2004年度 修士論文

TVサッカー映像の自動要約
自動IndexingとIndexの重み生成

Auto-Making Soccer Video Digests

提出日:2005年2月2日

指導教授

白井克彦 教授

早稲田大学大学院 理工学研究科 情報・ネットワーク専攻

3603U043-0

川口 克則

Katsunori Kawaguchi

目 次

図 目 次

表 目 次

第1章　序論

1.1　研究背景

近年，インターネットにおけるマルチメディアコンテンツの急増，CS・BS 等の TV 放送局の増加，家庭用・業務用ビデオカメラの発達等によって，身近に存在する映像の量が増加している．また，今後の映像の蓄積速度はさらに上昇することが予想される．

しかし，これら全ての映像を一人の人間が見ることは時間的に不可能であり，興味がある全ての映像だけを見ることも困難である．そこで，映像の要約を自動的に生成し，元映像の代用とすることが必要となる．要約映像は，時間が短く，情報量が元映像に近いものであることが望まれ，場合によっては，見る人間の嗜好（プロファイル）を考慮し，演出効果を施す必要がある．

1.2　研究の目的

映像の要約を生成するためには，映像の意味を理解し，映像に情報を付加する作業（Indexing）を行う必要がある．それらの先行研究として，映像の断片化を行ったもの [1]，スポーツ画像のスコアブック作成を自動化したもの [4]，カット構成やカメラワークの規則性によって映像を分類したもの [5] [6]，等が挙げられる．要約作成手法として，スポーツ画像に関するもの [7]，脳波によるもの [8] 等が挙げられる．

本研究チームでは，TV サッカー映像の自動要約システム作成を最終目標としている．まず，全ての動画像に対する要約生成モデルを検討し，サッカー映像の要約として，テレビで放送されるダイジェスト等に代表される一般的な要約と，個人の趣向に応じた要約の 2 つを設定した．システムの特徴としては，入力を TV サッカー映像のみとしているため，導入のコストが非常に少ないこと，TV 放送のみが残っている過去の放送にも対応できることが挙げられる．本稿では，要約システム生成の要素技術として，画像処理に

よるシュートやパス等のサッカーイベントの Indexing 手法，サッカーイベントの要約への影響度（イベント重み）の算出手法の確立を目指した．

1.3　論文の構成

本論文は全 6 章からなる．

2 章では，以降の章で用いられる画像処理技術について述べる．

3 章では，本研究が最終的な目的とする映像の要約手法と，サッカー映像に対する要約手法を提案し，サッカー映像の Index について述べる．

4 章では，本研究で行ったサッカー映像に対する画像処理の内容を述べる．

5 章では，前章で行った結果についての評価を行う．

6 章では，本論分のまとめを行い，今後の課題について述べる．

第2章　画像処理基礎技術

本章では，以後の章で用いる基本的な画像処理技術についての解説を行う．

2.1　画像データ

2.1.1　画像の表現

画像とは水平及び垂直に設定された2つ座標を x,y により表現される 2 次元の情報であるとする．この2つの変数で示される位置における輝度 (brightness) あるいは濃度値 (gray level value) を次式のように関数で記述する．

$$f(x, y) = g$$

2.1.2　ディジタル画像

2つの座標軸 x,y および濃度値 g の連続値で与えられるが画像をアナログ画像という．それに対して，x,y 座標軸をある周期 T で基盤の目状に区切り，各交点における離散的な位置における濃度だけを対象とした画像を標本化画像という．また，画像の濃度値を離散的な濃度値で表現したものを量子化画像という．一般的にディジタル計算機で画像を取り扱う場合，すべて離散的な情報として処理する必要があるため，画像を標本化し，かつ量子化しなければならない．このような画像をディジタル画像という．図 2.1 にディジタル濃淡画像の行列表現の例を示す．

2.1.3　ラスタ走査

画像は2つの変数 x,y により記述される2次元情報となっている．この2次元画像情報を，距離の離れた場所に伝送するためには1次元の画像信号に変換する必要が生じる．ま

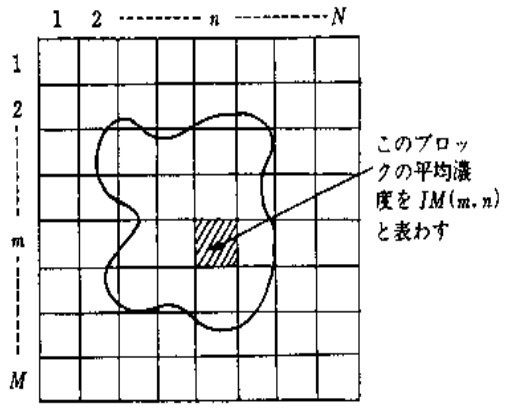


図 2.1: ディジタル濃淡画像の行列表現の例

た，電子計算機内で画像を処理し蓄積する際にも，1 次元に変換することにより取り扱いが容易になる．2 次元画像情報を 1 次元に変換する方式としては，各種の方式が考えられるが，図 2.2 に示すように，画像の左上を始点として，最上行から順次下位行の画像の濃度値を 1 次元配列にさせて，1 次元画像信号を作成する方式が広く利用されている．この 1 本の水平方向への画像変換操作を水平走査とよび，水平走査群により構成された画像をラスタ (raster)，さらにこのような走査により変換された画像をラスタ走査画像またはラスタスキャン画像 (raster scan image) とよぶ.

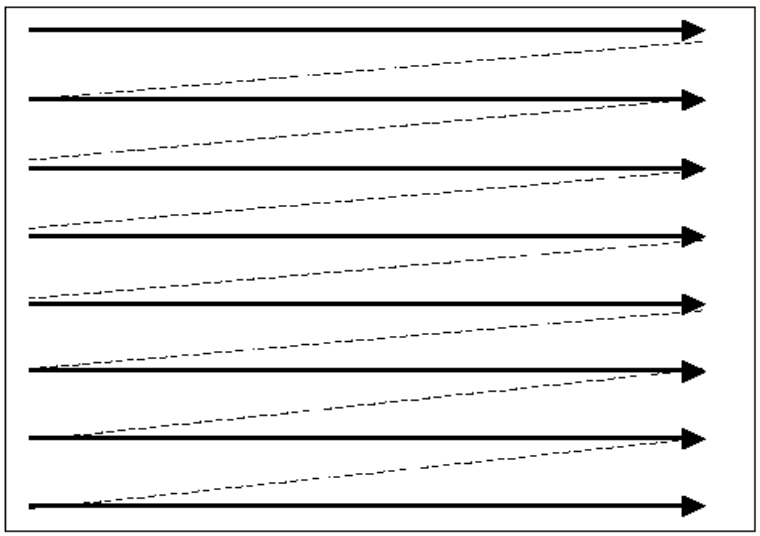


図 2.2: ラスタ方向の走査

2.2　カラー画像

色を表現する方式を表色系 (color model,color base) と言う．その内代表的なものについて説明する．

2.2.1　RGB 表色系

RGB とは赤 (Red)，緑 (Green)，青 (Blue) の加算混合 (additive mixing) の 3 原色で色を決める方法である．計算機上でのカラー画像はこの形式で扱われ，一般には数値として，R，G，B，それぞれ 0〜255 の値 (8 ビット) をとり，計 24 ビットの値で表す（図 2.3 参照）．

2.2.2　CMY 表色系

CYM とは RGB の捕色である水色 (cyan)，紫 (magenta)，黄 (yellow) の減算混合 (subtractive mixing) の 3 原色で色を決める方法である．主に印刷関係で使用され，それらの用途では黒 (black) を加えた CMYK という表現を用いる（図 2.4 参照）．

RGB から CMY への変換式は次のようになる．

$$C = 255 - R, \quad M = 255 - G, \quad Y = 255 - B$$

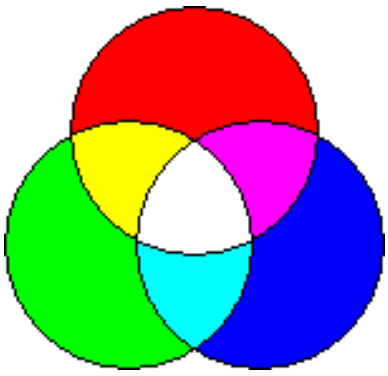


図 2.3: RGB 表色系

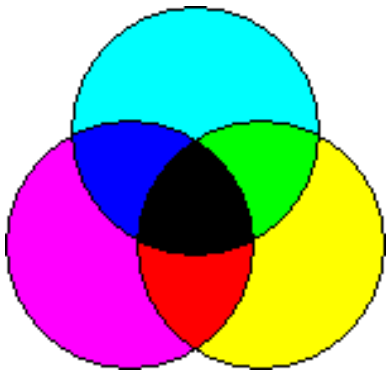


図 2.4: CYM 表色系

2.2.3　HSV 表色系

HSV とは色相角度 (Hue angle)，彩度 (Saturation)，強度 (Value) という HSV 六角形で色を決める方法である．

H は色相角度であり，$0 \sim 2\pi$ の値で指定する．赤が 0 で黄が $\pi/3$ ，緑が $2\pi/3$ で水色が π ，青が $4\pi/3$ で紫が $5\pi/3$ ，そして 2π で再び赤に戻る（図 2.5 参照）．

S は彩度であり，低いほど無彩色で，高いほど有彩色となる．

V は色の強度であり，高いほど強度は強くなる．

強度 (Value) の代わりとして，明度 (Lightness) を用いた HSL 表色系や HLS 表色系も存在する．

RGB から HSV への変換式は次のようになる．

$$V = \max(R, G, B)$$

$$S = \begin{cases} 0 & (\max(R,G,B) = 0 \text{ のとき}) \\ 1 - \min(R,G,B)/\max(R,G,B) & (\text{それ以外}) \end{cases}$$

$$H = \begin{cases} 0 & (S = 0 \text{ のとき}) \\ (G-B)/(\max(R,G,B) - \min(R,G,B)) & (\max(R,G,B) = R \text{ のとき}) \\ (B-R)/(\max(R,G,B) - \min(R,G,B)) & (\max(R,G,B) = G \text{ のとき}) \\ (R-G)/(\max(R,G,B) - \min(R,G,B)) & (\max(R,G,B) = B \text{ のとき}) \end{cases}$$

図 2.5: 色相環

2.2.4　YCC 表色系

YCC とは NTSC(National Television System Committee) 方式のカラーテレビ放送で使われる表色系であり，白黒テレビとカラーテレビの互換性を保つために，輝度 (Y,CIE 表色系での Y 軸であることに由来)，青色の色差 (Cb)，赤色の色差 (Cr) という，輝度情報とカラー情報を分離して色を決める方法である．

YCbCr 表色系とも言う．

同じような表色系として，YUV 表色系（アジアやヨーロッパのテレビ信号規格）や YIQ 表色系（北米でのテレビ信号規格）がある．

RGB から YCC への変換式は次のようになる．

$$\begin{aligned} Y &= 0.29891 * r + 0.58661 * g + 0.11448 * b \\ Cb &= -0.16874 * r - 0.33126 * g + 0.50000 * b \\ Cr &= 0.50000 * r - 0.41869 * g - 0.08131 * b \end{aligned}$$

2.3　画像の表示

ディジタル画像においては，画像内の各画素の濃度値そのものが画像のもっとも重要な情報を担っている．そこで，的確に画像を表示するための主な濃度変換手法について示す．

2.3.1　濃度変換

処理対象とする画像がどのような濃度で分布しているかを調べることは，画像の前処理として重要な作業である．$M \times N$ の画像領域の対象画像の全画素の濃度分布を濃度の頻度で示したグラフを濃度ヒストグラム (density histgram) という．(図 2.6 , 2.7 参照）濃度ヒストグラムを用いると画像全体の濃度の分布が容易に把握できるようになる．

図 2.6: 元画像

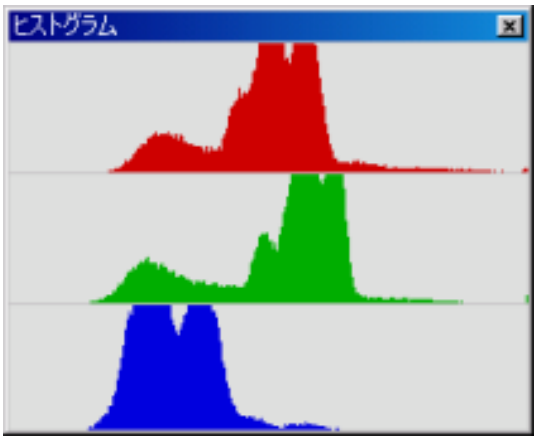


図 2.7: RGB ヒストグラム

2.3.2　2 値化処理

階調画像に対して領域の画素数が一定であれば，量子化数が 1 ビット，すなわち 1 と 0 の 2 値による表現が，もっともデータ量を少なくした状態である．このように，0 と 1 の 2 値により表現された画像を 2 値化画像 (binary image) という．適切な 2 値化により，対象画像の性質や特徴を保存させることが可能になり，2 値画像を処理対象とすることにより，処理に要する CPU 時間の大幅な減少ならびに記憶容量の大幅な低減が可能になる．

2.3.3　閾値処理

階調のある濃淡画像から 2 値画像を得るための 2 値化の方法としては様々な方法があるが，もっとも簡単な方法は，濃度情報を直接用いてあるレベルで区切り，そのレベルより明るい部分を ”0 ”，暗い部分に ”1 ”を割り当て 2 値化を行う方法である．このようにあるレベルで分割する処理を閾値処理と呼ぶ．閾値処理においては，閾値の設定が重要であり，閾値の設定に濃度ヒストグラムが使用される．

2.3.4　アフィン変換

ユークリッド幾何学的な線型変換と平行移動の組み合わせによる図形や形状の移動，変形方式のことをアフィン変換 (Affine transformation) と呼ぶ．アフィン変換には，元画像

の幾何学的性質を保存するという特徴がある．

2 次元のアフィン変換

2 次元のアフィン変換とは，アフィン変換を 2 次元空間の中で行ったもので，これにより画像の回転・拡大縮小・平行移動を表すことができる．

点 (x,y) を点 (X,Y) に変換する 2 次元のアフィン変換は次のような行列式で表すことができ，展開してまとめると，2 次元のアフィン変換の一般式を得る．

2 次元のアフィン変換を用いるには，変換後の座標の判明した 3 点が必要になる．

$$\begin{bmatrix} X & Y & W \end{bmatrix} = \begin{bmatrix} x & y & 1 \end{bmatrix} \begin{bmatrix} a & d & 0 \\ b & e & 0 \\ c & f & 0 \end{bmatrix}$$

$$X = ax + by + c, \quad Y = dx + ey + f$$

3 次元のアフィン変換

3 次元のアフィン変換とは，2 次元のアフィン変換を 3 次元空間に拡張したもので，3 次元空間での幾何学変換を行うことができる．

点 (x,y,z) を点 (X,Y,Z) に変換する 3 次元のアフィン変換は次のような行列式で表すことができる．

$$\begin{bmatrix} X & Y & Z & W \end{bmatrix} = \begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} a & e & i & 0 \\ b & f & j & 0 \\ c & g & k & 0 \\ d & h & l & 0 \end{bmatrix}$$

透視変換

透視変換とは，アフィン変換の一種で，3 次元空間内に置かれた 3 次元図形を任意の視点から眺めて 2 次元平面に投影する変換である．透視変換を用いることによって，2 次元画像に遠近感を持たせることが可能になる．(図 2.8 参照)

点 (x,y,(0)) を点 (X,Y,Z) に変換する透視変換は 3 次元のアフィン変換の行列式の変形によって得ることができる．その後，得られた座標を Z で割ることによって平面上の座標 (X',Y') に展開する．行列式及び一般式を次に示す．

透視変換を用いるには，変換後の座標の判明した 4 点が必要になる．

$$\begin{bmatrix} X & Y & Z & W \end{bmatrix} = \begin{bmatrix} x & y & 0 & 1 \end{bmatrix} \begin{bmatrix} a & d & p & 0 \\ b & e & q & 0 \\ z_1 & z_2 & z_3 & 0 \\ c & f & r & 0 \end{bmatrix}$$

$$X' = \frac{ax + by + c}{px + qy + r}, \quad Y' = \frac{dx + ey + f}{px + qy + r}$$

図 2.8: 透視変換

2.4　画像の認識

2.4.1　パターン認識

パターン認識システムは図 2.9 に示すように，入力データからそのデータに関する特徴量を抽出し，その特徴をあらかじめ設定した特徴と照合し，合致しているかどうかを判定する，という手順で構成される．

2.4.2　前処理

パターン認識作業の第一歩が入力パターンに対する前処理である．未知の入力パターンは一般に各種の歪みやノイズを含んでいる．従って，特徴抽出を行う前に，これらを除去しておく必要がある．入力パターンの前処理としてパターンの正規化が行われる．

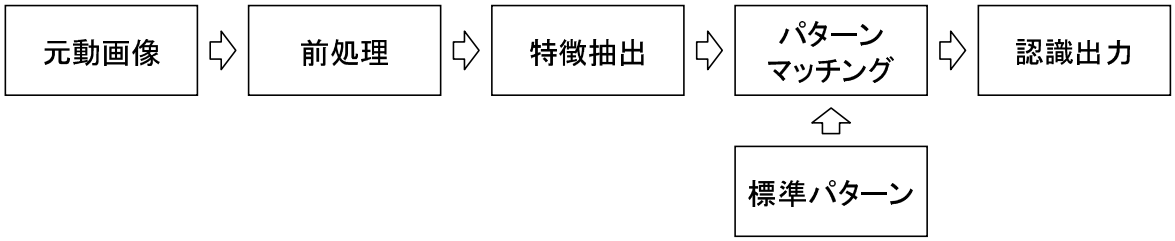


図 2.9: 透視変換

2.4.3　パターンマッチング

入力された画像から求まった特徴パラメータ，即ち未知のパターンが，前もって用意してある標準パターンと一致するかどうかを認識する作業がパターンマッチングである．あらかじめ計算機内に標準パターンを蓄えておき，その標準パターンと入力した未知のパターンの特徴が一致した時，認識したということができる．蓄積された標準パターン系列を辞書という．

入力画像が標準パターンと完全に一致すればよいが，画像は不確定な要素を多く含むため完全に一致することは困難である．その場合は入力画像の特徴パターンと，各標準パターンとの類似度，または，距離を計算する規則をあらかじめ作っておき，その影響がもっとも類似しているパターンをその画像として認識するという手段が用いられる．簡単な距離としてユークリッド距離が用いられる．

2.4.4　テンプレートマッチング

画像内である特定の対象物を認識したり，複数枚の画像を対象に部分画像が入力画像のどの部分に対応し，一致するかを調べる問題がパターンマッチング (pattern matching) である．このうち，対象物が画像パターンとして表され，探索する画像領域の対象部分との類似度を調べることによって一致する位置を求める手法がテンプレートマッチング (template matching) である．テンプレート画像として認識対象のもととなる理想的な画像パターンを用意する．そして，入力パターンとテンプレートを重ね合わせ，距離を計算する．最後に，距離の最小値を与えるテンプレートに対応する図形クラスを識別結果とする．パターン間の距離の定義としては，いくつか考えられるが，代表的なものは「ユークリッド距離」である．

$$\text{入力パターン} \qquad P_i = (P_{i,1}, P_{i,2}, \ldots, P_{i,n})$$
$$\text{テンプレート} \qquad P_T = (P_{T,1}, P_{T,2}, \ldots, P_{T,n})$$

上式のユークリッド距離は次式で与えられる．

$$D_u = \sqrt{\sum_{k=0}^{n} (P_{i,k} - P_{T,k})^2}$$

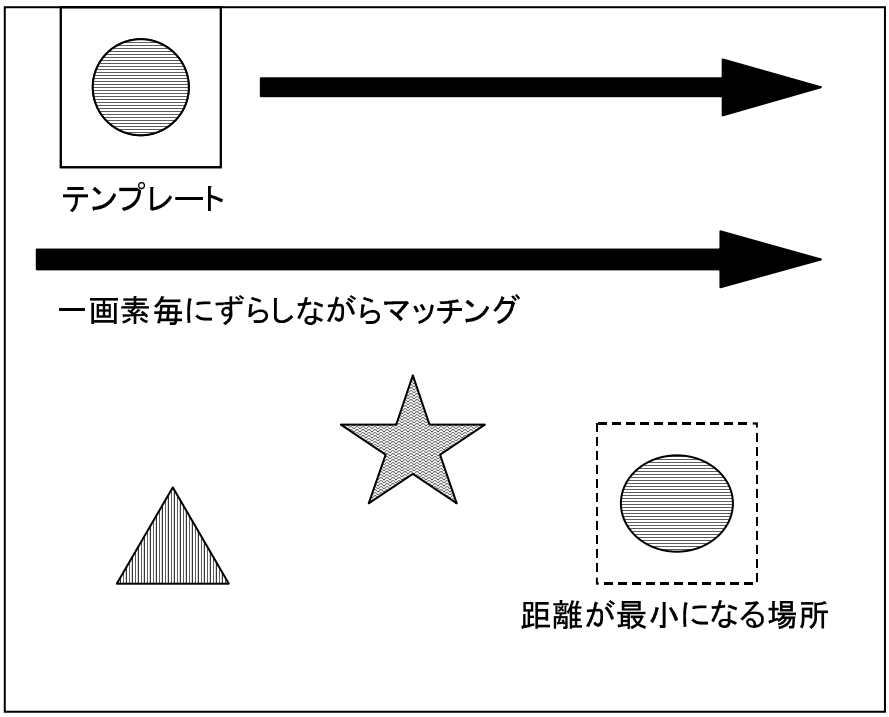


図 2.10: テンプレートマッチングの概念図

テンプレートマッチングは画像のどこに対象とする画像パターンが存在するかを探索する問題にも適応できる．この場合は対象パターンのテンプレートを画像上で 1 画素ずつ動かしながらラスタ走査し，その都度パターン間の距離を計算し，最終的に最小距離を与える場所を検出する．(図 2.10 参照)

第3章　要約生成モデル

本章では、要約を生成するモデルの提案を行う。提案するモデルには、全ての映像に対するモデル、サッカー映像に対するモデルの 2 つがある。

3.1　一般映像の要約生成モデル

要約を生成する上での入力、処理、出力を以下のように定め、処理の概要を図 3.1 に示した。これら全ての処理は、計算機によって自動的に行われることが望ましい。

入力映像は映像分類処理によって対応する映像意味理解処理へ渡され、Index を付加される。付加された Index と付加情報・嗜好情報によって、映像選択処理は入力映像の中から要約映像に使用する部分を決定する。演出効果処理は、選ばれた部分に対して演出を付加し、要約が完成する。

1. 入力

   元映像 要約の元となる映像（入力映像）

   付加情報 元映像に関する情報（映像中の人物・物体情報、センサー情報、元映像に関連する他の映像等）

   嗜好情報 要約画像を見る人の情報（プロファイル）

   演出効果 映像を加工するデータ（実況・解説データや映像・音声加工データ等）

2. 処理

   映像分類 入力された映像のジャンルを識別し、該当する意味理解に渡す。全ての映像に対して有効だが、その映像の特性（競技のルール、取引の仕組み等）を理解することはできない。構造は、階層構造を取る場合もある。（全ての映像の分類処理の下に、スポーツの分類処理がある等。図 3.2 参照）

映像意味理解 ある限られた分野の映像について分析し、Indexing する。自分の専門分野以外のことは認知しない。

映像選択 得られた Index とプロファイルから、要約に使用する画像を選択する。

演出効果付加 映像に演出を付加する。オプションであり、使用されないこともある。

3. 出力

要約映像 目的物

副産物 indexing された映像や被写人、物の評価など

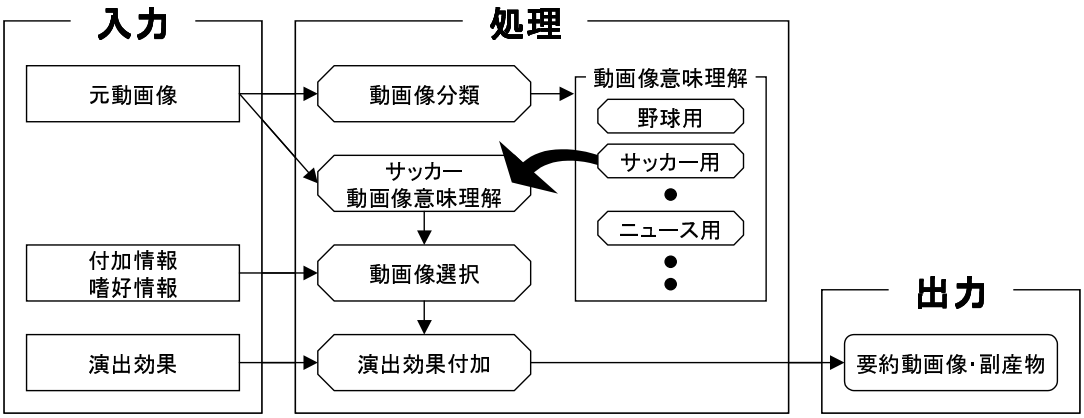


図 3.1: 要約生成の流れ

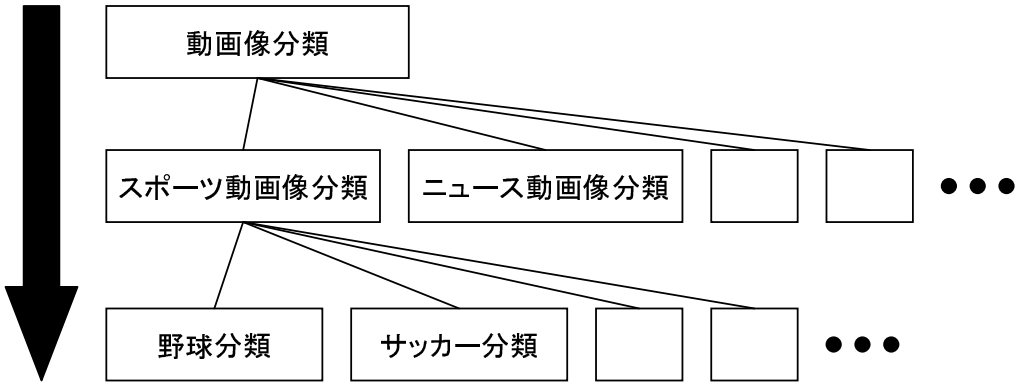


図 3.2: 映像分類階層構造

3.2　サッカー映像の要約生成モデル

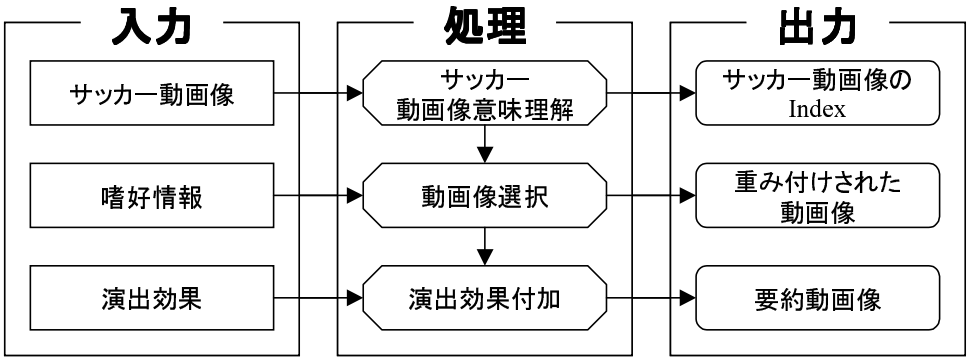


図 3.3: サッカー映像における要約生成の流れ

入力をサッカー映像に限定した場合の処理概要を図 3.3 に示した。この場合、映像分類は必要ない。また、本研究では簡略化のため、入力のうち付加情報・演出効果を、処理のうち演出効果付加を排除した。よって、行う処理は映像意味理解処理と映像選択処理のみとなる。その 2 つを合わせたサッカー映像の要約生成方法を以下のように定義した。

1. カメラワークの変化で映像をシーンに分割する
2. 得られた各々のシーンに対して Indexing を行う
3. シーンを Index 毎の重みによって評価し、得点付けする
4. 出力すべき映像の長さとなるように、得点の高いシーンを選択し、つなぎ合わせる

1,2 が映像意味理解処理、3,4 が映像選択処理に当たる。また、この処理により、出力における副産物として、Index 付けされたサッカー映像、嗜好情報によって重み付けされたサッカー映像を得ることができる。

この処理を数式で表すと以下のようになる。

元映像を M とした場合、1. の処理によって、M はシーン $S_1, S_2, \ldots, S_n$ に分割される。また、元映像の時間長を T とすると、各シーンは、長さ $T_1, T_2, \ldots, T_n$ を持つ。

$$M = \{S_1, S_2, \ldots, S_n\}$$

$$T = \{T_1, T_2, \ldots, T_n\}$$

続いて、2. の処理で Indexing を行い、各シーン S_i に対して Index $I_{S_i1}, I_{S_i2}, \ldots, I_{S_ih}$ が与えられる。

$$S_i = \{I_{S_i1}, I_{S_i2}, \ldots, I_{S_ih}\}$$

また、入力である嗜好情報によって、各 Index に対応する重み $O_1, O_2, \ldots, O_l$ が定義される。そして、3. の処理によって、各シーンの評価 $P(S_i)$ が定まる。

$$\begin{aligned} P(S_i) &= P(\{I_{S_i1}, I_{S_i2}, \ldots, I_{S_ih}\}) \\ &= \{I_{S_i1}, I_{S_i2}, \ldots, I_{S_ig}, \ldots, I_{S_ih}\} \times \{O_1, O_2, \ldots, O_k, \ldots, O_l\} \end{aligned}$$

ここで、演算子 $\times$ は、IndexI_{S_ig} と、その Index に対応する重み O_k を掛けた和をとることを意味する。

求める要約映像を m、時間長を t とすると、4. の処理によって、t を越えない時間で $P(S_i)$ の高い順に $S_1, S_2, \ldots, S_m$ が時系列順に並べられ、m が完成する。X は結果的に得られる $P(S_i)$ の閾値となる。

$$m = \{S_{r_1}, S_{r_2}, \ldots, S_{r_i}, \ldots, S_{r_j}\}$$

$$P(S_k) \begin{cases} \geq X & (k \in \{r_1, r_2, \ldots, r_j\}) \\ < X & (\text{それ以外}) \end{cases}$$

$$r_i < r_{i+1}$$

$$t \geq \sum_{k=1}^{j} T_{r_k}$$

3.3　サッカー映像における Index

Indexing によって付加される Index には、カード名、スコア等多様なものがある（表 3.1 参照）が、本研究では、選手・ボールのサッカーフィールドにおける絶対座標の取得に重点を置いて作業を行った。絶対座標とは、その名の通り、選手・ボールのサッカーフィールドにおける位置を一意的に表すものである（図 3.4 参照）。

カード名 試合を行った 2 チームの名前

スコア 得点状況

時間 試合時間。サッカーの場合、前半後半開始からの時間と試合通算の時間という 2 種類の表記方法がある

選手名 試合を行う選手の名前

プレイ ドリブル、パス、シュートなどの選手の行う行動

イベント ゴール、フリーキック、コーナーキックなどのサッカーの試合で起こる事象

選手座標 選手のフィールド上での座標

ボール座標 ボールのフィールド上での座標

カメラアングル カメラの映しているフィールドの位置や角度

表 3.1: サッカー映像の Index（一部）

カード名	スコア	時間
選手名	プレイ	イベント
選手座標	ボール座標	カメラアングル

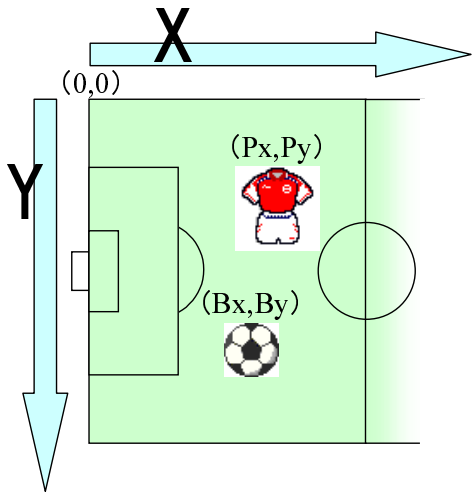


図 3.4: フィールドにおける絶対座標

第4章　要約システム概要

この項では，本研究が提案するTVサッカー映像の要約手法について述べる．サッカー映像の画像をカメラアングルによって分類すると，フィールドの広範囲を映したフィールド遠景（遠景），選手のアップを映したフィールド近景（近景），観客席や監督を映したその他，の3つに分類できる（図4.1参照）．

これらの3つの映像に対して処理を行うことで得られる要約に関係するIndexを挙げると，遠景では選手やボールの位置，シュートやゴール等のイベント（逆に選手名は認識困難），近景では選手名やシュートやゴール等のイベント（逆に選手やボールの位置は認識困難），その他ではフィールド外の様子等である．さらに，TV放送において付与される映像効果を認識することによって，リプレイ画像を認識することも可能である．また，音声部分については，観客の声援や実況解説の音響・音声情報が挙げられる．（本研究チームのこれまでの研究内容については[1][3]参照）これらを組み合わせ，以下の様なTVサッカー映像の要約生成システムを発案した．

図 4.1: 遠景（左上），近景（右上），他（下）

4.1　一般的な要約生成システム

一般的な要約とは，試合をする 2 チームを公平に扱った TV のダイジェストや，試合の流れが判るダイジェストのことを挿す．システムの評価は TV のダイジェストと比較する方法が最も簡単である．

一般的な要約生成システムは，図 4.2 の様になる．一般的な要約を生成する上では，音響情報やリプレイ認識による絞込で効率良く要約候補シーンを検出できることが明らかになっている．そのため，これらの処理を最初に施すことによって，処理に時間のかかる Indexing 処理（広義では音響情報処理やリプレイ認識処理も Indexing に含まれるが，ここでは選手やボールの認識処理を Indexing としている）は，全体の動画像の 10%から 50%程度（要約の精度と生成時間のバランスによって決定される）に対してのみ行うだけでよく，要約生成時間を大幅に高速化することができる．

現在 TV で放送されている一般的な要約には，質の高いものから，ただ単に得点シーンをつなげたものまで様々なものが存在する．これを，客観的に判断する要約システムにまとめることで，質の高い要約を効率的に生産することが可能になる．

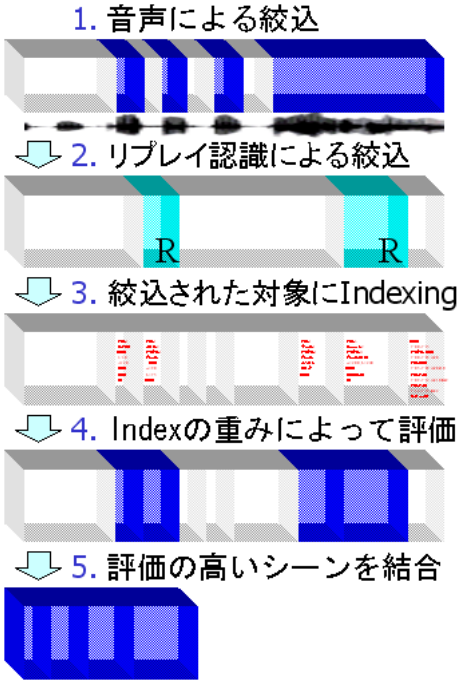


図 4.2: 一般的な要約生成システム

4.2　理想的な要約生成システム

理想的な要約とは，個人の嗜好を満たす要約と定義する．具体的には，あるチームの攻撃シーンのみを集めたものや，ある選手のドリブルシーンのみを集めたものが考えられる．システムの評価は，個人の満足度を調査する方法，国家代表の試合のダイジェストや，地方の TV 局の地元チームに偏ったダイジェスト，選手個人に注目した番組との比較が考えられる．

理想的な要約生成システムは，図 4.3 の様になる．理想的な要約生成システムでは，一般的な要約生成システムと違い，音響情報やリプレイ認識による絞込が不可能なため，全フレームに対する Indexing 処理を行う．

Indexing 処理自体は一度行えばよく，異なる Index の重み係数（個人の嗜好データとして実装）と掛け合わせれば様々な要約を生成できるため，非常に効率的と言える．また，プレイの連続する区間毎に切り分けることが可能なため，システムの分散化も容易である．

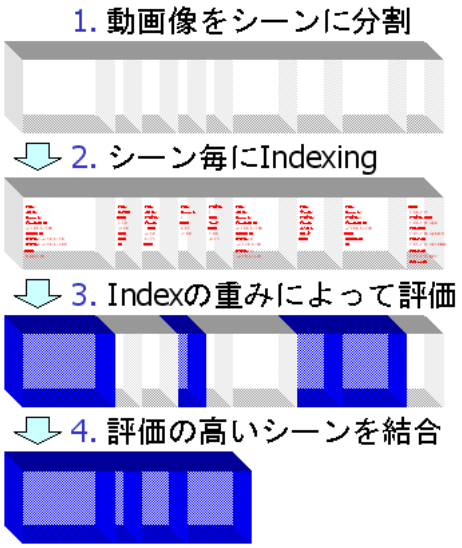


図 4.3: 理想的な要約生成システム

第5章　サッカー映像の意味理解手法

本章では，実際に行った意味理解処理の内容を示す.

遠景の Indexing 処理手法は，フィールド上の選手，ボールの位置を一意な座標に変換する絶対座標取得部と，その座標を元にドリブルやパス，シュートを認識するイベント認識部に分かれる．また，イベント重みの生成は，TV 放送映像とその要約を手本とし，認識処理によって得られたメタ情報と比較することで生成することとした.

処理を行うプログラムは Microsoft の VisualC++で作成した.

5.1　入力映像

本研究で使用した入力映像（＝元映像）のフォーマットは以下のとおりである.

- 幅 (width) 320,360,640,720 dot
- 高さ (height) 240,480 dot
- 29.97fps
- 24bit Color
- VfW(Video for Windows) 規格に準拠した Avi ファイル
- マスメディアによって放送されたサッカー映像

入力映像を M とし，入力映像中に含まれる画像（フレーム）を $F_1, F_2, \ldots, F_{frameMAX}$ とする．以後の処理は，各フレーム F_i に対して行うこととする.

5.2　絶対座標取得部

この項では，フィールド遠景（フィールド広範囲を映した画像）に画像処理を行い，選手やボールの一意な座標を取得する作業にについて述べる．

本研究では，全ての遠景画像は，フィールド上での一意な位置を表す絶対座標上に変換することによって処理する．各処理によって認識された選手やボールの位置は，絶対座標上に置かれることで，プレイやイベントの認識に貢献する．この処理によって，選手やボールの絶対座標上での動きや，それによるイベントの認識が可能となる．処理の結果は，フレーム毎の結果として予め定められたデータ形式で出力される．

各処理は，シーン分割，シーン分類，フィールド変換，ボール認識，選手認識に分けられる．以下に，各処理毎の処理内容をまとめた．

5.2.1　シーン分割

まず，入力映像をシーンに分割する．分割は，カメラワークの切り替わりを検出することによって行う．これは，入力映像が持つ，選手のアップ，フィールド遠景，観客席等の様々な景色は，カメラワークの切り替わりによって変化するからである．

具体的には，フレーム F_t とフレーム F_{t+1} での RGB ヒストグラムの変化量 $S(t)$ を計算し，シーンの切り替わりを判定する閾値 k より大きい場合に，シーン分割と判定した（図 5.1 参照）．この処理によって，M はシーン $S_1, S_2, \ldots, S_n$ に分割される．ここで，各シーン S_i は複数のフレーム $F_j, F_{j+1}, \ldots, F_k$ で構成され，$j = S_iST, k = S_iED$ とおくことにする．（当然，$S_iED + 1 = S_{i+1}ST$ が成立する）

$$R(t) = \sum_{i=0}^{255} \mid r_{t+1}(i) - r_t(i) \mid$$

$$S(t) = R(t) + G(t) + B(t)$$

$r_t(i)$: フレーム F_t中での r 濃度 i の画素数

$S(t) \geq k$ 以上のフレームで，シーンを分割（本研究では $k = 100000$ を採用）

$$M = \{S_1, S_2, \ldots, S_n\}$$

$$\begin{aligned} S_i &= \{F_j, F_{j+1}, \ldots, F_k\} \\ &= \{F_{S_iST}, \ldots, F_{S_iED}\} \end{aligned}$$

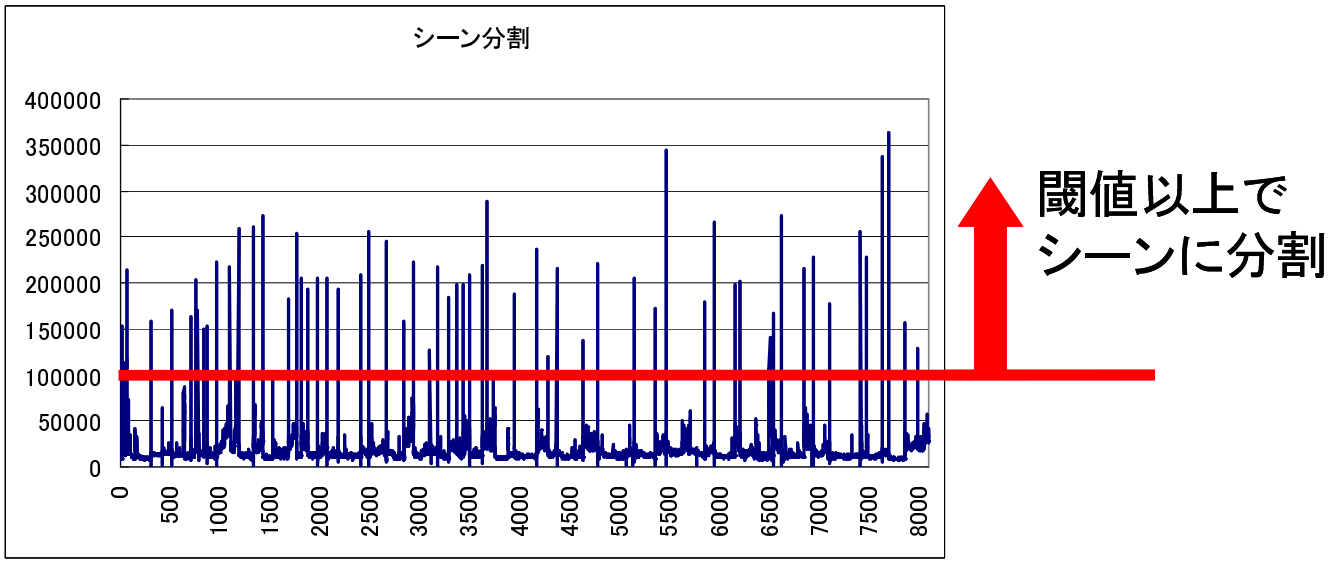


図 5.1: シーン分割グラフ

5.2.2　フィールド認識

続いて，サッカー画像におけるフィールド（＝グラウンド，ピッチ，芝生）を認識する．これは，サッカーフィールドを認識することによって，以後の選手・ボール探索等の手助けとするためである．(選手・ボールは当然フィールド上にしか存在しない)．

具体的には，まず，輝度 Y の値によって，入力映像の画素を白，黒，その他に分類した（$Y < Y_{cut}$ を黒，$Y > 255 - Y_{cut}$ を白とし，本研究では $Y_{cut} = 40$ を採用）．続いて，F_t におけるその他の部分の色相ヒストグラム h_t を元に，その和（入力映像全体の色相ヒストグラム）h を作成する．そして，h の最大値画素数の 20 % を閾値 H_{cut} として，それ以上の画素数を持つ色相 i をフィールドの色相とした．(図 5.2 参照)

$h_t(i)$: フレーム F_t における色相 i の画素数

（ただし，$Y_{cut} \leq Y \leq 255 - Y_{cut}$ である画素のみが対象）

$$h(i) = \sum_{t=1}^{frameMAX} h_t(i)$$

$$H_{cut} = MAX(h(i)) * 0.2(0 \leq i \leq 359)$$

$$h(i) \geq H_{cut} \Rightarrow i \in H_{feild}$$

H_{feild} : フィールドの色相

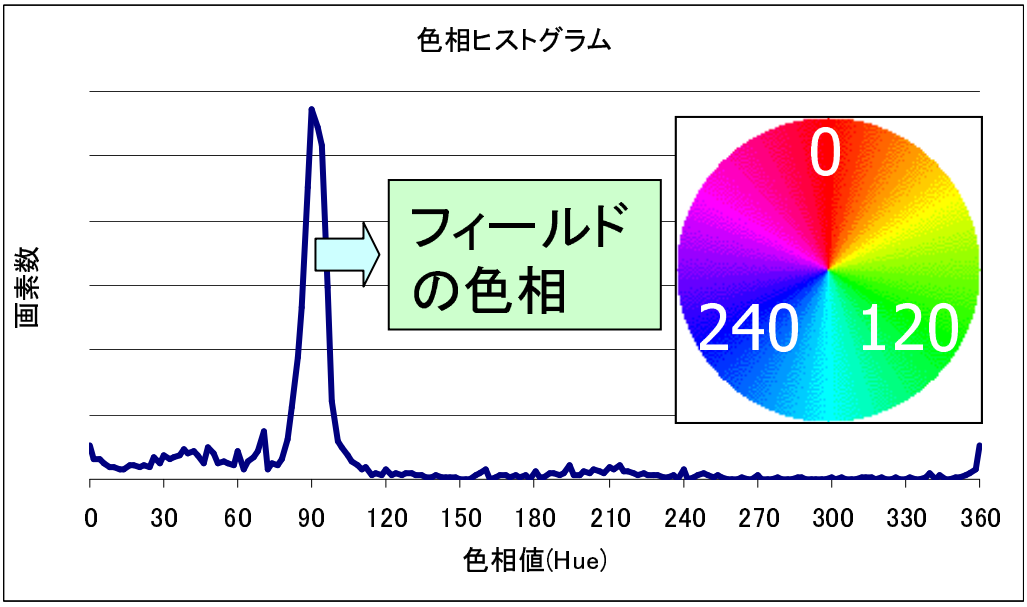


図 5.2: シーン分割グラフ

5.2.3　シーン分類

次に，4.2 で分割したシーンを分類する．これは，シーンを分類することによって，以後の処理を行うシーンを決めるためである．分類するシーンは以下の 3 つに定めた．

フィールド遠景　フィールドを，遠く離れた視点から見たシーン

フィールド近景　フィールドを，近く寄った視点から見たシーン

その他　上記 2 つに属さない，フィールドを映していないシーン

フィールド遠景とフィールド近景の境界は曖昧だが，選手複数人及びボールの位置が認識できるということをフィールド遠景の条件とした．

分類には，4.3 で求めたサッカーフィールドの色相 H_{feild} を利用する．まず，各シーン毎にフィールドの色相 H_{feild} である画素の平均値 F_{Av}，平均変化量 F_{CH} を求める．また，各フレームの画素を色相によって白，黒，それ以外を色相によって 6 分割したもの（$60*(i-1) \leq H \leq 60*i (1 \leq i \leq 6)$ の式によって 6 分割）の 8 つに分類し，それらの連結成分の合計 $H(f)$（f はフレーム番号）から，平均値 H_{Av} を求める．

$$F_{Av} = \left(\sum_{i=S_tST}^{S_tED} h_i(j \in H_{feild}) \right) / (S_tED - S_tST + 1)$$

$$F_{CH} = \left(\sum_{i=S_tST}^{S_tED-1} \mid h_i(j \in H_{feild}) - h_{i+1}(j) \mid \right) / (S_tED - S_tST)$$

$$H_{Av} = \left(\sum_{i=S_tST}^{S_tED} H(i) \right) / (S_tED - S_tST + 1)$$

H_{feild} は，フィールドの色相であるから，F_{Av} の値が大きいほど，そのシーンがサッカーフィールドであるといえる．また，遠景であるほど，画像のフレーム間での変化量は小さいので，F_{CH} の値が大きいほど，そのシーンは近景であるといえる．さらに，遠景であるほど画像は細かくなるため，H_{Av} の値が大きいほど，そのシーンは遠景であると言える．

最後に，以上の処理によって得られた F_{Av},F_{CH},H_{Av} に対応する閾値 k_1,k_2,k_3 を用いて，シーンを分類する．ここで，各条件（図 5.3 参照）は，以下のようにした．

条件 1 ： $F_{Av} \geq k_1$

条件 2 ： $F_{Av} \geq k_2, F_{CH} < k_3, H_{Av} \geq k_4$のうち 2 つ以上に該当

本研究では，各閾値に次の値を採用した．

$$k_1 = 20000, k_2 = 50000, k_3 = 500, k_4 = 700$$

さらに，分類に用いるフィールドの色相を自動的に決定するアルゴリズムを追加した．また，全体の処理を 1-pass で行う改良後 1 と，2-pass で行う改良後 2 の 2 つを実装した．この 2 つのアルゴリズムは，用途によって使い分けるべきである（逐次的な処理が必要な場合は 1 を，それ以外は 2 を用いるべき）．改良前のアルゴリズムは，芝・45 分・暗のデータに最適化されている．

実際に処理を行った結果を，表 5.1 に示す．また，以後の処理はフィールド遠景に分類されたシーンに対してのみ行う．

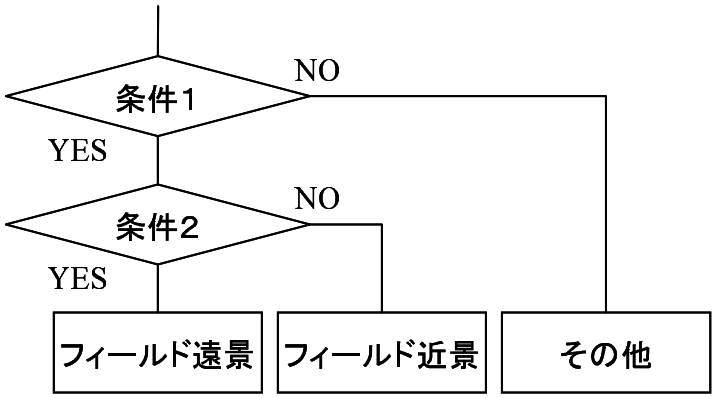


図 5.3: シーン分類処理フロー

表 5.1: シーン分類・分割結果

シーン番号	開始フレーム	終了フレーム	F_{Av}	F_{CH}	H_{Av}	分類
1	0	299	1404	139	747	その他
2	300	512	7177	252	2718	その他
3	513	705	23092	383	653	フィールド近景
4	706	745	68611	785	551	フィールド遠景
5	746	776	55996	1534	925	フィールド遠景
⋮	⋮	⋮	⋮	⋮	⋮	⋮

5.2.4　フィールド外の除去

続いて，画面中からフィールド以外の領域を排除する（図 5.4 参照）．この処理によって，以降で行う選手・ボール認識の精度を向上させることができる．具体的には，4.3 で求めたフィールドの色相 H_{feild} で入力映像を 2 値化し，フィールド色相の領域のうち最大のものをフィールドとする．そして，フィールド領域に含まれない部分を黒 $(r, g, b) = (255, 255, 255)$ に変換した．

5.2.5　フィールド変換

得られた選手，ボールの座標をフィールドでの絶対座標に変換するために，元画像に対してアフィン変換を行う（図 5.5 参照）．ラインの交点等，3 次元アフィン変換の係数

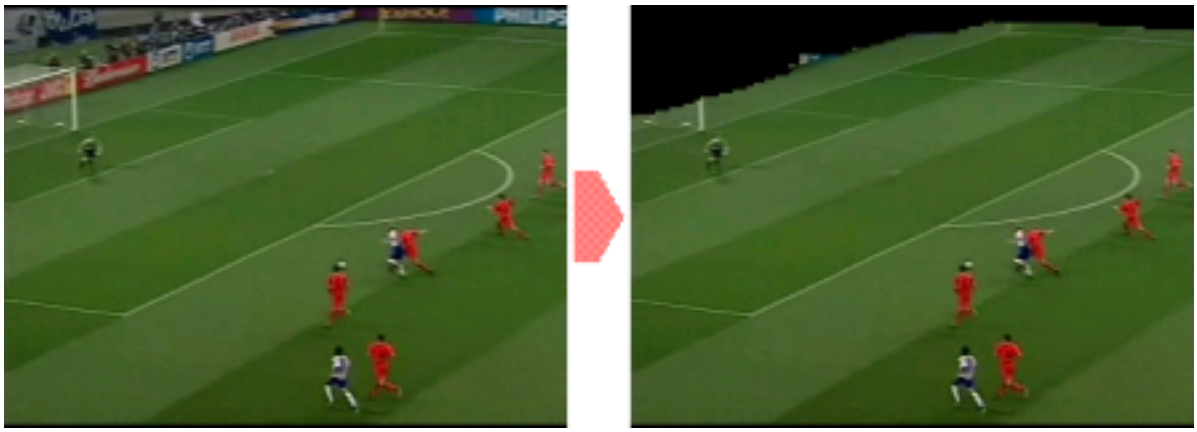


図 5.4: フィールド外の除去

を求める基礎点 35 点（図 5.6，表 5.2 参照）の中から，画面内に移っている 4 点を指定し，元画像全体をサッカーフィールドの大きさに変換する．元映像 (h, w) からサッカーフィールド (H, W) に変換する 3 次元のアフィン変換式は以下である．

$$H = C\frac{Ah + Bw + 1}{Ph + Qw + 1}, W = F\frac{Dh + Ew + 1}{Ph + Qw + 1}$$

フィールド変換は，輝度値で 2 値化された画像を Hough 変換することによって 2 組の平行線対を検出し，その平行線の組み合わせを満たす最も適当なサッカーフィールドをマッチングによって求め，その結果選ばれたフィールド上に Affine 変換（図**??**）する（処理 1）．変換においては，前後フレーム（時間軸）での補完を行い，認識率の向上を図っている．具体的には，前フレームから連続的（距離的にある閾値を越えない移動量）な変化と思われる変換先には，一定の優先度を設けることによって実装した．また，ラインだけでなく，ペナルティアークと呼ばれる円弧の認識や，画像のマッチングを行う際に細線化，太線化等の処理を施すことによって，精度の向上を図っている（処理 2）．

5.2.6　ボール認識

ボールの認識には，サッカーボールの白色・球という性質を利用する．入力画像を輝度 Y の値によって 2 値化し，連結成分ごとに面積 S と周囲長 l を求め，円形度 e を求める．円形度が最大の物をボールとする（一般的な図形の円形度については表 5.3 参照）．

$$e = \frac{4\pi S}{l^2}$$

図 5.5: フィールドアフィン変換

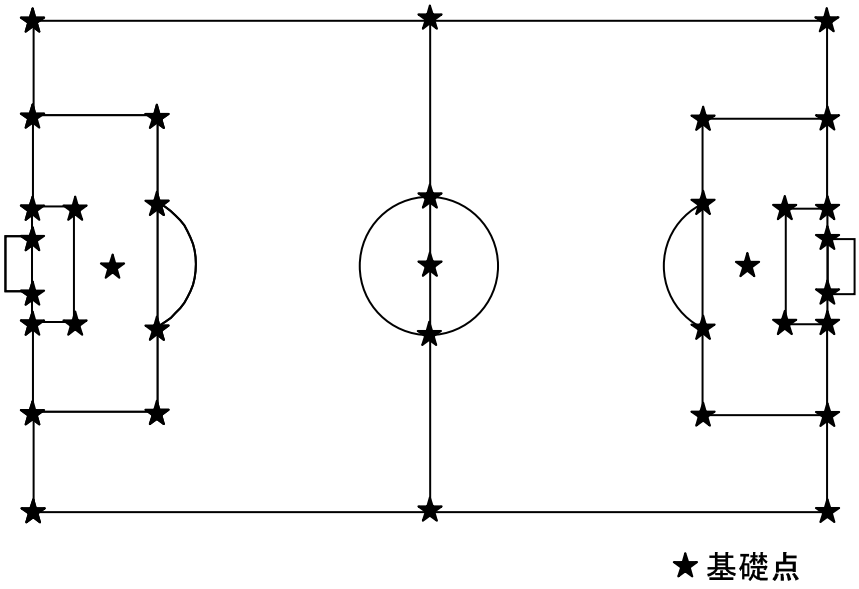


図 5.6: フィールド変換基礎点

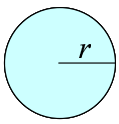
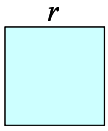
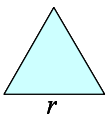
ボールの面積 s は，フィールド変換の値から算出する．シーン開始時は画像すべての範囲に対してボール探索を行い，以降のフレームでは，前フレームでボールがあった位置を中心に 32×32 ピクセルの範囲でボール探索を行った．(図 5.7，5.8 参照)

さらに，ボールに速度とベクトル（と内部的には高さ）パラメータを持たせ，フレーム前後での補完処理を施した．前後処理では，前数フレームの結果を用いて，ボールの移動しうる位置を計算し，精度の向上を図っている（処理 2）．

表 5.2: 基礎点の座標

番号	名前	H(m)	W(m)
1	コーナー左上	0	0
2	コーナー左下	68	0
3	コーナー右上	0	105
4	コーナー右下	68	105
5	ペナルティエリア左・左上	13.84	0
6	ペナルティエリア左・左下	54.16	0
7	ペナルティエリア左・右上	13.84	16.5
8	ペナルティエリア左・右下	54.16	16.5
9	ペナルティエリア右・左上	13.84	88.5
10	ペナルティエリア右・左下	54.16	88.5
⋮	⋮	⋮	⋮

表 5.3: 円形度比較

種類	円	長方形	正三角形
画像			
面積	πr^2	r^2	$\sqrt{3}r^2/4$
周囲長	$2\pi r$	$4r$	$3r$
円形度	1.0	$\pi/4 = 0.79$	$\pi\sqrt{3}/9 = 0.6$

5.2.7　選手認識

選手の認識は，チームのユニフォーム（シャツ・パンツ・ソックス）の絵柄によるパターンによって元画像に対するパターンマッチングを行う（図 5.9 参照）．パターンの大きさは，フィールド変換の結果から定められる．

シーン開始時は画像すべての範囲に対して選手探索を行い，以降のフレームでは，前フレームの選手・新たに画像に現れる選手を追跡するために，前フレームで選手がいた位置と，フレームの外周部に対して選手探索を行った．

図 5.7: ボール認識：最初のフレームへの処理

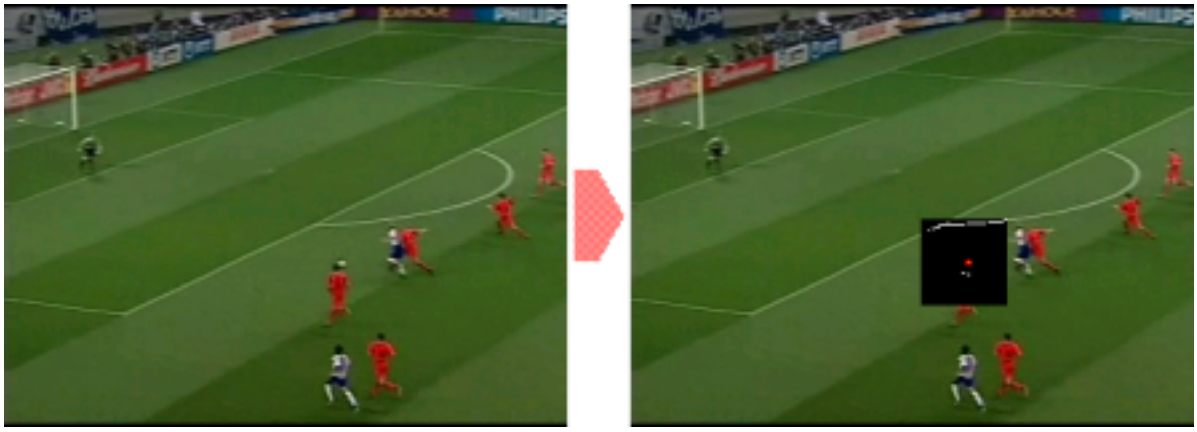


図 5.8: ボール認識：連続フレーム間での処理

改良アルゴリズムとして，選手は絶対に消えてなくなりはしないという知識に基づき，選手が 2 人以上重なっているという内部状態を追加した（処理 2）.

5.2.8　手動補正

以上の処理の誤りを訂正するため，手動による補正ツールを作成した（図 5.10 参照，補正ツールは，処理ツールの改造版として実装されている）．補正ツールには以下の機能がある.

- 自動 Indexing の結果を表示し，修正・補正する
- 修正した結果を元に，以降を再自動 Indexing する

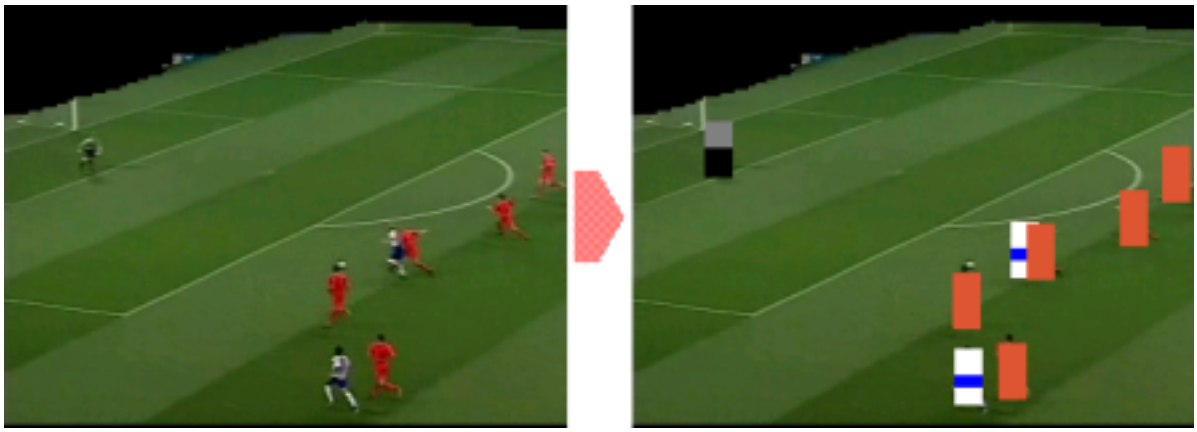


図 5.9: 選手認識

- ライブ映像と VTR を区別するため，シーンに対して VTR 属性を付加させる

5.2.9　出力データ

以上の処理によって得られた結果は，処理ツールによってテキストファイルとして出力される（図 5.11 参照）．ここで，出力データ中の $[d_1, d_2, d_3, d_4, d_5]$ というデータは，以下の値を表している．

(d_1, d_2) : 変換画像中での座標 (H, W)

(d_3, d_4) : 元画像中での座標 (h, w)

d_5 : 選手の背番号

図 5.10: 補正ツール画面

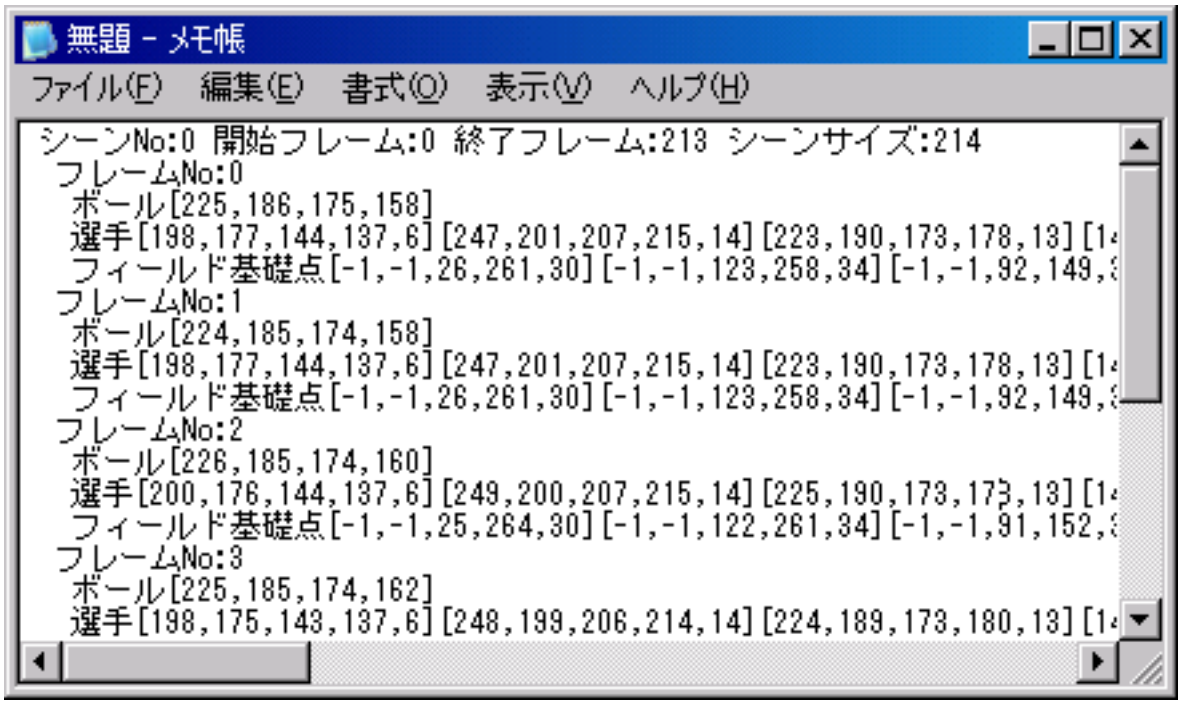


図 5.11: 出力データ

5.3　イベント認識部

この項では，得られた選手・ボールの絶対座標を元に行うイベント認識について説明する．ここで，認識するイベントは，ドリブル，パス，クロス，シュート，ゴールとし，パス，クロス，シュートには成功，失敗の区別も行った．各イベントの認識規則は以下のようにした．この処理によって，自動的に Index を取得することが可能になる．

1. ドリブル
   (a) 選手がボールの半径 3m 以内に存在するものをドリブルとする
   (b) 候補選手が複数いる場合，ドリブル時間が長い選手を優先する
   (c) ドリブル終了時までに，一度もボールのベクトル，速度が閾値以上変化しない場合は，ドリブルとして認めない
2. パス
   (a) 味方のドリブルから味方のドリブルの間を成功パスとする
   (b) 味方のドリブルから敵のドリブルの間を失敗パスとする
   (c) パスの条件を満たし，かつ，フィールドのサイドからフィールド中央へのパスだった場合，特別にクロスとする
3. ゴール
   (a) ゴールの枠内にボールが移動した場合，ゴールとする
4. シュート
   (a) ある選手から，敵チームのゴール付近にボールが移動した場合，シュートとする
   (b) シュートの後にゴールが発生した場合，成功シュートとする
   (c) それ以外のシュートは失敗シュートとする

5.4　イベント重み生成部

この項では，前項までに得られたイベントを元に，各イベントの重みを生成する．対象となるイベントは表 5.4 の様になり，同時にこれが各遠景シーンに振られるメタ情報となる．メタ情報は，絶対座標とイベントの認識結果から，要約に深く関係しそうな項目

を人間が選んだ．サッカー映像の完全な解析を目指すならば，このメタ情報を学習によって導出することを検討する必要がある．

このメタ情報を各シーン毎に算出したベクトル V_s（s はシーン番号）と，手本となる要約がそのシーンを採用したかどうかの情報 $A_{s(n)}$（s はシーン番号，採用されれば 1，されなければ 0，n は手本となる要約の番号）によって，イベント重み $W_{(n)}$ を導く．また，$W_{(n)}$ をまとめた W も生成する．

$$V_s(s = 0, 1, \ldots, scene_{MAX})$$

$$\begin{aligned} A_{s(n)} &= 1 \quad (\text{手本要約 } n \text{ に } V_s \text{が含まれる場合}) \\ &= 0 \quad (\text{それ以外}) \end{aligned}$$

$$V_s^t W_{(n)} = A_{s(n)}$$

$$W = Av(W_{(n)})$$

イベント重みの評価は，n 試合の Indexing されたデータとその試合の手本要約で作った重みを，他の m 試合の Indexing されたデータに対して適用し，その結果得られる要約と，その試合の手本要約の採用シーンの一致率で判断する．

表 5.4: 1 シーンのメタ情報

シーン情報		型
スコア		int[2]
時間		time[2]
攻め手		int
パス	本数	int
	平均速さ	double
	平均長さ	double
	最大速さ	double
	最大長さ	double
ドリブル	回数	int
	平均速さ	double
	平均長さ	double
	最大速さ	double
	最大長さ	double
シュート	回数	int
	平均速さ	double
	平均長さ	double
	最大速さ	double
	最大長さ	double
特殊イベント	CK(IN)	bool
	CK(OUT)	bool
	FK(IN)	bool
	FK(OUT)	bool
	GK(IN)	bool
	GK(OUT)	bool
	SI(IN)	bool
	SI(OUT)	bool
ボールとゴールの最短距離		double
GK のボールへの接触回数		int
ゴール前のフリー度		double

第6章　意味理解の評価

本章では，意味理解処理を行った結果の評価を行う．

6.1　絶対座標取得部

6.1.1　シーン分割

1試合90分のデータに対してシーン分割処理を行い，手動で調べた場合と比較した．(表6.1参照)

表 6.1: シーン分割の評価

項目	数	確率
シーンチェンジを正しく認識	174	94.6%
シーンチェンジ認識できず	10	5.4%
シーンチェンジでないものを誤認識	7	-

実用に足る十分な認識率を示している．シーンチェンジを認識できないケースとしては，低速なフェードが，誤認識するものとしては，急激な揺れが挙げられる．

6.1.2　シーン分類

1試合90分のデータに対してシーン分類処理を行い，手動で分類したものと比較した．(表6.2参照)

フィールド遠景，その他については高い正答率を示した．フィールド近景の値が低いのは，その他に分類されてしまうためである．近景で選手が画像領域の大部分を占めると，フィールド領域が減少し，その他と区別するのは困難になる．

ただし，この分類でもっとも重要な点は，その後の処理を行うフィールド遠景と他と

表 6.2: シーン分類の評価

		自動分類			総数	正答率
		遠景	近景	その他		
手動分類	遠景	50	8	0	58	86.2%
	近景	3	24	19	46	52.2%
	その他	4	0	16	20	80.0%

表 6.3: フィールド変換の評価

データ	フレーム数	処理 1	処理 2
シーン 1	200	74.6%	92.0%
シーン 2	200	69.6%	88.6%
5 分データ 1	18000	68.6%	87.5%
5 分データ 2	18000	65.9%	88.2%

を区別することであり，その点では非常に高い認識率が得られている．

6.1.3　フィールド外の除去

1 試合 90 分のデータと 3 試合 20 分のデータにおけるフィールド遠景において，サッカーフィールドの色相である最大の連結成分が，サッカーフィールドである確率は 100%であった．よって，この処理は効果的に働いていると言える．

6.1.4　フィールド変換

結果（表 6.3）の通り，特に改良された処理 2 において，高い精度を得ることができた．認識率を落とすような画像としては，2 組の平行線対が見つからないものや，低地からのアングルで，平行線対を見つけにくいものが挙げられる．これらの認識には，パノラマ画像を生成することによる認識，芝目やフィールド外の広告看板による認識，等が有効に働くと考えられる．

表 6.4: ボール位置認識の評価

データ	フレーム数	処理 1	処理 2
90 分試合	200	61.0%	67.5%
20 分試合	200	53.5%	53.5%
20 分試合	200	66.5%	84.0%
20 分試合	200	55.3%	68.0%
上記平均	800	59.6%	68.3%
90 分平均	18000	53.9%	62.5%

6.1.5　ボール認識

90 分のデータ 1 試合と 20 分のデータ 3 試合におけるボール認識率を示す（表 6.4 参照）. 調査フレームは，連続しないことを条件に無作為に選択した.

平均して 6 割前後の認識率が得られたが，ゴール前の混戦等，選手とボールが集中するシーンでは，認識率が 30%程度まで低下してしまうことがある．特に，白いシャツ，パンツ，ソックスの選手がいると，誤認識を起こしやすい.

また，前フレームのボール位置によって追跡を行うので，一度誤認するとその後，継続的に誤認する（バースト誤り）．さらに，フレーム外やフィールド外除去で消された場所にあるボールは認識できない.

そして，映像は 2 次元の情報で与えられるため，ボールの高さを検出することが不可能である.

誤認識については，映像の画質を上げることにより，ある程度克服可能であるが，その他の問題は入力映像をマスメディア映像に限定すると，克服するのは困難である.

6.1.6　選手認識

パターンマッチングにより，選手の認識は詳しく行うことができるが，すべての画素に対して，5 回の走査（選手 2 回，キーパー 2 回，審判 1 回）を行っているため，処理に時間がかかる.

評価は，選手が単体フレームで認識できれば良いというもの（継続無）と，前後フレー

表 6.5: 選手位置認識の評価（継続無）

データ	フレーム数	処理 1	処理 2
ゴール前	200	86.6%	87.5%
中央	200	89.1%	89.1%
上記平均	400	87.9%	88.3%
90 分試合	18000	84.4%	86.2%

表 6.6: 選手位置認識の評価（継続有）

データ	フレーム数	処理 1	処理 2
ゴール前	200	41.5%	67.0%
中央	200	87.4%	87.6%
上記平均	400	64.5%	77.3%
90 分試合	18000	70.3%	80.5%

ムで選手が追えているかも加味するもの（継続有）の 2 つで行った．なお，明らかに，継続無の確率 > 継続有の確率 である．処理を行った結果を表 6.5（継続無），表 6.6（継続有）に示す．

結果を見るに，処理 2 は，ゴール前で特に強力な成果を残している．これは，ゴール前では選手が重なりやすく，新たに追加したアルゴリズムが有効に作用し易かったからだと思われる．認識率が低いシーンとしては，FK（フリーキック），CK（コーナーキック）等で，特にゴール前に選手が集中するような状況である．特に，FK で作られる選手の壁では，多数の（4 人以上の）選手が密接し，シーン認識率（継続有）が 30% 程度にまで落ち込む．これらを解決するには，内部状態としてさらに”壁”等を追加する必要があると思われる．

6.2　イベント認識部

評価は，人間が手動で Indexing 処理した結果との比較によって行った．

実際に，イベントに絡む選手 2 人のボールとの距離と，得られた Index の例を以下に示す.（図 6.1 参照）．各添字は，上側がイベントの Index を，下側が選手の背番号（自動

Indexing の場合は内部番号）を表している．

発生しやすい誤りについて，以下にまとめた．

(1) 挿入，欠落誤り

本来は存在しない Index が振られる挿入誤り，存在するはずの Index が振られない欠落誤りが起きる．挿入誤りは,「成功パス」の Index が「成功パス」「成功パス」となるケースが，欠落誤りは,「ドリブル」「成功パス」「ドリブル」の Index が「ドリブル」となるケースが多い．

これらの問題は，選手，ボールの移動速度，ベクトルを利用する，各イベントの最低必要時間を定める等の方法を用いれば解決できる可能性がある．

(2) イベントの認識誤り

本来得るべき Index とは違った Index を振ってしまう場合も考えられる．これは，特に「パス」や「クロス」を「シュート」と認識するという形で起こりやすい．手動 Indexing においては，ラベラーが選手の意図を汲むことができるが，自動 Indexing の場合には，ボールがゴールに向かった場合に，画一的にシュートと判定してしまうためである．

この問題の解決には，シュートのイベント定義を見直す必要がある．

(3) 開始，終了時刻（フレーム）の誤差

図 6.1 の 2 番目の Index（成功パス）に現れているように，Index の開始，終了時刻に誤差が出る．これは，手動で振られた Index 自体がラベラーの影響を受けてしまうため，正解データをどう定めるかという点で問題がある．

しかし，なるべく手動の結果に近くなるという観点で処理を行うならば，最適な結果の出るように各閾値を調整する必要がある．

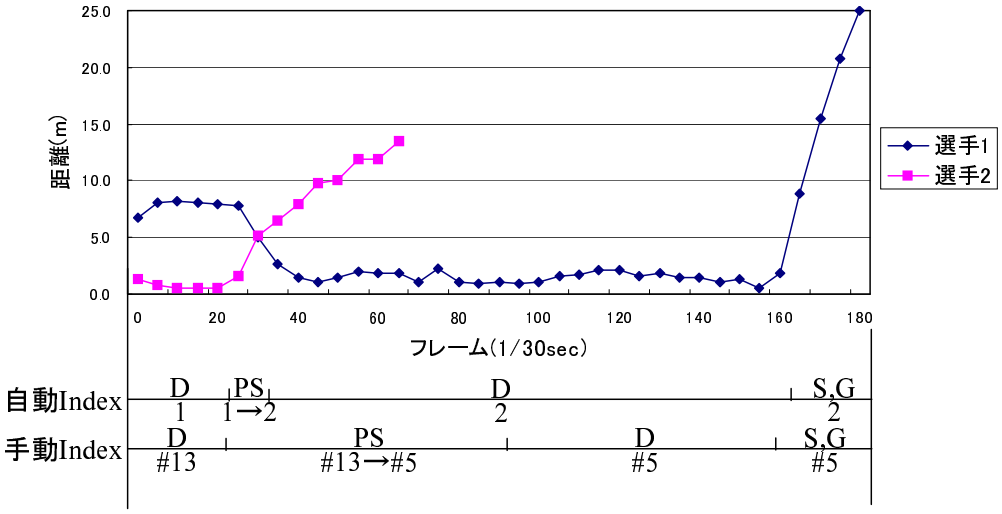


図 6.1: イベント認識の例

第7章　まとめ

本章では，本研究のまとめを述べる．

7.1　まとめ

画像処理を施すことによって，選手・ボールのフィールド絶対座標という Index を自動的かつ高精度にに取得することができた．取得できた座標をグラフ上に展開したものを図 7.1 に示す．この座標は，図 7.2 に対して処理を行った結果として得られたものである．

さらに，イベント認識を行うことによって，サッカーイベントを自動的かつ高精度に取得することができた．また，イベント重みを自動的に取得する手法を提案し，ある程度の結果を出すことができた．

また，本研究と同様の手法によって，ラグビー・バスケットボール等の球技も Indexing することが可能である．さらに，選手・ボールの座標によって，客観的なサッカー選手評価システム OPTA [10] の評価値を自動的に算出することも可能になる．

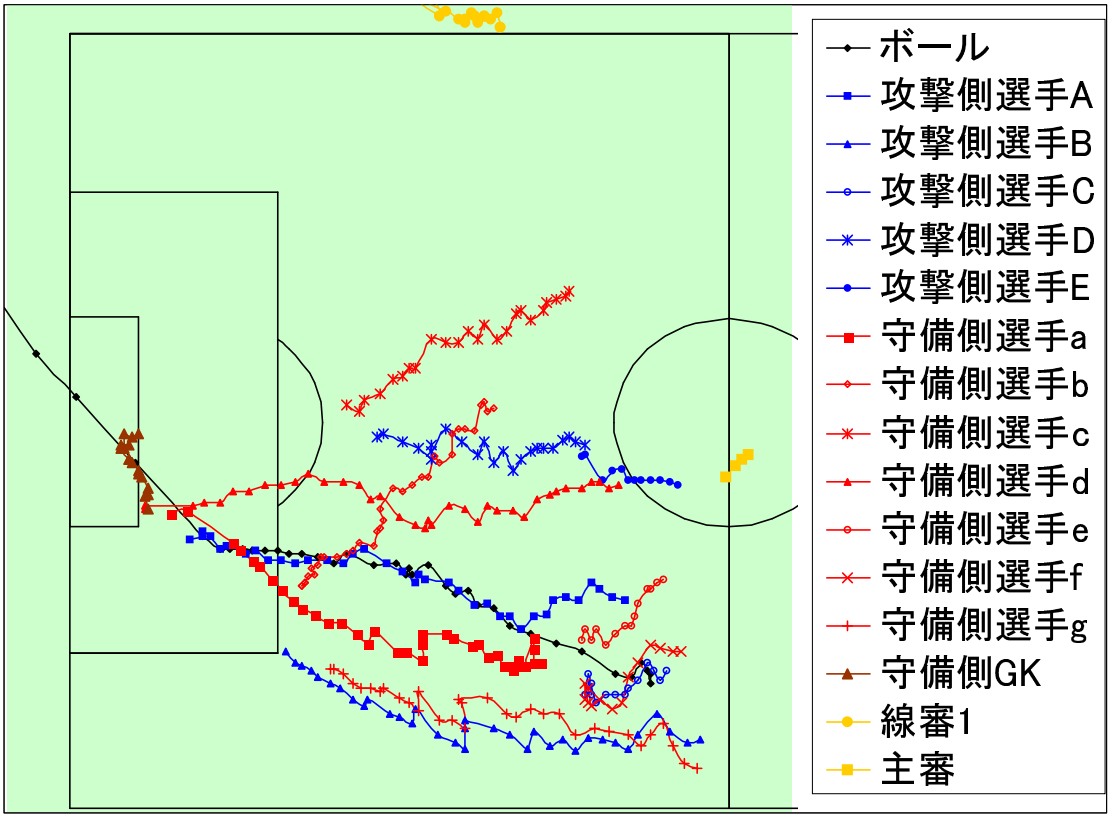


図 7.1: 取得座標

図 7.2: 処理を行ったシーンの流れ

関連図書

[1] 稲葉大樹, “ 要約映像生成における映像の断片化と自動インデクシング”, 2001 年度早稲田大学白井研究室卒業論文,Feb 2002

[2] 稲葉大樹, “ フィールド情報に基づくサッカー近景映像の自動インデクシング”, 2003 年度早稲田大学白井研究室修士論文,Feb 2004

[3] 塩崎崇, “ 音響信号処理に基づくサッカー映像のインデクシング手法”, 2003 年度早稲田大学白井研究室卒業論文,Feb 2004

[4] 中川靖士, 羽田久一, 今井正和, 砂原秀樹, “ サッカー画像の自動ゲーム分析” マルチメディア通信と分散処理 106-33 コンピュータセキュリティ 16-33,Feb 2002

[5] 谷本真人, 椋木雅之, 池田克夫, “ シーンを構成するイベントの検出に基づくスポーツ映像のインデクシング”, 信学技報 TECHNICAL REPORT OF IEICE. PRMU 2000-169,Jan 2001

[6] 寺尾元宏, 椋木雅之, 池田克夫, “ カット構成の規則性を利用したスポーツ映像の構造化”, 信学技報 TECHNICAL REPORT OF IEICE. PRMU 2000-170,Jan 2001

[7] 河合吉彦, 馬場口登, 北橋忠宏, “ 個人適応を指向したスポーツ要約映像の生成法”, 信学技報 TECHNICAL REPORT OF IEICE. PRMU 2000-171,Jan 2001

[8] 石島健一郎, 椎名誠, 相澤清晴, “ 個人体験映像の構造化と要約 -生体情報を用いた映像要約によるライフメディア-”, 信学技報 TECHNICAL REPORT OF IEICE. IE2000-23,PRMU2000-48,MVE2000-52,Jul 2001

[9] 村瀬洋,V.V.Vinod, “ 局所色情報を用いた高速物体探索-アクティブ探索法-”, 電気情報通信学会論文誌 D-ll Vol. J81-D-ll No.9,Sep 1998

[10] http://www.optaindex.com OPTA INDEX

謝辞

本研究を進めるにあたり，研究環境を整え，折に触れて適切なご指導・ご助言を下さった白井克彦教授，誉田雅彰教授に深謝いたします．

適切なアドバイスを下さった画像班 OB の村上さん，白井研 OB の大平さん，本当にありがとうございました．

また，研究室配属当初から様々なことでお世話になった白井研究室の先輩方に深く感謝いたします．特に画像班 OB の籠谷さん，稲葉さん，大隈さんにはゼミなどで色々なアドバイスを頂き大変感謝しています．

さらに，卒業論文という大変な作業を共に頑張ってきた白井研究室の M2 のみんなに感謝いたします．特に，一緒に徹夜などの苦労を共にした，画像班 M2 同期のみんなには感謝の気持ちでいっぱいです．

加えて，共に同じ道を歩んだ後輩の皆にも感謝いたします．来年度以降も研究がんばってください．

自分が白井研究室を選び，卒業までの約 3 年間を過ごしたことは，とても良い選択だったと思っています．本当にありがとうございました．

最後に，本大学への進学に理解を示し，6 年間もの学業生活を支え，温かく見守って下さった両親に深く感謝いたします．これからはきっと楽をさせることができると思います．たぶん．

2005 年 2 月

川口克則