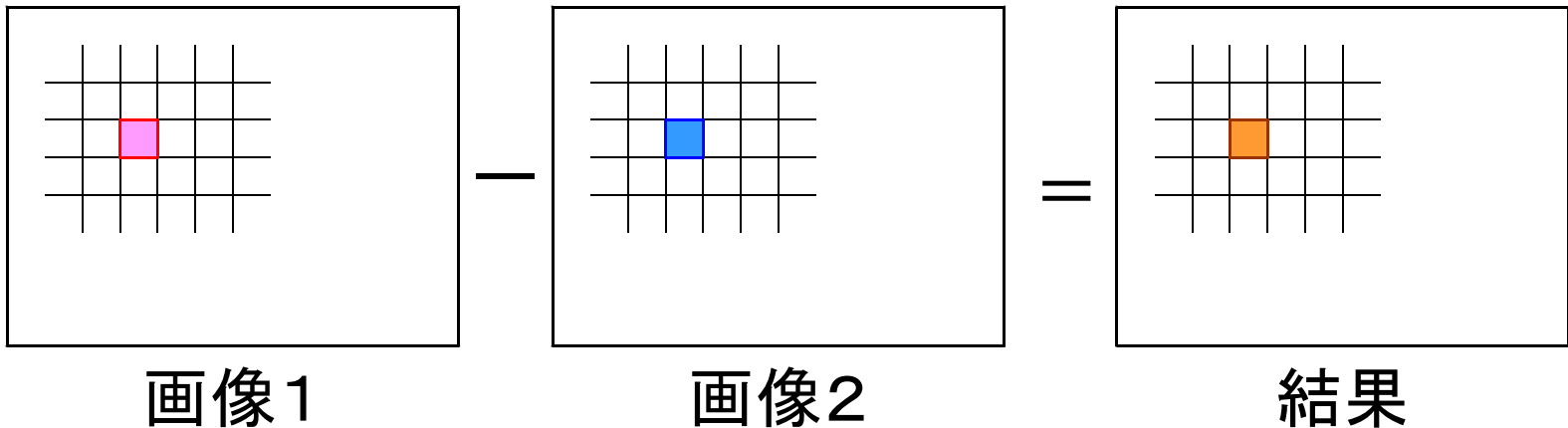
- ・ラベリング、特徴量

画像間の処理

○ 差分

◇画像と画像の差

- ・画素ごとに差をとる。
- ・画素ごとに差の絶対値を取る



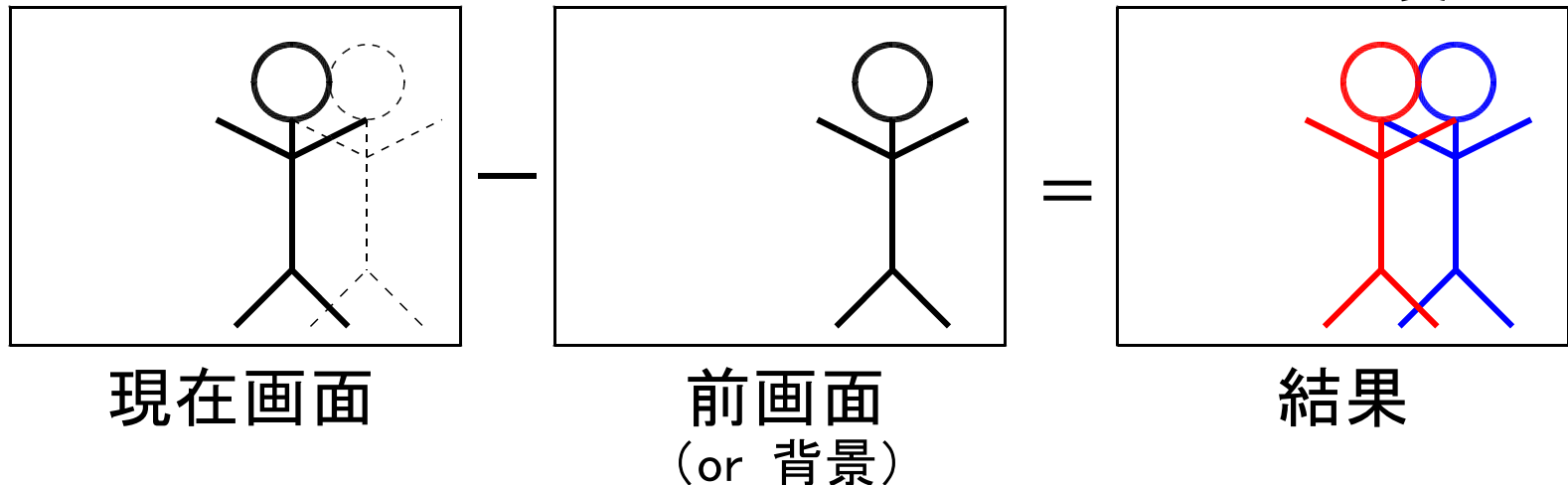
画像間の処理

○ 差分による動きの検出

◇時系列の画像間の差を計算

- ・動きがあれば、その差異が得られる。
- ・背景差分＝「背景」との差、異物検知など。

正負に二つ

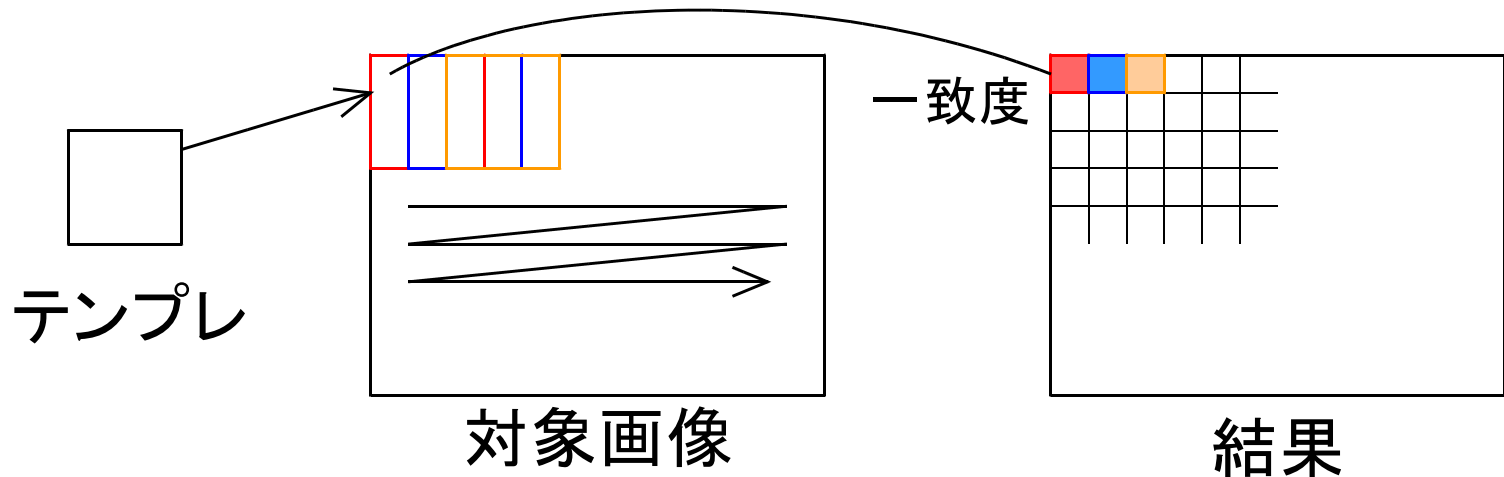


画像間の処理

○ テンプレートマッチング

◇画像の一致度 および そのピーク箇所

- ・ 一般に一方は小さい画像片(テンプレート)
- ・ 差の絶対値の和(SAD)、正規化相関値

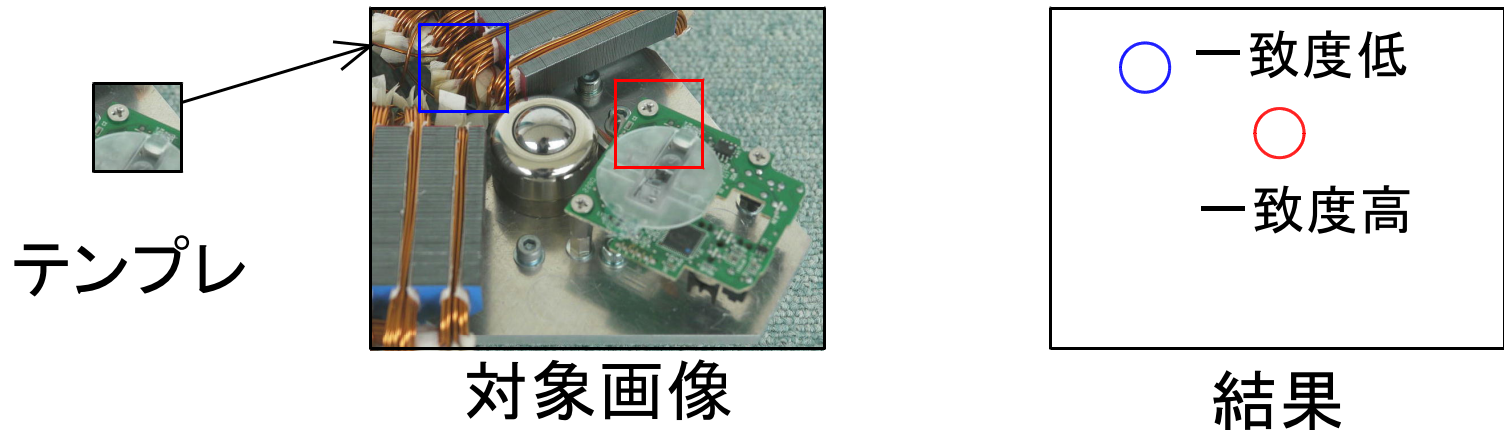


画像間の処理

○ テンプレートマッチング

◇「同一物の探索」「ずれを含めた一致判定」

- ・画面内から特徴的な部分を探す。
- ・部品のマーキング検査など（位置ずれ許容）



画像間の処理

○ ステレオビジョン

◇ 立体視による奥行き情報の取得

- ・ カメラ＋カメラ（＋カメラ...）
- ・ カメラ＋特殊な光源（Kinectなど）
- ・ 右目と左目の画像で対応点を探す
（テンプレートマッチングなど）
→ 三角測量で視差から距離に変換
- ・ 処理量は非常に多い。
- ・ 対象に特徴が必要。

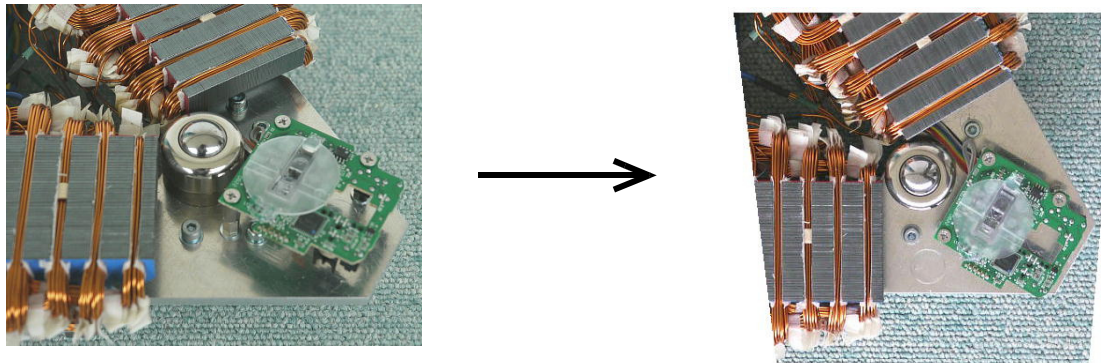
画像の変形

○ 撮影で変形した画像の復元

◇対象を斜めにとってしまったので直す

- ・ 1画素ごとに座標の変換をして貼り直し。
- ・ 変換コスト大 & 完全にもどせない。

→そもそも必要な撮像をする重要さ。



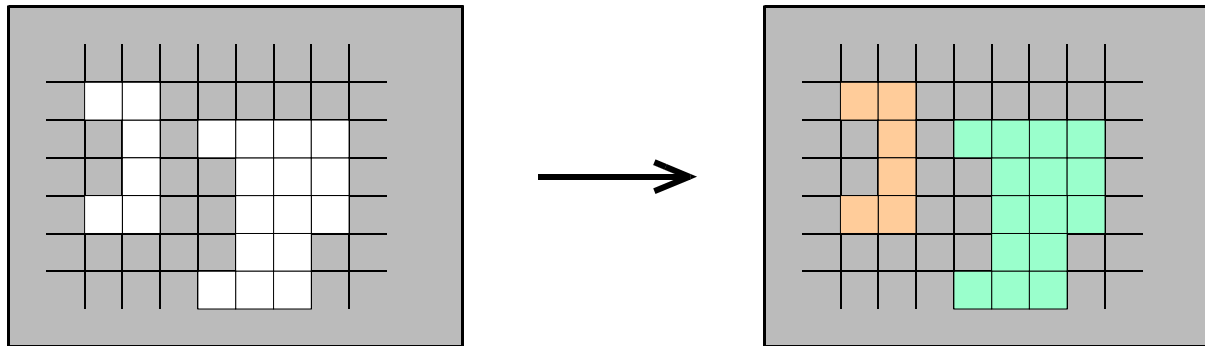
※操作イメージであって実際の変換結果ではありません

画像の分析（2値化後）

○ ラベリング

◇ひとつつながりの点群を探す

- ・色の抽出、2値化などで特徴を絞り込んだうえで、その領域を個々に分けて扱う。

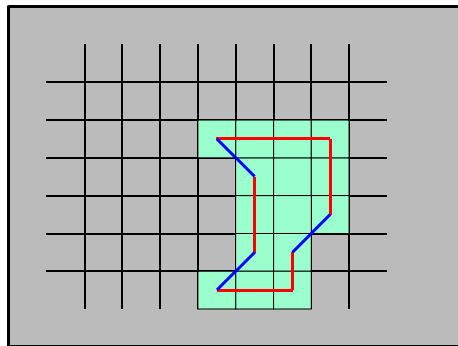


画像の分析（2値化後、ラベリング後）

○ 特徴量

◇連続した画素の図形的特徴

- ・縦横寸法、面積S
- ・外周長L $\rightarrow$ 円形度 $= 4 \pi S / L^2$
- ・重心座標など



縦 5px 横 4px

面積 15px

外周長 = $10 + 1.4 * 3 = 14.2$

円形度 = 0.93

画像の認識

○ 何らかの根拠でYes/No決める

◇あるかないか，正しいか正しくないか

- 画像の**一致度**がある閾値を越えたか。
例)製品の外観検査
- 画像の**特徴を数値化**して、閾値を越えるか。
例)顔の検出，目鼻口，その相対位置など
- 複数条件の組合わせ。
- 根拠をコンピュータに教えられないことは認識できない。

認識のエラー

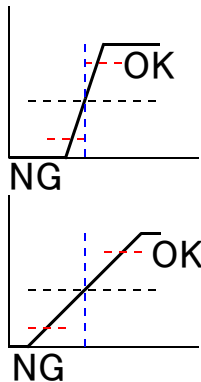
○ エラーには2種類ある

◇NGをNGとして判断する（正常）

◇NGをOKとして判断する（誤, false positive）

◇OKをNGとして判断する（誤, false negative）

◇OKをOKとして判断する（正常）



- ・ 閾値の設定で比率が変わる。

- ・ 製品チェックなどでは

前者：NGの製品を出してしまう

後者：実際より歩留まりが下がる(安全側)

画像の認識

○ 画像認識できるかどうか

◇人間が判断できないことは一般に無理

- ・コンピュータの得意なところ：
連続稼働、判断の一定さ、高速(ものによる)
- ・認識判断力は人間(+道具使用)が圧倒的。

◇「なにを根拠に判断するか」を抽出、指示：

- ・あくまで機械に教えないとならない。
- ・根拠が見つからないと教えられない。

今回の目的

○ カメラと画像処理の基礎

テーマ1: 画像処理の目的と要素

- ・ 画像計測の利点と欠点
- ・ 画像計測の三要素: カメラ, 処理, 認識

テーマ2: 画像の処理、認識の基礎

- ・ 画像処理
- ・ 画像からの情報の抽出

テーマ3: 画像処理の実例

- ・ 家庭用? 防犯システム

画像処理の実装例：自宅監視システム

○ 開発の背景

◇防 犯

- ・いかにも、な監視カメラで心理的防犯
- ・何かあったときのログ
- ・警備サービス + α

◇高齢世帯の玄関防御

- ・悪質セールスへの抑止/証拠映像

画像処理の実装例：自宅監視システム

○ 市販品は？

◇あります

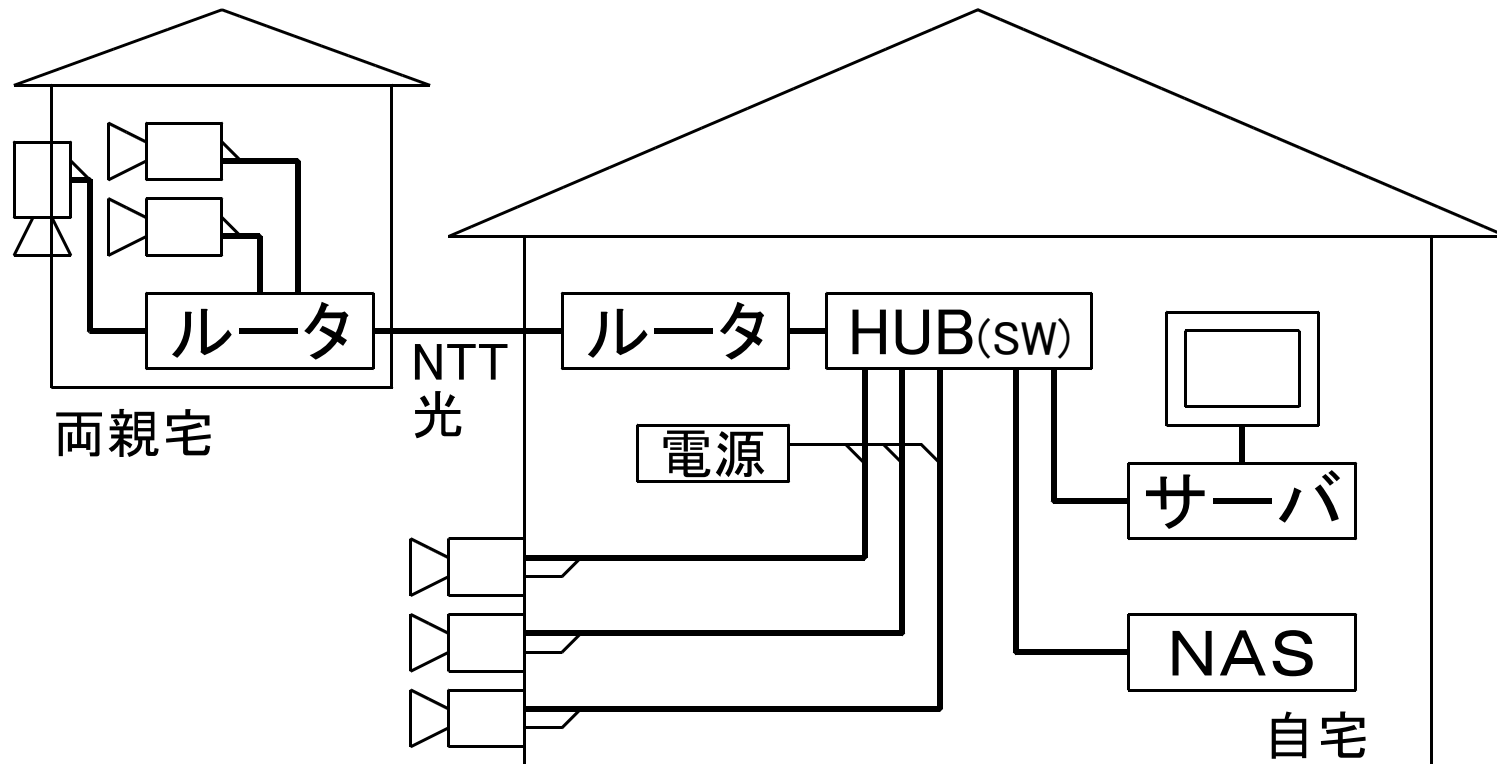
- ・カメラのおまけソフトとして
- ・たぶん独自開発品も多数

◇なぜ自作？

- ・そのためにWindows機を常時稼働させたくない → 自宅のLinuxサーバで兼用
- ・つくれる
- ・自前のほうが設定や改良しやすい

画像処理の実装例：自宅監視システム

○ システムの概要(ハード面)



画像処理の実装例：自宅監視システム

○ 現行仕様

◇ハード

- ・ 自宅および両親宅にHTTPカメラ7台設置。
HTTPカメラ: WEBサーバを内蔵し、ネットにつなぐだけでWEBブラウザで確認できる。
- ・ 両親宅からはNTT回線経由で映像取得。
320x240, 5fps, (常時2台、留守時4台)
- ・ カメラはCat5線の余りペアを利用して
遠隔給電＝宅内はCat5線の配線のみ。

画像処理の実装例：自宅監視システム

○ 現行仕様

◇ソフト

- ・カメラ視界での移動物検出。
- ・移動検知時は5コマ/秒で画像保存、
通常は1コマ/分で画像保存。
- ・1日に3回、画像を結合して映像に変換。
- ・環境変化への対応。
- ・OS起動中は常時動作できるように。

(グラフィック画面時は監視状況表示)

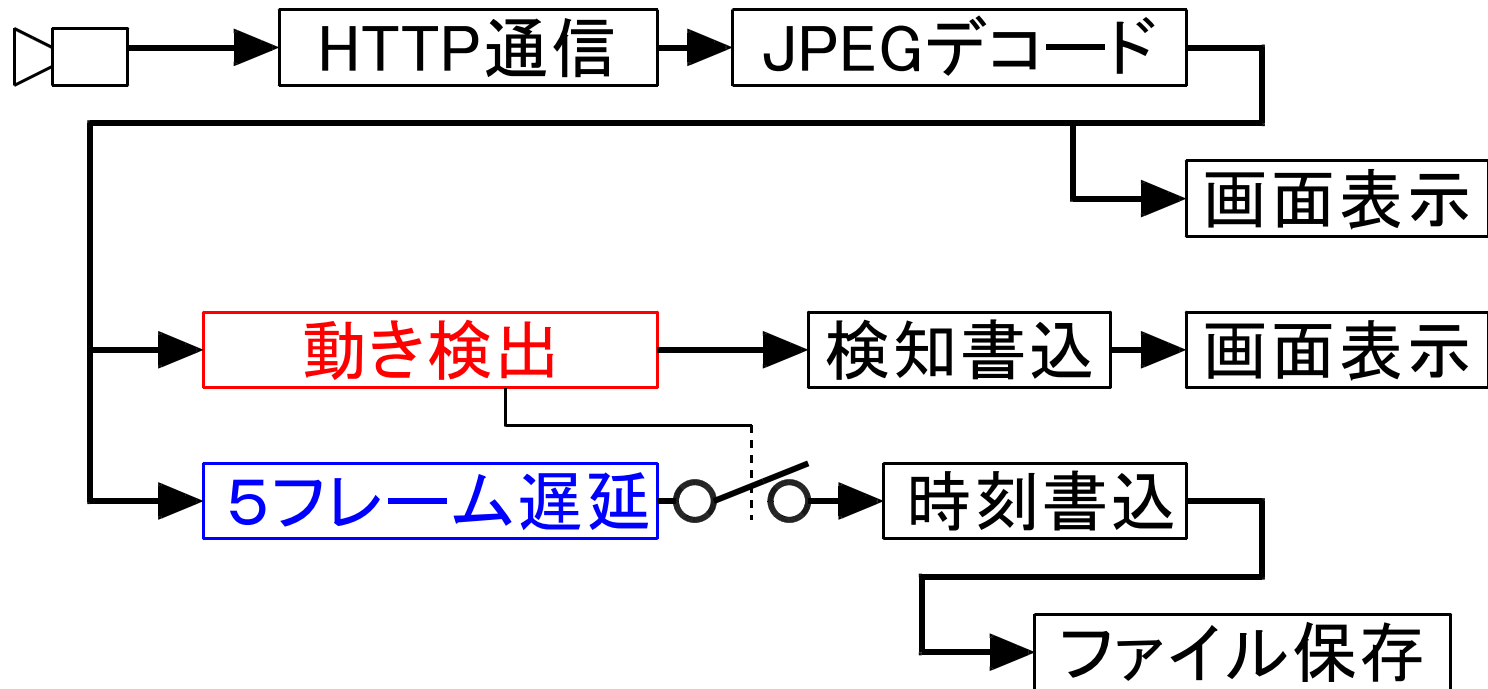
画像処理の実装例：自宅監視システム

○ システムの概要(サーバ画面)



画像処理の実装例：自宅監視システム

○ システムの概要(ソフト面)

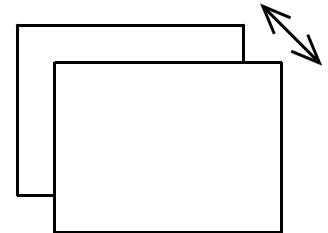


画像処理の実装例：自宅監視システム

○ 動き検出方法

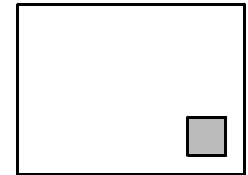
◇基本は前画面との差分

- ・各画素ごとに**差**(絶対値)を計算。
- ・差が大きければ、動いたと見なす。



◇「差の大きさ」の評価

- ・画面全体の**平均値** は**使えない**



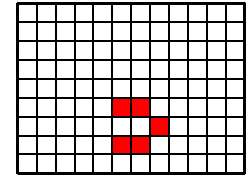
画面の一部の動きは全体平均には影響小。
閾値を下げると全体的なノイズを拾う。

画像処理の実装例：自宅監視システム

○ 動き検出方法（解決策）

◇「差の大きさ」の評価

- ・画面を小さなブロックに分割。
- ・各ブロックごとに差の平均を閾値と比較
→「**動いたブロックの個数**」で検知。

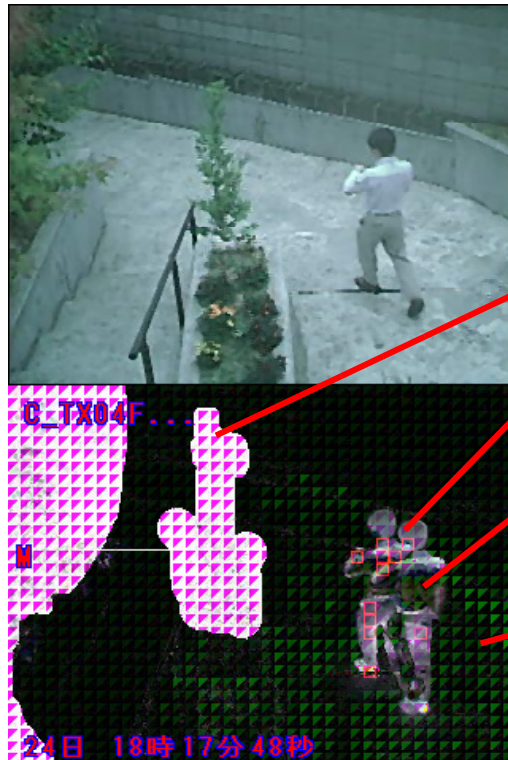


◇「植物が揺れる」「影が揺れる」

- ・画面に検知対象外の領域を設定する。
- ・動きのあるところは閾値を自動的に上げていく。

画像処理の実装例：自宅監視システム

○ 処理結果の例



←カメラ画像

検出対象外

□動き検出

前画面との差

自動的に変更された閾値
※直前に歩き回っていた

画像処理の実装例：自宅監視システム

○ 実用性

◇性 能

- ・人はもちろん、ネコ、鳥すら検知。
- ・暗くなると無理（感知照明は効果あり）。
- ・カメラの性能次第(屋外はつらい)。

◇安定性

- ・3年間ほぼ連続稼働（最長停止は震災の4日）。
- ・その期間の映像はNASに保存。
- ・閾値動的変更で、誤検出が大幅に低下。

まとめ

○ 画像計測の基礎

- ・ 画像は多くの情報を含む(含みすぎる)。
- ・ 画像から情報を得るには、認識が必要。
- ・ 情報の取り出しをしやすくするために、
撮影の工夫で画像をなるべく単純化する。
- ・ もし、画像以外の直接的な計測手段があれば、画像の利用を避けるべき。
- ・ 「人間並み」の判断は一般に困難。

まとめ

○ 画像処理

- ・ 認識の手段として、認識の前処理として、比較的単純な画像処理を行う。
- ・ ノイズ除去にメディアンフィルタは効果的。
- ・ 微分系の演算はノイズ強調に注意。
- ・ 処理の組合わせ、手順、アレンジは重要。
- ・ 画像処理の手法はカメラ画像のみではなく、2次元のデータに対しても効果がある。