2014年度　卒業論文

**画像の平滑化処理を用いた**

**パーティクルフィルタによる物体追跡**

法政大学　理工学部　応用情報工学科

11X3014　井垣　円

|  |  |  |
| --- | --- | --- |
| 指導教員 | 提出確認日 | 印 |
| 平原　誠 |  |  |

目次

1. はじめに
2. OpenCVとは
3. テンプレートマッチング
4. パーティクルフィルタ

４.１　予測

４.２　評価・観測

４.３　選択

1. 切り替え追跡手法

５.１　テンプレートマッチングからパーティクルフィルタ

５.２　パーティクルフィルタからテンプレートマッチング

1. 画像処理（平滑化）

６.１　ガウシアンフィルタ

６.２　メディアンフィルタ

６.３　輪郭保存フィルタ

1. 結果
2. 考察

謝辞

参考文献

付録

1. **はじめに**

長年、スポーツ映像の中継は大きな注目を集めている。特に野球やサッカーの中継は多くの人々に支持されているコンテンツである。しかし、ゲーム中の試合の分析には選手とボールを含めたフィールド全体を捉えることが重要であるが、実際に人が目で試合全体を捉えることは困難であり、負担もかかる。この負担を解消するためには選手やボールの精度の高い自動検出技術が必要不可欠になる。またスポーツ中継にかかわらず物体の自動検出技術は重宝されるだろう。コンピュータビジョンの分野では近年，動画像からの移動物体検出，追跡に関する研究が数多く行われている。動画像中から目的の物体を頑健に追跡することが出来れば，セキュリティや人物検出等様々な分野において応用が期待できる。

さらに、安全のために入口に監視カメラを設置している企業や建物は多い。不審な行動をとる人物を検知するためには動画中の移動物体を追跡する技術が欠かせないだろう。

従来の研究では動画像における物体の検出と追跡を目的とし，物体検出法であるテンプレートマッチングとパーティクルフィルタの切り替え追跡手法の提案と検証を進めている。この切り替え追跡手法では，ボールの追跡状態に応じて，テンプレートマッチングとパーティクルフィルタとを自動的に切り替える手法であり、両検出法のデメリットを補う形で機能している。この手法の実行により精度向上に成功したことは確認されている。

本研究では更なる精度向上を目指し画像処理主に画像の平滑化処理を用いた研究を進めた。

1. **OpenCVとは**

OpenCV（Open Source Computer Vision Library）とは、コンピュータビジョンに必要な機能を揃えたC/C++ライブラリ集である[2]。OpenCVそのものは、コンピュータビジョンの研究者/開発者向けのライブラリであり、研究/開発の過程で幾度となく作られてきた処理群が揃っている。本来ならば1から書かなくてはならない処理が関数コール１つで処理可能である。コンピュータにおいて画像や動画を処理するのに必要な、さまざま機能が実装されており、オープンソースで配布されていることから、学術のためだけでなく商品用の目的でも利用できる。オープンソースで開発されていて、誰でも利用できるのでGoogleなど様々なWebサービスやアプリケーションで用いられている。さらに、マルチプラットフォーム対応されているため、幅広い場面で利用されていることが特徴である。

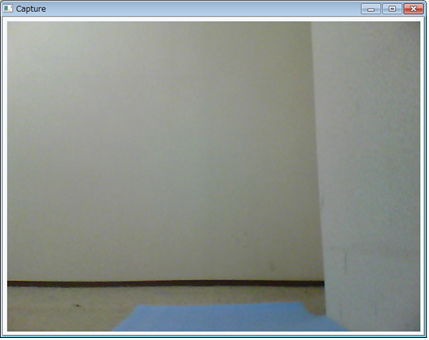


図2.1　カメラキャリブレーション例

OpenCVを使うと利用できる機能のうち、比較的一般的にも知られている機能を以下に列挙した。また、この他、開発者自身が独自にアルゴリズムを実装するために用いることができる基本的な機能や、便利な機能も備えている。

機能例

・オブジェクトトラッキング

・フィルター処理

・機械学習

・カメラキャリブレーション

・オブジェクト検出

・特徴点抽出

・行列演算

・GUI（ウィンドウ表示、画像ファイル、動画ファイルの入出力、カメラキャプチャ）

など様々な機能性が挙げられる。

**３ テンプレートマッチング**

　物体追跡の最も一般的な手法の一つとして，テンプレートマッチングが挙げられる。テンプレートマッチングとは探索対象である動画像の全領域に対してテンプレート画像（図3.1 参照）との相関値を求め、最も相関値の高い領域を探し出し（図3.2 参照）、その領域中心をボール座標として検出する方法である。つまり、あらかじめテンプレートと呼ばれるパターン画素単位で表現したものを用意し，このテンプレートを移動しながら画像と重ね合わせ，類似性の高いパターンを探索対象画像から検出する手法である。

類似性の高いパターンを検出するための類似度の計算には様々な方法があり，代表的な手法に正規化相互相関法(NCC：Normalized Cross-Correlation)がある。これは探索領域内に対して，テンプレート画像との相関値を求め，最も値の高い領域を対象物体領域とする手法である。テンプレートの大きさをとすると相関値 は次式によって表される。

探索領域内で最も高い相関値が閾値(相関閾値)を超えた場合、その領域を赤い矩形で示し、これをボール領域とする。なお、本実験では経験的に求めた0.60を相関閾値として定めた。もし、ボール領域が2つ以上になってしまった場合は、1番最初に見つかったものを採用するものとする。なお本研究で対象とする映像データはカラー画像である。入力画像，テンプレート画像を赤，緑，青の色画像に分解し，それぞれの色画像に対して正規化相互相関法を用いる。そして，それぞれの色画像の相関値の積を最終的な画像の相関値とする。また，いずれかの色画像の相関値が負の値をとる場合，その他の色画像の相関値に関わらず，最終的な画像の相関値は 0 としている。



図3.1　テンプレート画像



図3.2　テンプレートマッチング適用例

なお，本研究では，テンプレート画像はあらかじめ用意している。ボールの回転による模様の変化に対応するため，回転処理を用いた模様の配置の違うテンプレート画像を4 枚用意した。回転処理を行ったテンプレート画像を 図3.3～図3.6 に示す。実際にはテンプレート画像のサイズは23 × 23 画素である。

次に回転処理について記す。画像を水平からy 軸方向にだけ回転させる処理を行う。画像を回転させるには以下の変換行列を用いた。

回転後の座標は次の通りである。

図3.3　ボールのテンプレート1　　　　　　　　図3.4　ボールのテンプレート2

図3.5　ボールのテンプレート3　　　　　　　　図3.6　ボールのテンプレート4

テンプレートマッチングを行った際の物体の検出・追跡の例を図3.7 及び図3.8 に示す。



図3.7　探索対象の画像



図3.8　探索結果

図3.2 の様に、テンプレートマッチングは全画面に対して探索を行なうことから、全画面探索法とも呼ばれている。テンプレートマッチングには次の様な特徴がある。

(1).　動画像にたくさんのノイズがある場合や、テンプレート画像と類似する箇所が複数ある場合、誤検出が発生しやすい。

(2).　手前にある物体が背後にある物体を隠してしまうというオクルージョンへの対応が困難である。

(3).　キックやヘディングなどによってサッカーボールの形状に大きな変化が見られる場合、テンプレート画像との差が大きくなるため、検出が困難となる。

(4).　全画面探索のため、処理時間が長い。

本研究で用いる動画像では、ビデオカメラからサッカーボールまでの距離によってサッカーボールに大きさの変化が生じる。この、大きさの変化への対応にはテンプレート画像に対してバイリニア法（Bilinear法）を用いる。バイリニア法とは、テンプレート画像の拡大もしくは縮小をした際に、画像の輝度値を修正する手法である。修正方法は、求めたい点に最も近い4 点を基準に、その点からの距離による加重に基づいて輝度値を求める。

バイリニア法の簡略図を図3.9 に示す。求める座標値の周辺の２\*２画素（４画素）を使い、輝度値を直線的に補い、輝度値を求める。求めたい座標値 の輝度値を、周りの4 点を使い表す。座標値 における輝度値を で表すと、 は次式で与えられる。

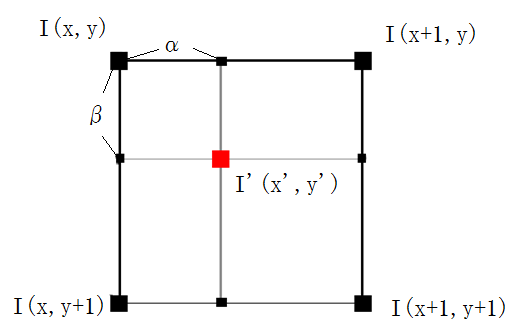


図3.9　バイリニア法簡略図

このバイリニア法により、テンプレート画像に対して、一辺の長さの拡大率が110%,105%,100%,95%,90%の全5パターンの拡大・縮小処理を行なう。

また、バイリニア法で求めることができない外側の箇所に関してはニアレストネイバー法を用いる。（とに0を代入）ニアレストネイバー法とは参照する位置に最も近い位置にある画素の輝度値を参照する。求めたい位置に最も近い値をそのまま使う手法である。求めたい座標をとすると、その位置の輝度値は次式で表される。（ は原画像の輝度値を表す。）

ここで[ 　] は小数部分の切り捨てを表す。座標値を四捨五入し、その画素の輝度値とする。仮に、図3.10～図3.11 の様に拡大処理が行なわれたとすると、バイリニア法が適用できない図3.12 のピンク色の箇所はニアレストネイバー法により、修正が行なわれる。また、図3.13 の黄色の箇所の場合、その上もしくは左の箇所がバイリニア法により修正が行なわれたのち、ニアレストネイバー法によりその輝度値と同じ輝度値とする。



図3.10　拡大前の例

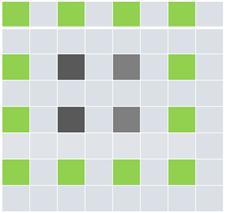


図3.11　拡大後の例

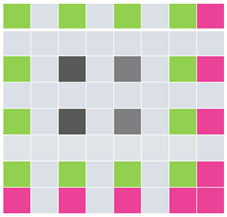


図3.12　バイリニア法が適用できない例１

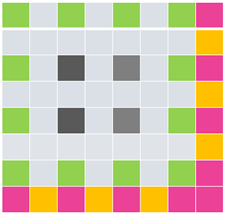


図3.13　バイリニア法が適用できない例２

**４ パーティクルフィルタ**

テンプレートマッチングによる手法では，画像にノイズが多い場合や，テンプレートの特徴が乏しく類似した形状の物体が複数ある場合，誤検出が発生しやすくなる。また，追跡対象の物体が，他のオブジェクトの影に入る等して画面から見えなくなってしまうオクルージョンと呼ばれる問題に対応することも困難である。

このような欠点を補うためにパーティクルフィルタという手法が提案され，その有効性が報告されている[1]。パーティクルフィルタとは追跡対象の状態(座標と速度)の仮説(粒子)を多数生成し，各仮説の当てはまりの度合いを計算して，対象の状態推定を行う方法である。パーティクルフィルタは計算も軽く、処理時間はテンプレートマッチングに比べると短い。また、パーティクルフィルタには次の様な特徴がある。

１．短時間であればオクルージョンが発生してもサッカーボールの位置を推定することが出来る。

２．一度ボールを見失うと再び追跡することが難しい。

パーティクルフィルタは以下のStep1,2,3を繰り返し行なうことで物体の追跡を行なう。

パーティクルフィルタのアルゴリズムを以下に記す。

Step1：予測

　時刻t-1における各粒子をボールの取り得る座標・速度、そしてノイズ(誤差)により変位させ、時刻tにおける粒子とする。変位はサッカーボールの状態方程式に基づいて行なわれる。

Step2：評価・観測

　粒子の座標を中心に動画像を一定の大きさで切り取り、その画像の中でテンプレートマッチングを行なう。これを各粒子について行ない、全ての中で最も相関値の高い領域の中心をボール座標とする。そして、各粒子とボール座標との距離から、各粒子の評価値と重みを計算する。評価値とは、距離が短いほど高い値を取るものであり、重みとは、その評価値を正規化したものである。

Step3：選択

　粒子の新たな選択・撒き直し(リサンプリング)を行なう。距離α内にある粒子は残し、その他の粒子は確率的に残す。次時刻まで残さないと決められた粒子は新たに撒き直される。この時、粒子の数は変わらない。これにより、物体が存在すると考えられる付近に粒子が集中し、追跡が行なわれる。また、次の予測(Step1)で状態が変位するため、もし座標の重なっている粒子が存在していても次の時刻での座標は同じにはならない。

４.１　予測

本研究では，個の粒子があるものとし時刻における，番目の粒子の状態を表す状態ベクトルを次のように定義する。

ここで，対象映像における，時刻の2 次元画像上の横軸の座標値を ，縦軸の座標を とする。 は横方向の速度，は縦方向の速度とする。

番目の粒子の状態は次のサッカーボール状態方程式に従って遷移する。

ここで，粒子は等速直線運動をしていると仮定して，状態遷移行列 は次のように定義した。

また　 は空気抵抗や風，ピッチの凹凸などを吸収するためのシステムノイズを定義する誤差ベクトルである。以下に定義する。

誤差ベクトルの各要素の最大値と最小値はそれぞれ50と-50に定める。この、最大値と最小値の値の範囲からランダムに決定した。

ただし，粒子の位置や速度の上限には制限を課している。これは，物理的に通常あり得ない状態の粒子が生じることを防ぐためである。

４.２　評価・観測

観測値(ボール座標)に対して予測ステップで求めた各粒子の座標の評価値を求める。評価値計算は以下のステップで行う。

1. 粒子の座標を中心として画像を切り取る( 図4.1 参照)。





図4.1　粒子の座標を中心とした際の切り取り画像

その画像の中でテンプレートマッチングを行う。そして，切り取り画像内の最大相関値を求める。

(2) 各切り取り画像から得られた最大相関値を比較し，最も高い最大相関値が得られた領域をボール領域とする。そして，この領域の中心の座標をボール座標とする。一例として、図4.2 にボール座標を紫の点で示す。ここで青い点は粒子、赤色の四角い枠はボール領域である。



ボール領域

図4.2 切り取り画像内でのボール座標

(3) 粒子とボール座標との間の距離から，その粒子の評価値を求める。はボール座標と各粒子とのユークリッド距離である。

ここで，σ は経験的に求めた50.0とする。

(4) 評価値をもとに各粒子の重みを次のように求める。

４.３　選択

得られた重みをもとに粒子を新たに選択し直す。この作業をリサンプリングと呼ぶ。の低い(適合度が低い)粒子を減らし，その代わりに，の高い(適合度が高い)粒子は分裂して増やす。このとき粒子数は変化しない。つまり粒子は確率　に基づいて残すか破棄するか決められる。なお，この後にStep1を行うので同一の状態の粒子に収束することはなく，多様性は保たれる。ここで，ある時刻ｔで10 個の粒子 が図4.3 のように配置されていると仮定する。

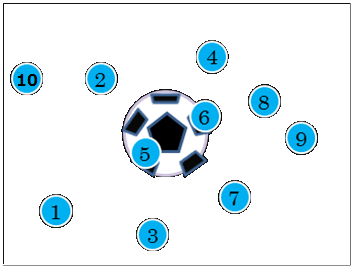


図4.3 リサンプリング前の粒子

この10 個の粒子を重みに従って選択しなおすと，例えば図4.4 のようになる。

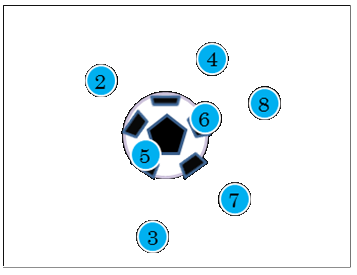


図4.4 リサンプリング後の粒子

これは重みの低い，つまりボール座標との距離が長い粒子が高い確率で破棄され，重みの高い粒子が高い確率で増えた結果である。なお，0番，1番，9番の粒子は消えてしまったように見えるが，それぞれ5番，6番，2番の重みの高い粒子に重なって配置されている。つまり，重みの低い粒子は次のようになっている。

このように，重みの低い粒子は，重みの高い粒子と同じ状態になっている。なお，このステップを繰り返していくと，粒子の多様性が失われるように思われるが, 重なっている粒子は次の予測で，状態遷移するため，現時刻で同じ状態の粒子でも次の時刻では同じ状態の粒子になることはほとんどない。

**５ 切り替え追跡手法**

　テンプレートマッチングは，常に対象領域の全画面を探索することができる。しかし，この手法では，オクルージョンが生じた時刻のフレームにおいて，ボールを検出することができない。また，キックやヘディング等によって，ボールの形状が大きく変化しているフレームでは，ボールとテンプレートとの差が大きくなり，検出が難しくなる。一方，パーティクルフィルタによる追跡は，短時間であればオクルージョンが生じた場合でも，ボールの位置を推定することができる。しかし，一度ボールを完全に見失ってしまった場合，再び追跡することが困難である。このようにテンプレートマッチングとパーティクルフィルタには，それぞれメリットもデメリットもある。これらのデメリットを解決するために，本研究ではボールの追跡状態に応じて，適切な追跡手法を取るように，切り替え制御を行う。パーティクルフィルタでボールを見失ってしまった場合でも，テンプレートマッチングに切り替えれば，追跡状態を回復できる可能性がある。

本手法はまず初期フレームでテンプレートマッチングによって，ボールの検出を試みる。以降のフレームからはパーティクルフィルタによる追跡を行っていく。そして，追跡状況に応じてテンプレートマッチングに切り替える。

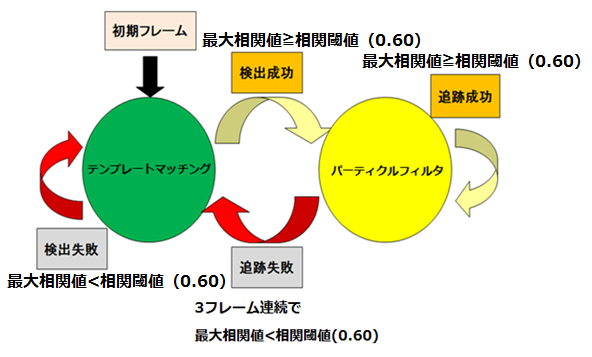


図5.1　切り替え追跡手法全体の概略図

図5.1にテンプレートマッチングからパーティクルフィルタの切り替え及びパーティクルフィルタからテンプレートマッチングへの切り替えの全体の概略図を記す。

５.１　テンプレートマッチングからパーティクルフィルタ

テンプレートマッチングからパーティクルフィルタへの切り替えは，図5.2よりテンプレートマッチングでの最大相関値が，相関閾値 0.60 以上である場合に行われる。なお，この値 0.60 は予備実験から経験的に設定したものである。また，パーティクルフィルタへの切り替えの際，各粒子はテンプレートマッチングから得たボール座標周辺にランダムに配置する。また，テンプレートマッチングにおける相関閾値（つまり本研究では 0.60 ）を超える領域を発見できなかった場合は，次のフレームでもテンプレートマッチングによって，再びボールの検出を試みる。

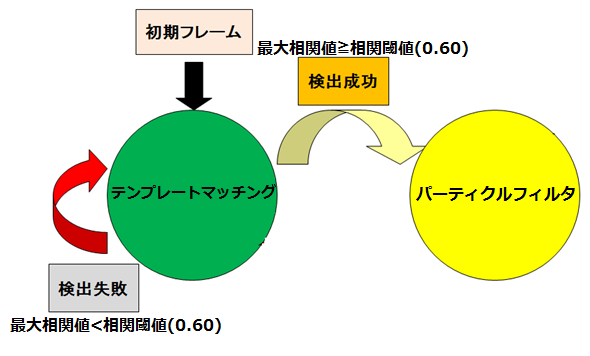


図5.2　テンプレートマッチングからパーティクルフィルタへの切り替え概略図

５.２　パーティクルフィルタからテンプレートマッチング

パーティクルフィルタで，連続して 3 フレーム以上最大相関値が相関閾値 0.60 より低い場合，図5.3 よりボールを見失った（図5.3 の追跡失敗）ものと判定し，次のフレームからは，テンプレートマッチングに切り替えて画面全体を探索し，ボール検出を試みる。

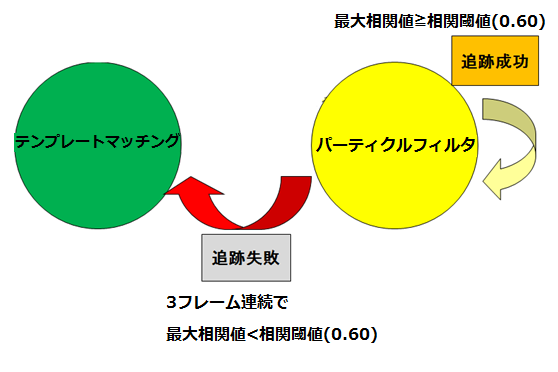


図5.3　パーティクルフィルタからテンプレートマッチングへの切り替え概略図

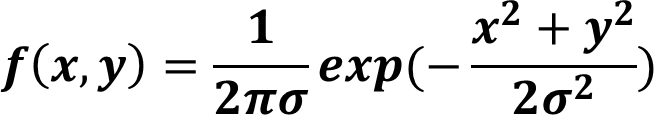
一方，最大相関値が相関閾値より低い場合でも，連続で3 フレーム未満のときは，次のフレームでもパーティクルフィルタのままでボール追跡を行う。パーティクルフィルタにおいて，最大相関値が相関閾値より高い場合は，次のフレームでも引き続きパーティクルフィルタを用いて，ボールの追跡を行う。

**６ 平滑化**

平滑化（スムージング，smoothing）は，画像上の濃淡変動を滑らかにする処理のことである。雑音の低減を図る場合や画像をぼかす効果を得たいときに使用される。

6.1 ガウシアンフィルタ

加重平均フィルタリングとも呼ばれている。注目する中心画素に対して周辺の画素の寄与は小さいと仮定したフィルタ処理である。平滑化の程度を重み係数を変えることによって制御できる。緩やかな平滑化を行うことが可能である。ガウス分布をもつことが知られており，そのため正規分布型の重み係数が使用されることが多い。一般的な画像では注目画素から近い部分の輝度値は注目画素との差が小さく，逆に注目画素から遠い部分は注目画素の輝度値との差が大きくなる場合が多い。この事を考慮し，ガウシアンフィルタでは注目画素に近い部分ほど重み付けを大きくし，逆に注目画素から遠い部分の輝度値ほど重み付けを小さくするガウス分布関数を用いている。



ガウス分布関数で使われている標準偏差σの値が小さいほど平滑化の効果は小さくなり，大きいほど平滑化の効果は大きくなる。ガウス分布関数を３×３のフィルタの重み付けに用いると図6.1となる。画像の輪郭部分がぼやけてしまう欠点がある。

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

図6.1　３×３の場合のガウシアンフィルタの係数

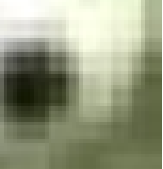
****

図6.2　原画像の一部拡大画像

原画像 図6.2 に対して，フィルタサイズ 3×3 のガウシアンフィルタを適用した例を以下の図6.3 に示す。

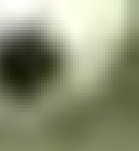
****

図6.3　ガウシアンフィルタを適用した画像

原画像 図6.2 に比べてガウシアンフィルタを適用した図6.3 画像はサッカーボールの輪郭部分がぼやけていることがわかる。

6.2 メディアンフィルタ

メディアンフィルタは，注目画素の輝度値と周辺の輝度値を大きさ順に並べ，メディアン(中央値)を注目画素に置き換えてノイズを除去する。特にスパイク状の雑音（ごま塩雑音とも呼ばれる）を容易に取り除くことが可能である。

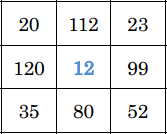


図6.4　３×３の場合における局所領域の濃度値（メディアンフィルタ例）

中央の注目画素 12(値)と周辺画素の輝度値を取得する。取得した輝度値を順番(昇順)に並び替える。

{12 20 23 35 52 80 99 112 120}

並び変えた輝度値の中から中央値 52(値)を取得し，元の注目画素の輝度値 12 と置き換える

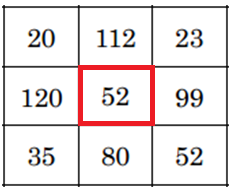


図6.5　置き換えたフィルタ画像の画素の輝度値

例として，原画像 図6.6 に対して，フィルタサイズ ３×３ のメディアンフィルタを適用した例を以下の図6.7 示す。

メディアンフィルタを適用した画像は，原画像図6.6 に比べて画像の輪郭部分がぼやけていることがわかる。

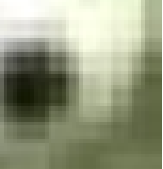


図6.6　原画像の一部拡大画像

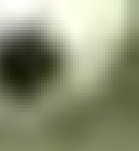
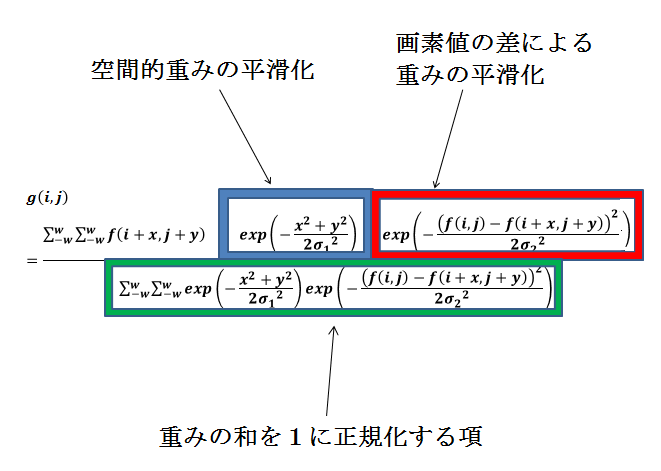


図6.7　メディアンフィルタを適用した画像

6.3 輪郭保存フィルタ

ガウシアンフィルタを用いると，画像の輪郭部分がぼやけてしまう欠点があった。それに対して，輪郭保存フィルタは画像を平滑化しながら画像の輪郭部分をあまりぼやけさせないようにしたフィルタのことである。その方法は注目画素に近い部分ほど重み付けを大きくし，逆に注目画素から遠い部分の輝度値ほど重み付けを小さくするようにしたガウス分布関数と併せて注目画素からの輝度値の差に基づいた平滑化をする。つまり，輝度値が似ていれば似ているほど，似ていない輝度値より大きく重み付けをするガウシアンフィルタである。以下に輪郭保存フィルタの式を示す。

輪郭保存フィルタの式で使われているw はフィルタサイズ，は注目画素における標準偏差， は輝度値輝度差を制御する標準偏差である。の値が小さいほど平滑化の効果は小さくなり、大きいほど効果が大きくなる。 の値はフィルタの注目画素の輝度値との差に関する基準値を表している。そのため， の値を大きくするほど輝度値の大きく異なることを許容することになるから，ガウシアンフィルタに近づき，画像の輪郭部分もぼやけてしまう。



ここでは処理前の画像データの配列を𝒇(𝒊,𝒋) と表し、処理後の画像データの配列を として表す。例として，原画像図6.8 にフィルタサイズ 3×3 の輪郭保存フィルタを適用した例を示す(図6.9を参照)。輪郭保存フィルタを適用した画像は，ガウシアンフィルタを適用した画像に比べ，輝度値の差による重み付けを加えたので画像の輪郭部分があまりぼやけずに強調された。



図6.8　原画像の一部拡大画像



図6.9　原画像に輪郭保存フィルタを適用した画像の一部拡大した画像

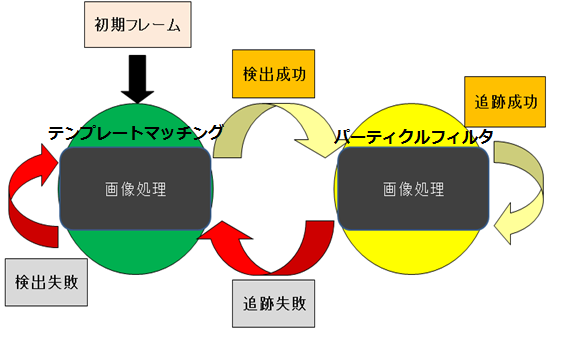


図6.10　本実験の概要図

図5.3で紹介したように研究ではテンプレートマッチングとパーティクルフィルタの切り替え追跡手法を用いる。さらに本研究では上記で紹介したバイリニア法と画像平滑化（ガウシアンフィルタ、メディアンフィルタ、輪郭保存フィルタ）を切り替え追跡手法に取り入れる（図6.10参照）。初期フレームではテンプレートマッチングを用いてテンプレート画像と合致する領域を探索画像から見つけるわけだが、用意したテンプレート画像に画像処理（バイリニア法と画像平滑化）を加える。その後、テンプレート画像と合致するボール領域を中心に粒子を撒く。そして，次フレームからパーティクルフィルタに切り替えてボールの追跡を行う。パーティクルフィルタにおいて3 フレーム連続して低い評価値が続いた場合，ボールを見失ったものと判断し，次のフレームではテンプレートマッチングに切り替え物体の検出から始める。

**７ 結果**

本研究では表7.1に示す7つの動画像（data 1,…,data 7）を用意した。テンプレートマッチング単体による手法，テンプレートマッチングとパーティクルフィルタの切り替えによる手法（ここでは切り替え追跡手法と表す），ガウシアンフィルタを用いた切り替え追跡手法、メディアンフィルタを用いた切り替え追跡手法、輪郭保存フィルタを用いた切り替え追跡手法でボール追跡の実験を行った。それぞれの手法で行ったボール追跡の実験の結果を表2～8に記す。図7.1で各データごとの追跡精度を比較した図を示す。図7.2で各手法ごとの追跡精度を比較した図を示す。なお、表中の追跡精度は以下の式によって求めた。

**追跡精度(%) =**

表7.1　7つの動画像とそのフレーム数

|  |  |
| --- | --- |
| data1 | 271フレーム |
| data2 | 243フレーム |
| data3 | 333フレーム |
| data4 | 289フレーム |
| data5 | 214フレーム |
| data6 | 203フレーム |
| data7 | 730フレーム |

表7.2 data1追跡精度比較

|  |  |
| --- | --- |
| 手法 | 追跡精度 (271フレーム) |
| テンプレートマッチングのみ | 64．2% |
| 切り替え追跡手法 | 81．9% |
| ガウシアンフィルタを用いた切り替え追跡手法 | 83．2% |
| メディアンフィルタを用いた切り替え追跡手法 | 82．3% |
| 輪郭保存フィルタを用いた切り替え追跡手法 | 85．1% |

表7.3 data2追跡精度比較

|  |  |
| --- | --- |
| 手法 | 追跡精度 (243フレーム) |
| テンプレートマッチングのみ | 66．3% |
| 切り替え追跡手法 | 83．1% |
| ガウシアンフィルタを用いた切り替え追跡手法 | 84．0% |
| メディアンフィルタを用いた切り替え追跡手法 | 84．0% |
| 輪郭保存フィルタを用いた切り替え追跡手法 | 86．7% |

表7.4 data3追跡精度比較

|  |  |
| --- | --- |
| 手法 | 追跡精度 (333フレーム) |
| テンプレートマッチングのみ | 57．4% |
| 切り替え追跡手法 | 75．4% |
| ガウシアンフィルタを用いた切り替え追跡手法 | 76．9% |
| メディアンフィルタを用いた切り替え追跡手法 | 76．9% |
| 輪郭保存フィルタを用いた切り替え追跡手法 | 77．7% |

表7.5 data4追跡精度比較

|  |  |
| --- | --- |
| 手法 | 追跡精度 (289フレーム) |
| テンプレートマッチングのみ | 60．9% |
| 切り替え追跡手法 | 81．7% |
| ガウシアンフィルタを用いた切り替え追跡手法 | 82．0% |
| メディアンフィルタを用いた切り替え追跡手法 | 82．0% |
| 輪郭保存フィルタを用いた切り替え追跡手法 | 84．7% |

表7.6 data5追跡精度比較

|  |  |
| --- | --- |
| 手法 | 追跡精度 (214フレーム) |
| テンプレートマッチングのみ | 70．1% |
| 切り替え追跡手法 | 80．4% |
| ガウシアンフィルタを用いた切り替え追跡手法 | 80．4% |
| メディアンフィルタを用いた切り替え追跡手法 | 80．4% |
| 輪郭保存フィルタを用いた切り替え追跡手法 | 80．4% |

表7.7 data6追跡精度比較

|  |  |
| --- | --- |
| 手法 | 追跡精度 (203フレーム) |
| テンプレートマッチングのみ | 69．0% |
| 切り替え追跡手法 | 82．8% |
| ガウシアンフィルタを用いた切り替え追跡手法 | 82．8% |
| メディアンフィルタを用いた切り替え追跡手法 | 82．8% |
| 輪郭保存フィルタを用いた切り替え追跡手法 | 82．8% |

表7.8 data7追跡精度比較

|  |  |
| --- | --- |
| 手法 | 追跡精度 (730フレーム) |
| テンプレートマッチングのみ | 67．3% |
| 切り替え追跡手法 | 84．0% |
| ガウシアンフィルタを用いた切り替え追跡手法 | 86．4% |
| メディアンフィルタを用いた切り替え追跡手法 | 86．4% |
| 輪郭保存フィルタを用いた切り替え追跡手法 | 87．7% |

図7.1　各手法の追跡精度比較図1

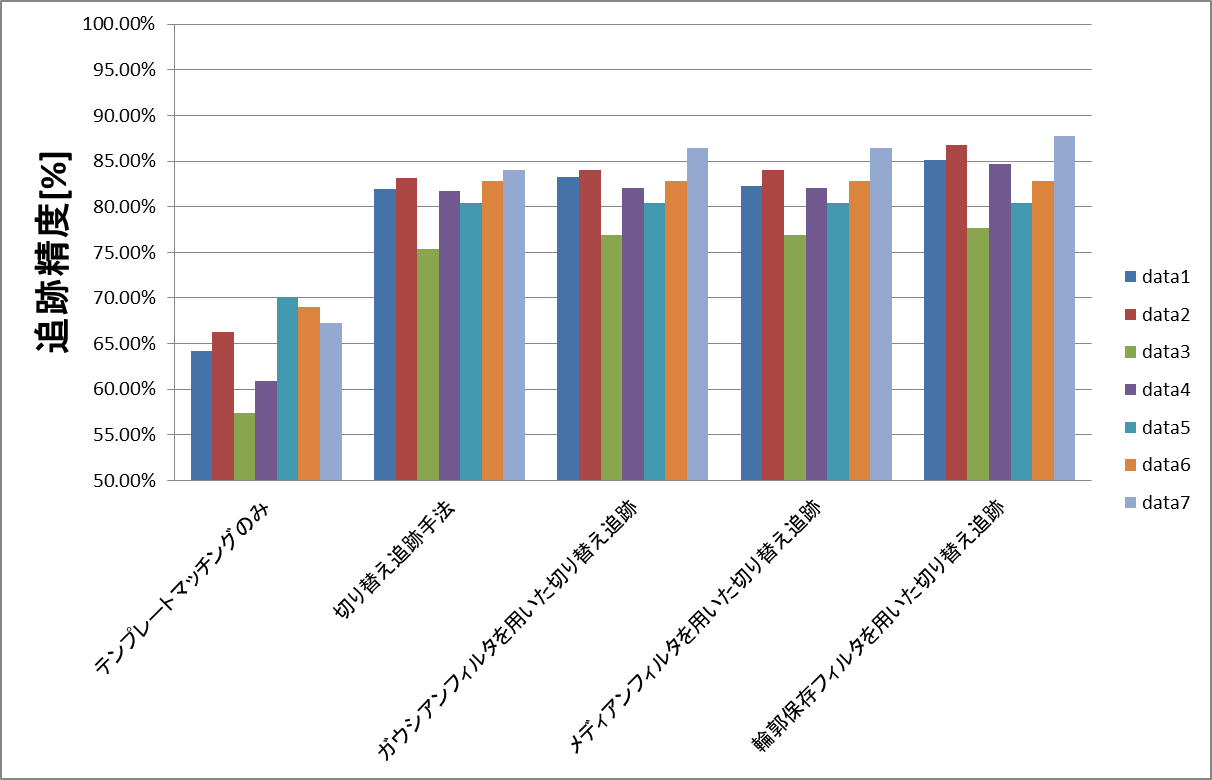


図7.2　各手法の追跡精度比較図2

テンプレートマッチング単体と切り替え追跡手法でのボール追跡手法で実験を行った。実験の結果は，テンプレートマッチング単体では表7.2～7.8 及び 図7.1 及び図7.2 からも分かるように，ボールの追跡精度は低い。表7.9 より切り替え追跡手法はテンプレートマッチング単体より，高い精度でのボールの追跡が実現できたことがわかる。

表7.9　各dataごとのテンプレートマッチング単体における追跡と切り替え追跡における追跡精度比較

|  |  |
| --- | --- |
| data1 | 17.7%の精度向上 |
| data2 | 16.8%の精度向上 |
| data3 | 18.0%の精度向上 |
| data4 | 20.8%の精度向上 |
| data5 | 10.3%の精度向上 |
| data6 | 13.8%の精度向上 |
| data7 | 16.7%の精度向上 |

画像の平滑化処理を行った追跡に関しては多くのdataで追跡精度の向上を確認することができた。ガウシアンフィルタを用いた場合とメディアンフィルタを用いた際の結果は数値としてはあまり変化が見られなかった。輪郭保存フィルタでは表7.10 よりガウシアンフィルタを用いた場合とメディアンフィルタを用いた場合よりも若干ながら精度向上が見て取れた。

表7.10　各dataごとの平滑化処理なしの切り替え追跡と各平滑化処理を施した際の切り替え追跡精度比較1（+は精度向上、－は精度低下を表す）

|  |  |  |  |
| --- | --- | --- | --- |
| 各データ | ガウシアンフィルタ | メディアンフィルタ | 輪郭保存フィルタ |
| data1 | +1.3% | +0.4% | +3.2% |
| data2 | +0.9% | +0.9% | +2.0% |
| data3 | +1.5% | +1.5% | +2.3% |
| data4 | +0.3% | +0.3% | +3.0% |
| data5 | 0.0% | 0.0% | 0.0% |
| data6 | 0.0% | 0.0% | 0.0% |
| data7 | +2.4% | +2.4% | +3.7% |

また平滑化処理なしの切り替え追跡において検出・追跡が成功した場合と失敗した場合の各平滑化手法を用いた切り替え追跡の追跡精度を相対比較した（表7.11参照）。「正」は平滑化処理なしの切り替え追跡において検出・追跡が成功した際における各平滑化処理を行った場合の検出・追跡の相対比較、「誤」は平滑化処理なしの切り替え追跡において検出・追跡が失敗した際における各平滑化処理を行った場合の検出・追跡の相対比較を示す。

|  |  |
| --- | --- |
| 正 | ±0.0% |

とは平滑化処理なしの切り替え追跡において検出・追跡が成功した際には平滑化処理を用いた切り替え追跡においても検出・追跡が成功したことを表す。

表7.11　各dataごとの平滑化処理なしの切り替え追跡と各平滑化処理を施した際の切り替え追跡精度比較2（+は精度向上、－は精度低下を表す）

|  |  |  |  |
| --- | --- | --- | --- |
| data1 | ガウシアンフィルタ | メディアンフィルタ | 輪郭保存フィルタ |
| 正 | ±0.0% | ±0.0% | ±0.0% |
| 誤 | +1.3% | +0.4% | +3.2% |

|  |  |  |  |
| --- | --- | --- | --- |
| data2 | ガウシアンフィルタ | メディアンフィルタ | 輪郭保存フィルタ |
| 正 | ±0.0% | ±0.0% | ±0.0% |
| 誤 | +0.9% | +0.9% | +3.6% |

|  |  |  |  |
| --- | --- | --- | --- |
| data3 | ガウシアンフィルタ | メディアンフィルタ | 輪郭保存フィルタ |
| 正 | ±0.0% | ±0.0% | ±0.0% |
| 誤 | +1.3% | +0.4% | +3.2% |

|  |  |  |  |
| --- | --- | --- | --- |
| data4 | ガウシアンフィルタ | メディアンフィルタ | 輪郭保存フィルタ |
| 正 | - 0.01% | ±0.0% | ±0.0% |
| 誤 | +0.9% | +1.3% | +3.0% |

|  |  |  |  |
| --- | --- | --- | --- |
| data5 | ガウシアンフィルタ | メディアンフィルタ | 輪郭保存フィルタ |
| 正 | ±0.0% | ±0.0% | ±0.0% |
| 誤 | ±0.0% | ±0.0% | ±0.0% |

|  |  |  |  |
| --- | --- | --- | --- |
| data6 | ガウシアンフィルタ | メディアンフィルタ | 輪郭保存フィルタ |
| 正 | ±0.0% | ±0.0% | ±0.0% |
| 誤 | ±0.0% | ±0.0% | ±0.0% |

|  |  |  |  |
| --- | --- | --- | --- |
| data7 | ガウシアンフィルタ | メディアンフィルタ | 輪郭保存フィルタ |
| 正 | ±0.0% | ±0.0% | ±0.0% |
| 誤 | +2.4% | +2.4% | +3.7% |

表7.11 よりどのdataにおいても平滑化処理を用いていない切り替え追跡において検出・追跡が成功した際には輪郭保存フィルタにおいても検出・追跡が成功したことがわかる。またほとんどのdataでガウシアンフィルタやメディアンフィルタにおいても輪郭保存フィルタと同様の結果が得られた。

**８ 考察**

表7.2～7.8を見ると全動画像においてテンプレートマッチング単体よりテンプレートマッチングとパーティクルフィルタとの切り替え追跡手法がより精度の高い追跡が可能であることがわかる。テンプレートマッチング単体より追跡精度が上がっているのは，テンプレートマッチング単体では追跡できなかったフレームまで，パーティクルフィルタに切り替えることによって追跡できるためである。パーティクルフィルタを用いることにより検出箇所をボール周辺にできるのでテンプレートマッチングを画面全体に用いるより効率的に追跡が可能になったので精度向上したと考えられる。言うなればボール位置から遠く離れた見当違いの箇所の誤検出を減らせる。

切り替え追跡手法により追跡精度が向上したことに関して考察すると切り替え追跡手法は、テンプレートマッチングの短所である「短時間のオクルージョン」をパーティクルフィルタで補うことが出来、そしてパーティクルフィルタの短所である「一度ボールを見失ってしまったら，再び追跡をすることが困難」といった特徴をテンプレートマッチングで補うことができるので追跡精度が向上したと考えられる。従来研究からもその有効性が確認されていた。実際に本実験からもその有効性を見出すことが出来た。

テンプレートマッチングは、オクルージョンやテンプレート画像との差が出てしまうと対応することが困難となる。(図8.1参照)



図8.1　テンプレートマッチングによる誤検出

しかし、パーティクルフィルタでは、粒子が等速直線運動に基づき、追跡対象物体の動きを予測しながら追跡を行なうので、対応することができた。(図8.2参照)



図8.2　パーティクルフィルタでの追跡

なぜ、同じテンプレートマッチングを用いているパーティクルフィルタでは追跡することが可能であったのか。テンプレートマッチングは毎フレーム常に全画面探索を行なうが、パーティクルフィルタは粒子がサッカーボールの状態方程式に基づいて変位をし、それぞれの粒子を中心に探索画像を一定の大きさで切り取り、その中でテンプレートマッチングを行なう。このため、テンプレートマッチング単体での探索よりも探索範囲がある程度限られているという違いがあるためであると考えられる。よって、切り替え追跡手法は、テンプレートマッチングとパーティクルフィルタそれぞれの特徴を活かし、効率的に追跡が行なわれていることがわかった。

画像平滑化についてみてみると画像平滑化を行っていないテンプレート画像を用いた場合より画像平滑化を行ったテンプレート画像を用いた方が追跡精度の向上が見られた。これは従来の研究では生じてしまった芝や選手のユニフォームの誤検出を減らせたことが要因として挙げられる。ガウシアンフィルタを用いた場合とメディアンフィルタを用いた際の結果は数字としてはあまり変化が見られなかった。これはアルゴリズムに違いはあれど両フィルタともに画像内のノイズを減らすことができたためだと考えられる。ノイズを減らせたことでノイズに左右される頻度が減った。輪郭保存フィルタではガウシアンフィルタのノイズの減少に加え、ボール部分を強調することで芝やユニフォームとの差別化が図れたと考えられる。

また平滑化処理を用いていない切り替え追跡では検出・追跡に成功したが平滑化を用いることで検出・追跡に失敗するフレームが存在した。これは平滑化処理におけるノイズ処理においてボール部分がぼやけてしまったため誤検出を招いてしまったと考えられる。

　以上のように、本研究では画像平滑化の有効性を確認することができたが、追跡の際の課題とされる図8.3のような長時間のオクルージョン、例えばボールが選手に長時間隠れている時、ゴールキーパーがボールを抱えている時などに関しては、切り替えの条件の変更を行なっても改善が見られずいまだに解決できなかった。



図8.3　オクルージョン例

この問題の解決策としては、隠してしまっている選手を追跡することによってボールの位置を推定できるようになるだろうと考える。各追跡手法、各処理のパラメータ、例えば相関閾値やパーティクルフィルタからテンプレートマッチングに切り替える際のフレーム数の設定の自動化なども今後の課題として挙げることができる。

謝辞

本研究を進めるにあたり、貴重なご意見、ご指導を頂いた、応用情報工学科平原先生に深く感謝致します。また，本研究に対し，御協力頂いた法政大学院応用情報工学専攻の中山先輩に深く感謝致します。

参考文献

[1]北川源四郎,竹村彰通, 21世紀の統計科学Ⅲ：数理・計算の統計科学, 東京大学出版会, pp.305-338, 2008.

[2]奈良先端科学技術大学院大学OpenCVプログラミングブック制作チーム著, 奈良先端科学技術大学院大学OpenCVプログラミングブック制作チーム, 毎日コミュニケーションズ,

2009.

付録

・プログラム実行方法

環境

Microsoft Visual C++ 2008 Express Edition

画像、映像の再生、処理にはintelがオープンソースで公開しているコ 　　　 ンピュータビジョン関連のライブラリである、OpenCVを使用している

設定方法

Visual C++でWin32 Contsole Applicationとしてプロジェクトを作成し、 プログラムをプロジェクトへ追加すればよい。ビルドするときには、「ツ ール」メニューの「プロジェクト」からプロパティを選択する。「構成」 で「すべての構成」を選択し、左側のツリーから「C/C++」タブを選択し、 「追加のインクルードディレクトリ」にインストールしたOpenCVのホ

ームディレクトリの下の以下のディレクトリを追加する。

cxcore╲include, cv╲include, otherlibs╲highgui, cvaux╲include

次に、「リンカ」タブを選択し、「追加のライブラリディレクトリ」に

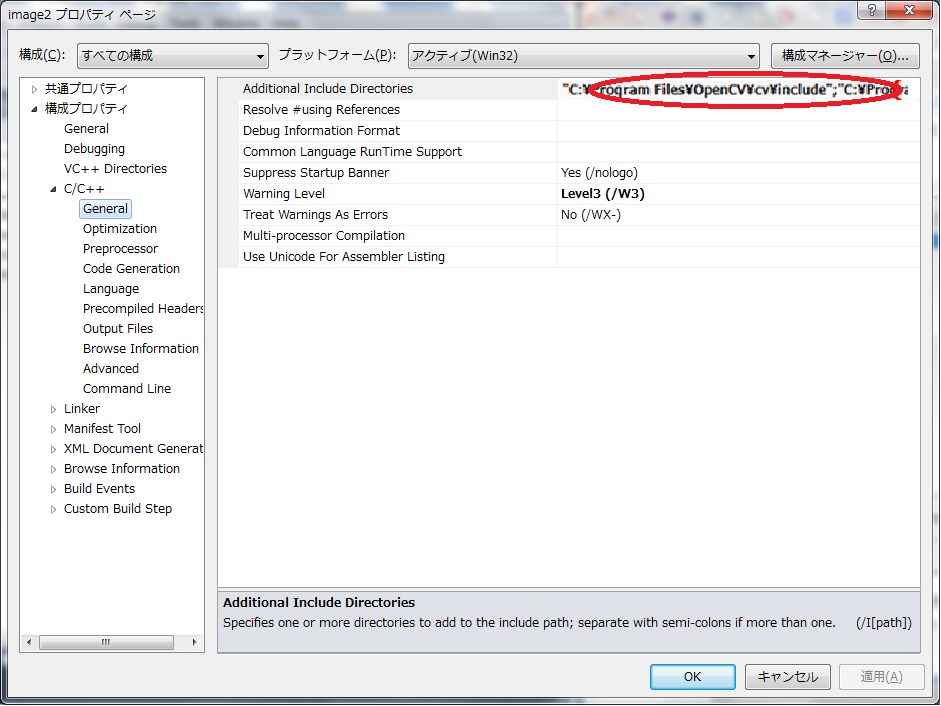
OpenCV ディレクトリ下の”lib”ディレクトリを追加する。

続いて、「構成」を「Debug」に変更し、「リンカ」タブを展開した中の「入 力」を選択し、「追加の依存ファイル」で”cxcore.lib highgui.lib cvaux.lib”を追加する。

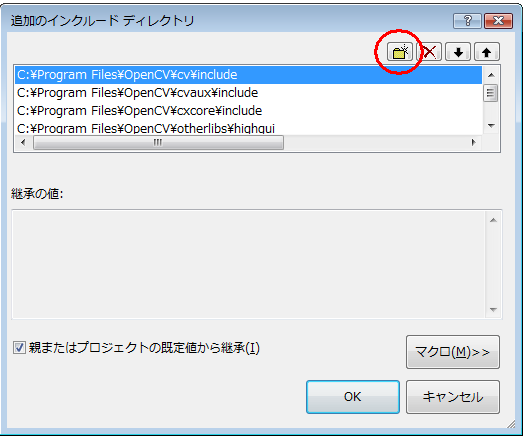
OpenCV導入

OpenCVライブラリを使うには、ダイナミックライブラリファイル（\*.dll）、ライブラリファイル（\*.lib）、インクルードファイル（ヘッダファイル）（\*.h）が必要になり、これらを使えるようにVisualStudioの設定をする必要がある。

インクルードファイル(\*.h)の設定は、メニューのプロジェクト→【プロジェクト名】のプロパティで設定ウィンドウを開く。

****

ここで、構成はすべての構成を選択し、構成のプロパティ→C/C++→全般を選択し、追加のインクルードファイルの右の方にある参照ボタンをクリックし、インクルードファイル(\*.h)のあるディレクトリを指定。



デフォルトのインストールではディレクトリは以下を追加

C:\Program Files\OpenCV\cv\include

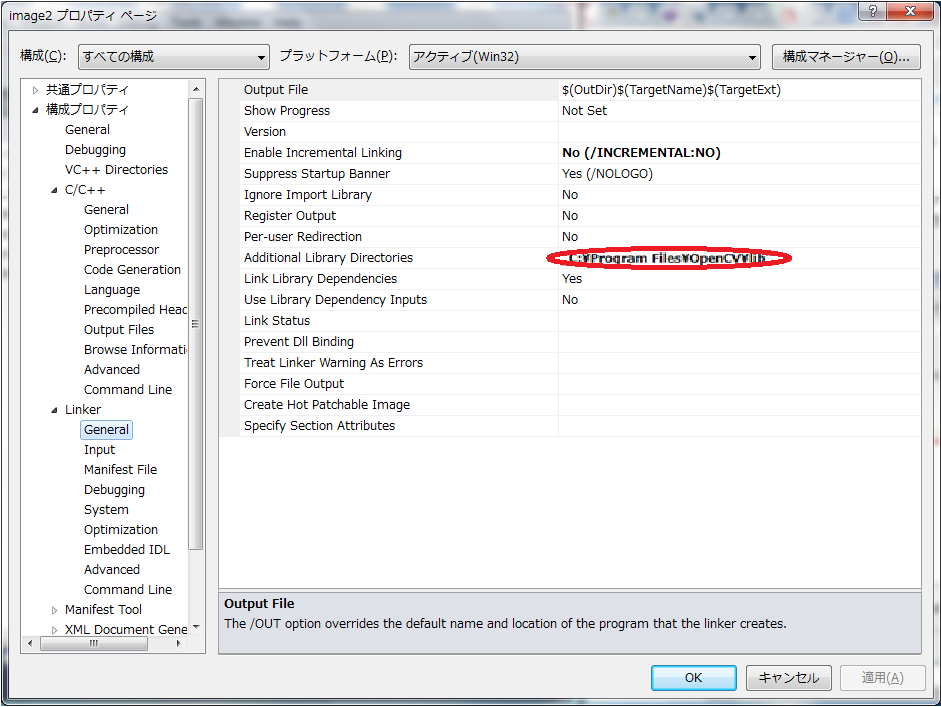
C:\Program Files\OpenCV\cvaux\include

C:\Program Files\OpenCV\cxcore\include

C:\Program Files\OpenCV\otherlibs\highgui

次にライブラリファイル（\*.lib）の設定を行う。

インクルードファイルの設定と同じように、メニューのプロジェクト→【プロジェクト名】のプロパティを選択し、今度は構成プロパティ→リンカ→全般で追加ライブラリディレクトリを選択。

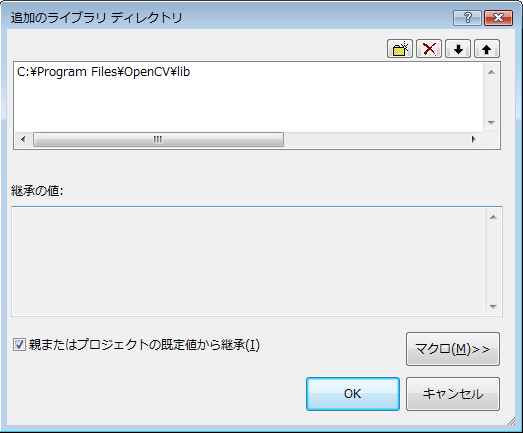


右側の参照ボタンをクリックし、ライブラリファイル(\*.lib)のあるディレクトリを指定

デフォルトのインストールでは

C:\Program Files\OpenCV\lib

を追加。

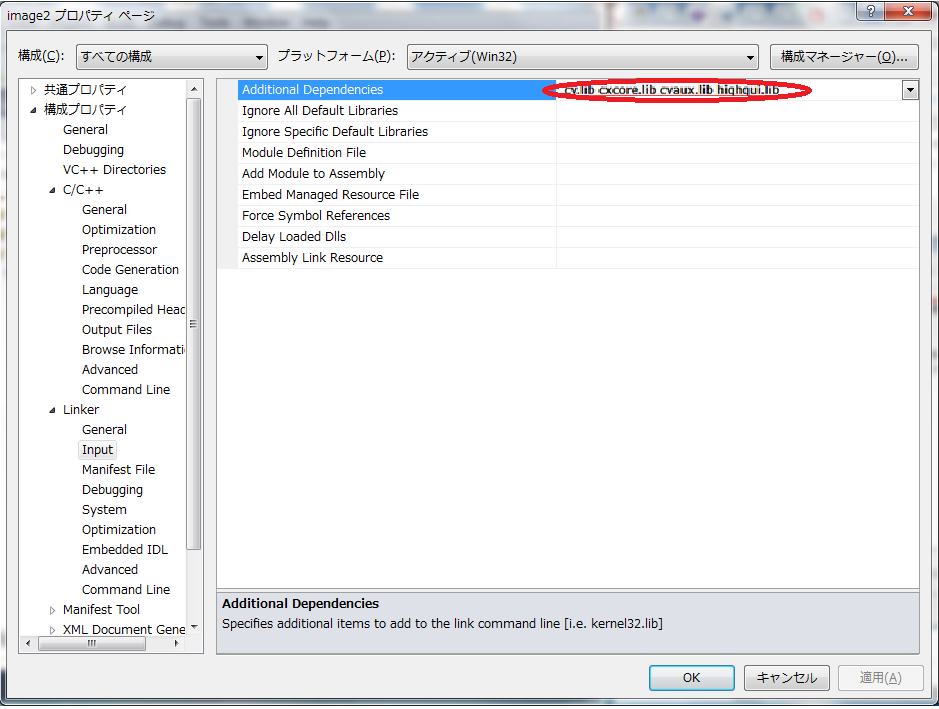


次に実際に使うライブラリファイル(\*.lib)ファイルの設定を行う。

この設定はメニューのプロジェクト→【プロジェクト名】のプロパティを選択し、構成プロパティ→リンカ→入力で追加の依存ファイルを追加。

以下の４つを追加しておく

cv.lib cxcore.lib cvaux.lib highgui.lib



これで、ダイナミックライブラリファイル（\*.dll）、ライブラリファイル（\*.lib）、インクルードファイル（ヘッダファイル）（\*.h）の参照設定は完了。

参照設定したライブラリはおおまかにいうと以下のように分類されている。

cv ：画像処理全般

cxcore ：行列などの数値演算

cvaux ：ステレオマッチングなどの高機能処理

highgui ：ファイル入出力

付録

　本研究を1つのフォルダにまとめる。

・卒論中間発表概要　（卒論中間報告\_11X3014\_井垣円.docx）

・卒論中間発表プレゼン　（卒業研究中間発表\_11X3014\_井垣円.pptx）

・卒論最終発表概要　（卒業研究最終報告\_11X3014\_井垣円.docx）

・卒論最終発表プレゼン　（卒業研究発表\_11X3014\_井垣円.pptx）

・卒業論文　（卒業論文\_11X3014\_井垣円.docx）

・プログラムファイル　（卒業研究プログラム\_11X3014\_井垣円）

・実験結果　（卒業研究実験結果\_11X3014\_井垣円.xlsx）