

Kinect を用いた撞球姿勢判定システムの開発

三島聖志[†] 菅沼明[†]

概要: ビリヤードは幅広い年齢層のプレイヤーが楽しむことができるゲームである。しかし、ビリヤード初級者にとって適切な撞球姿勢をとることはなかなか難しいため、思い通りに手球を撞くことができない。本研究では、プレイヤーの服の7か所に色マーカを取り付け、それを Kinect で撮影することで、肩・肘・手首・腰の3次元位置を取得し、撞球姿勢として不適切な箇所を検出するシステムを構築している。本稿では、撞球姿勢の推定法とその精度に関する評価について述べる。

キーワード: ビリヤード, Kinect, 画像処理, 3次元位置推定, 姿勢判定

Development of Billiard's Posture Detection System with Kinect

MASASHI MISHIMA[†] AKIRA SUGANUMA[†]

Abstract: players who wide range of ages can enjoy Billiards. However, billiard beginners cannot hit the cue ball as player wants due to be difficult to get a proper position. In this research, I get the three-dimensional position of shoulder, elbow, wrist, and waist by attaching color markers at seven places on the player's cloth and photographing it with Kinect. Afterward, I develop a system that detects inappropriate parts as a billiard posture. We describe estimation method of billiard posture and evaluation on its accuracy in this paper.

Keywords: Billiard, Kinect, Image processing, 3D position estimation, Posture detection

1. はじめに

ビリヤードは、室内で行われるメンタルスポーツ競技であり、子供から年配者まで幅広い年齢層のプレイヤーが楽しむことができる。その一方、ビリヤードには手球を狙った所へ運ぶ正確な撞球技術、ゲームを有利に進めていくための戦術など、技術面と知識面の両方を習得しなければならないため、初心者にとって難しいスポーツでもある。

我々はビリヤード初級者へ向けて、技術向上のための支援システムを開発している。これまでに、“手球をどこへ狙って撞けば良いのか分からないプレイヤーへ向けて、撞球する手球の軌道を予測し、その軌道を台上にプロジェクタで投影させることで、的球をポケットへ入れ易くする支援システム”や“手球と的球の間に別の球が存在し、クッションを使わなければ当てる事が困難な場合に的球をポケットへ入れる軌道を表示する支援システム”などを開発した[1][2]。そして、これらのシステムは、知識面で支援するシステムである。

確かに知識も経験も少ない初級者にとって、経験者と共にゲームをするためには、狙う方向を示し、ミスショットを防ぐための支援システムは必要である。その一方、知識面だけを支援するシステムでは、初級者支援は完成しない。なぜなら、今までのシステムはプレイヤーが正確な撞球技術を習得していることを前提に開発しているからである。

よって、狙った所へ手球を撞くことができない初級者へ向けた撞球姿勢判定システムを開発する。

手球を正確に撞くためには、正しい撞球姿勢を習得する必要がある。そして、初めは、プロのプレイヤーから教えてもらい、何度も不適切な箇所を指摘されながら修正し、正しい姿勢を身につけていく事が理想的な習得方法である。

しかし、多くの初級者はプロから学ぶことはできないであろうし、中上級者と一緒にプレイするまでに誤った癖がついてしまうこともあるだろう。よって、初級者が正しい姿勢を習得するまでには、長い時間が必要となる。それでは、ビリヤードの楽しさを体験する前に飽きてしまう。従って、指導者がいない初級者のために不適切な箇所を指摘するシステムが必要となる。

そして、我々は撞球姿勢の判定に以下の3項目を設ける

- 後方の上腕と前腕は直角に近い角度で交わる。
- 前方側（レスト側）の腕は真直ぐ伸びる。
- 後方側の脇は引き締める

更に、項目毎にその姿勢が撞球するために、適切か不適切か判定する閾値を取得するために、複数のプレイヤーを対象に計測を行う。これは、プレイヤー毎に体格や利き手など異なるため、多くのプレイヤーを対象としたシステムを開発するには、複数のプレイヤーからデータを取得することが必要になるからである。

最後に、プレイヤーに正しい姿勢や間違いの姿勢を構えて

[†] 有明工業高等専門学校
National institute of technology Ariake college.

もらい、判定精度を確かめ、現状の実用性を考察する。

2. 関節点の検出と3次元座標の取得

我々は、姿勢を推定するために、7ヶ所の関節位置（両肩、両肘、両手首、腰）にマーカを巻き、色検出することで検出した。各関節を検出した画像を図1に示す。そして、検出したマーカの領域座標から重心を求め、その重心を関節点と決めた。

次に、上腕と前腕が交わる角度などを算出するためがあるため、Kinectを使用し、関節点の三次元ワールド座標を取得した。また、Kinectには、各関節点を検出する機能が元々あるが、Kinectの正面に立ち、腕を前へ突き出した場合では、うまく各関節を検出できない。よって、より精度の高い検出を行うために、マーカを巻き、色検出によって特定する。

Kinectには、カメラと奥行きセンサが搭載されており、関節点の三次元ワールド座標を取得する事が可能になるが、同じ位置から撮影することが出来ないため、図2に示すように画像間で座標のずれが生じる。

よって、OpenCVのライブラリ関数を使用し、チェッカーパターンから特徴点を見つけ、その座標を使いセンサの内部・外部パラメータを算出する。そして、カメラキャリブレーションを行うと、図3に示すように座標のずれを解消できる。

位置合わせ後、関節点の三次元座標を取得すると、縦と横は画像上の座標、奥行きは奥行き値[m]であった。したがって、線形代数の式に従い肘の角度など計算するためには、縦、横をワールド座標に変換し、奥行き値と同じ単位に統一しなければならない。よって、対象物のワールド座標を画像平面上の座標に変換する射影投影の式を逆変換することにより、関節点の三次元ワールド座標を取得する。式(1)に変換式を示す。式中の M' は三次元ワールド座標を表し、 m' は画像平面に投影された点の座標を表す。また、 A はカメラの内部パラメータ行列、 $[R|t]$ は外部パラメータ行列と呼ぶ。

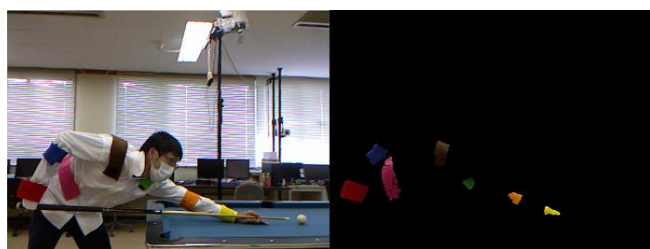


図1 各関節位置の検出

Figure 1 Detecting of all joint-positons

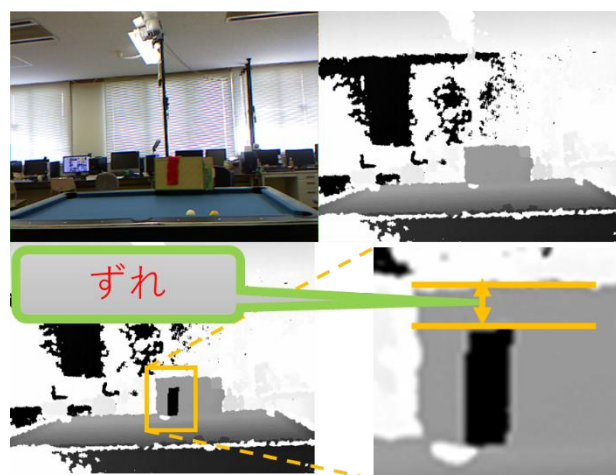


図2 センサ間の座標のずれ

Figure 2 Gap of coordinates between two-sensors

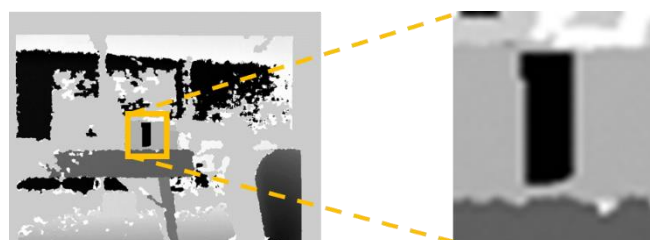


図3 座標の位置合わせ

Figure 3 Alignment of coordinates

$$sm' = A[R|t]M'$$

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \cdot \cdot \cdot (1)$$

3. 姿勢の推定と判定

我々のシステムは、プレイヤーの撞球姿勢を判定するために、以下の3つの項目に着目する。

- 後方の上腕と前腕は直角に近い角度で交わる。
- 前方側（レスト側）の腕は真直ぐ伸びる。
- 後方側の脇は引き締める

これらの判定を行うために、2つのベクトルがなす角度を算出する式(2)や2点間のユークリッド距離の式を利用する。

$$\cos \theta = \frac{A_x \cdot B_x + A_y \cdot B_y + A_z \cdot B_z}{\sqrt{A_x^2 + A_y^2 + A_z^2} \cdot \sqrt{B_x^2 + B_y^2 + B_z^2}} \cdot \cdot \cdot (2)$$

ここで、 $\cos \theta$ は算出する角度のコサイン値であり、変数 A_x , A_y , A_z はベクトルAの、変数 B_x , B_y , B_z はベクトルBのそれぞれx方向成分、y方向成分、z方向成分を意味する。

更に、我々のシステムは色検出によって関節位置を検出できるが、検出した関節が前方側、または、後方側か判断できない。したがって、判定を行う度にユーザが前方側の手首や後方側の肘がどれであるかシステムに教えなければ

判定ができない。

よって、我々は判定精度を確認するよりも先に、検出した関節が前方側か後方側か判断する機能を追加する。その後、撞球姿勢の判定を行い、プレイヤーに不適切な箇所を教えるなどといった支援を行う。

3.1 前方または後方の判別

この機能を実装するために、我々は手球に着目する。手球とはプレイヤーがキュースティックで撞く球である。

撞球時、手球は必ずキュースティックの先端の先にあり、必ず白色である。よって、手球から左右の手首までの距離を計測し、距離が短い方を前方側の手首であると判断する。

また、我々のシステムは図4に示すように、検出した各関節に番号を割り当てており、レスト側の手首を特定することができれば、割り当てられた番号から前方の腕の肘と肩を特定することができる。また、同様に後方の手首、肘、肩も特定することができる。

手球を検出する手法は、関節の位置につけたマーカを色検出によって検出し、関節点の三次元ワールド座標を取得した場合と同じである。したがって、我々は、図5に示すように手球を検出できた。

そして、手球の座標取得後、我々は手球から左右の手首までの距離をユークリッドの距離の式に従い計測し、より手球に近い手首が前方側であると判定する。



図4 各関節を番号管理

Figure 4 Number management for specifying joints



図5 手球の検出

Figure 5 Detecting the cue ball

3.2 後方側の腕に関する条件

キュースティックが手球を捕らえる瞬間をインパクトと呼ぶ。そして、キュースティックの力を効率よく手球に伝えるためには、インパクトの瞬間、肘をほぼ直角にしていることが理想である。また、インパクト時の肘の角度はプレイヤーが構えたときの肘の角度とほぼ一致する。よって、

我々は、プレイヤーが構えた際の姿勢で後方の腕の上腕と前腕が交わる角度を算出し、肘が直角になっているかを確認する。

肘の角度を計算するためには、図6に示すように、検出した後方の肘と肩、肘と手首の関節点を結び、2つのベクトルを求める。そして、これらの2つのベクトルがなす角度を式(2)に従い算出する。

3.3 前方側の肘に関する条件

前方側の腕を真直ぐ伸ばして撞球することで、身体が前に出ようとする事を防ぐことができる。もし体が前のめりになった状態でストロークを行うと、後方の肘を支点にキュースティックをストロークすることができない場合やストロークが完了する前に身体が起き上がり、手球を撞き損ねる場合がある。よって、撞球姿勢を判定するためには、レスト側の腕が伸びている事を確認しなければならない。

我々は、この項目を判定するため、図7に示すように検出した前方の肩と肘、肩と手首の関節点を結び、2つのベクトルを求める。これらのベクトルが同じ方向を向いていると肘が伸びていると判定できる。2つのベクトルが同じ方向を向いているかの判定は、式(2)を用いて2つのベクトルがなす角度を算出する。

最初、我々はこの項目の判定に、検出した前方の肘と肩、肘と手首を結びベクトルを求めて、これらの2つのベクトルがなす角度が180度ならば適切であると判定しようとした。しかし、図8に示すように、例え前方の腕が真直ぐ伸びていても、検出した関節点によっては算出する角度が180度でない場合がある。このため、検出する関節点によって、算出される角度にばらつきが出ないように、前方の肩と肘、肩と手首の関節点を結んだ2つのベクトルがなす角度を採用した。

3.4 後方側の脇に関する条件

キュースティックを手球に向けて真直ぐに動かせるようになると、手球の撞き損ねを防ぐことができる。そのためには、キュースティックが肘の真下に位置するよう構えることが必要になる。つまり、脇を締めることで、キュースティックが横方向に動きにくくなる。

我々は、この項目を判定するため、図9に示すように検出した後方の手首の関節点と腰の位置との距離をユークリッド距離の式に従い算出する。

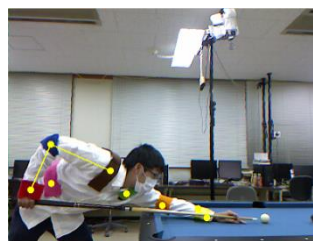


図6 肘の角度の計算

Figure 6 Calculate the angle of elbow

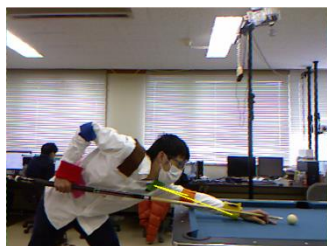


図 7 腕の伸びの判定

Figure 7 Judgement of arm stretching



図 8 肘の角度による腕の伸び判定

Figure 8 Judgement of arm stretching by the angle of elbow



図 9 脇の締め判定

Figure 9 Judgement of fastening aside

4. 閾値の取得

我々のシステムは、関節点を結んだベクトルがなす角度や手首から体までの距離を計測することで、撞球する姿勢が適切であるかを判定する。測定には誤差が含まれたり、プレイヤーの体格などに起因する差などが含まれたりするので、算出する角度や距離がある範囲内に入るときに適切な姿勢であると判定する手法を採る。適切な範囲を得るために、予め正しい撞球姿勢を習得した 10 人のプレイヤーに構えてもらい、1 項目につき 1 人当たり 200 回測定を行った。プレイ中は様々な撞球状況があることを想定して、プレイヤーには狙う方向や撞球する位置を変えてもらい撮影した。

4.1 後方側の肘に関する条件

後方側の肘の曲げ具合に関する閾値を取得するため、10 人のプレイヤーに、後方の肘の角度が 90 度になるように構えてもらい、実験を行った。システムで後方の肩、肘、腰の関節点を検出して肘の角度を計算した。このようにして測定した 2000 個のデータをまとめたグラフを図 10 に示す。このグラフの横軸は算出した肘の角度のコサイン値であり、

縦軸は算出した角度の個数を示している。

このヒストグラムは、平均値を含むバーが最も高く、平均値から離れる程バーが低くなる。このデータから生起確率プロット (Q-Q プロット) を作成すると図 11 のようになる。このプロットのようにデータがほぼ一直線上に並ぶ。さらに p 値を算出すると、0.05 よりも大きくなったため、ガウス分布に従っていると考えられる。

よって、我々はデータの平均値と標準偏差を求め、平均値 $\pm$ (標準偏差 $\times 2$) で得られる範囲を、後方の肘が適切であると判断する範囲とした。この範囲は 0.23 から -0.32 であり、肘の角度にすると約 77 度から 108 度となる。

4.2 前方側の肘に関する条件

前方側の肘が曲がっていないか判定する閾値を取得するため、10 人のプレイヤーに、前方の腕を真っ直ぐ伸ばして構えてもらい、実験を行った。前方の肩、肘、腰の関節点を検出し、肩から肘、肩から手首を結んだ 2 つのベクトルがなす角度を計算した。このようにして測定した 2000 個のデータをまとめたグラフを図 12 に示す。このグラフの横軸は算出した肘の角度のコサイン値であり、縦軸は算出した角度の個数を示している。

図 12 のヒストグラムは図 10 のヒストグラムと比べても明確だが、図 13 に示す生起確率プロットの点が直線になっていない。また、 p 値も 0.05 よりも小さい値となったため、ガウス分布に従っていないと考える。よって、他の方法で閾値を取得する。

我々は、この判定項目に対して、判別分析法を用いて適切と判定する範囲を取得する。その結果、その範囲は 0.94 から 1.00 であり、角度にすると 0 度から 20 度となる。これは、取得したデータの約 90% を含んでいる。

4.3 後方側の脇に関する条件

我々は、後方側の脇に関する条件で判定する閾値を取得するため、10 人のプレイヤーに、後方の脇を締めることを意識して構えてもらい、実験を行った。後方の手首の関節点と腰の位置を検出し、その二点の距離を計算した。このようにして測定した 2000 個のデータをまとめたグラフを図 14 に示す。このグラフの横軸は算出した二点の距離であり、縦軸は算出した距離の個数を示している。

このヒストグラムは、平均値を含むバーが最も高く、平均値から離れる程バーが低くなり、図 15 に示す 0 [m] から 0.25 [m] までの範囲の生起確率プロットの点は、ほぼ近似式上に一直線となっている。また、 p 値も 0.05 より大きくなったため、ガウス分布に従っていると考えられる。

今回も後方側の肘のときと同様に、データの平均値と標準偏差を求め、平均値 $\pm$ (標準偏差 $\times 2$) で得られる範囲を、脇を締めていると判断する範囲とした。この範囲は 0.05 [m] から 0.20 [m] であり、取得したデータの約 95% を含む範囲である。

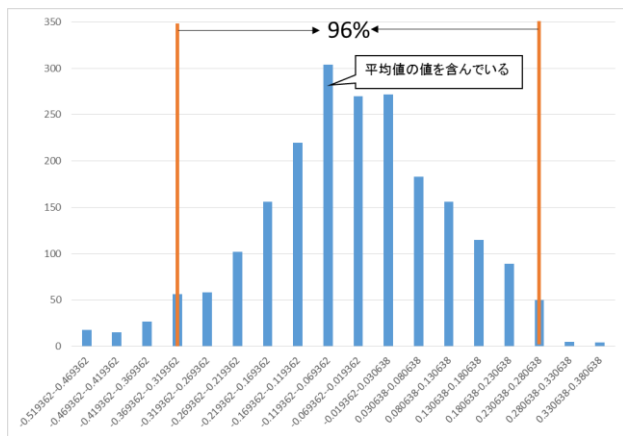


図 10 閾値の取得 (肘の角度)

Figure 10 Getting the threshold (The angle of elbow)

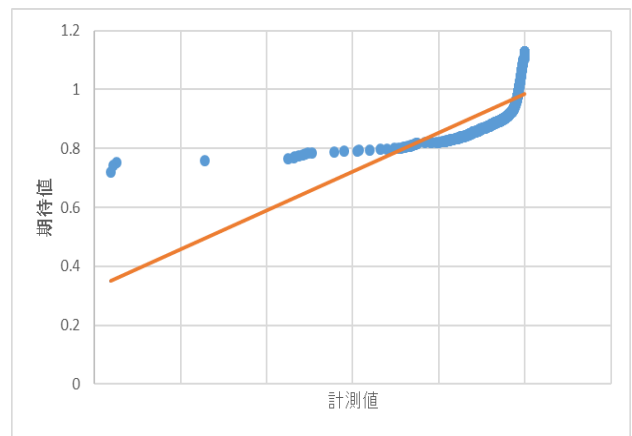


図 13 正規確率プロット (腕の伸び)

Figure 13 Normal Probability Plot (The stretching arm)

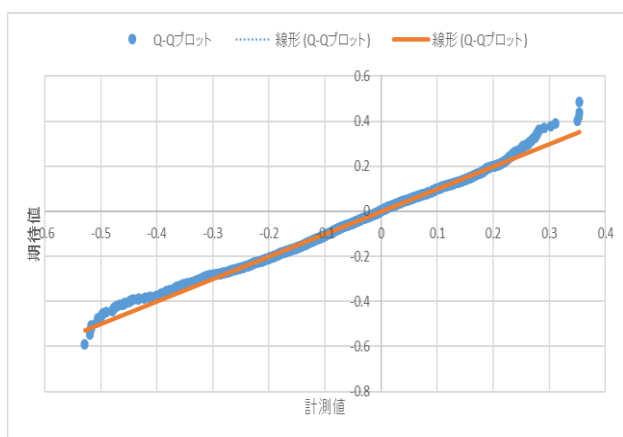


図 11 正規確率プロット (肘の角度)

Figure 11 Normal Probability Plot (The angle of elbow)

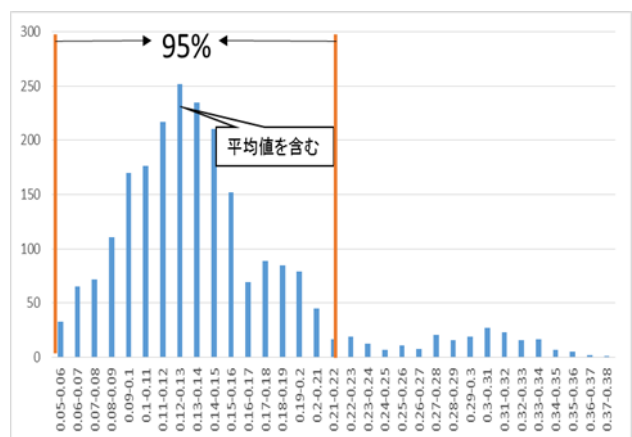


図 14 閾値の取得 (脇の締め)

Figure 14 Getting the threshold (The fastening aside)

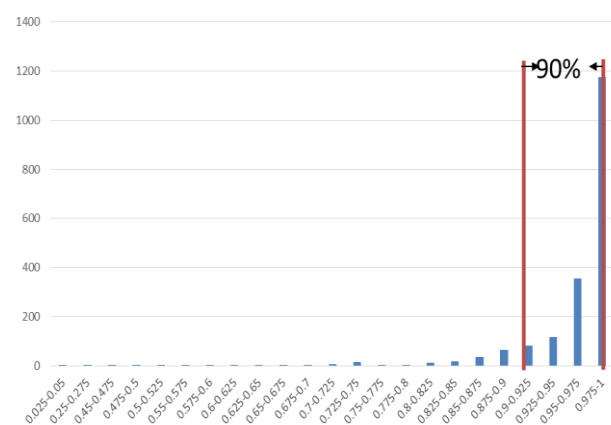


図 12 閾値の取得 (腕の伸び)

Figure 12 Getting the threshold (The stretching arm)

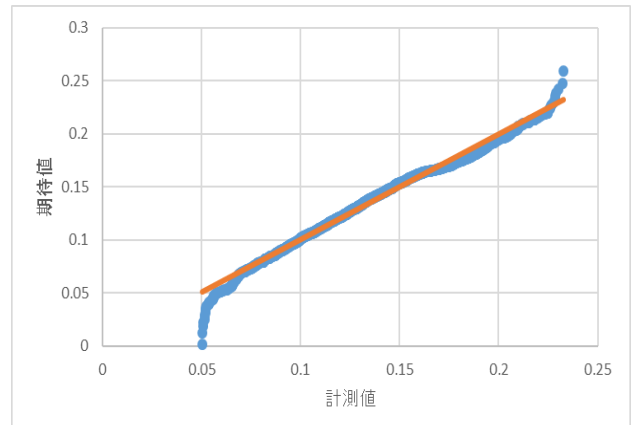


図 15 正規確率プロット

Figure 15 Normal Probability Plot (The fastening aside)

5. 撞球姿勢判定の評価

撞球姿勢の判定の精度を測定するために、予め姿勢に関するシナリオを作成し、被験者にそのシナリオ通りに構ってもらって評価実験を行った。シナリオの内容は、正しい撞球姿勢で構えた場合や後方の肘を鋭角（鈍角）に曲げて

構えた撞球姿勢などの様々なパターンを組み合わせたものである。また、今回使用する Kinect は 1 台であるため、プレイヤーはすべてのマークを撮影できる位置で構えてもらった。指定した姿勢で構えたときのシステムの判定結果を集め、判定精度を調べた。測定回数は被験者 1 人当たり 100 回である。

5.1 撞球姿勢全体での評価

最初に、撞球姿勢全体で適切・不適切を判定した場合の判定精度を調査した。その結果、我々のシステムは、プレイヤーの姿勢を 97% の精度で、撞球姿勢として適切か不適切かを判定できることがわかった。

このことから、現時点のシステムは、姿勢全体を判定する場合、高い信頼性があるのではないかと推測する。なぜなら、撞球姿勢として、適切な姿勢をとった場合、約 97.5% の精度で正しく判断できており、不適切な姿勢をとった場合もまた、約 96.5% の精度で不適切と判定することができた。従って、姿勢全体の適切・不適切の判定は現段階のシステムでも可能であると考えられる。

5.2 項目ごとの評価

判定に使用している 3 つの項目ごとに評価を行った。各項目の判定精度は姿勢全体を判定したときの精度よりも低い結果を得た。まず、肘の角度の判定では 93.6%、レスト側の腕の伸びの判定では 94% であった。これらは、9 割を超える精度であったが、脇の判定では 86.0% の精度と低い値であることがわかった。

レスト側の腕が真っ直ぐ伸びているか判定する場合、10 人中 9 人のプレイヤーでは 94% 以上、中には 100% の精度で正しく判定できている。しかし、1 人のプレイヤーに対しては、判定精度がとても悪かった。原因として、図 16 に示すように、プレイヤーによっては、左肩のマークの大部分が顔に隠れてしまい、検出した肩の関節位置が実際の関節位置から少しずれてしまう。その結果、算出する角度に影響が出てしまったのではないかと考える。この影響をなるべく小さくするためには、Kinect を 2 台使用し、前方側と後方側を分けて撮影することが必要であると考えられる。

また、後方の肘と脇の判定が同時に誤判定してしまう場合がある。特に、脇の締め判定は最も精度が悪く、この項目に関しては異なる方法も組み合わせで判定したほうがよいと考える。

3 つの項目を個別に判定する場合は精度が少し落ちる。そのため、適切な箇所であっても、不適切と判定する場合がある。評価の結果では、約 80% の場合では正しく判定できるが、プレイヤーに間違えた情報を与える恐れがあることを意味する。そのため、各項目に着目して指摘するためには、さらなる改善が必要であると考えられる。

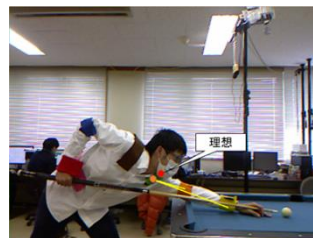


図 16 検出する関節点が及ぼす計算値のずれ

Figure 16 the effects of detected joint on deviation of calculated value

6. おわりに

我々は、熟練者と初級者の技術的な差を少しでも埋めて、誰もがビリヤードを楽しめるよう初級者の撞球姿勢判定を行うシステムの開発を目標としてきた。プレイヤーの左右の腕と脇に注目して撞球姿勢が適切であるかどうかを判定する手法を構築した。判定手法は両腕に関する 3 つの項目からなっている。

撞球姿勢をとるプレイヤーを Kinect で撮影して映像と距離画像からプレイヤーの関節位置を検出し、適切な姿勢かどうかを判定するシステムを構築した。検出する関節は両手首、両肘、両肩、そして腰の 7 ヶ所であり、更にそれぞれ関節位置の三次元座標を取得する。次に、判定の際に必要な項目毎の閾値を取得するため、正しい撞球姿勢を予め習得した 10 人の被験者に 1 人当たり 200 回撞球姿勢を構えてもらいシステムで計測した。このデータを用いて、判定条件を構築した。

最後に、本研究の目標であるシステムがプレイヤーに誤り箇所を指摘できるか評価を行った。姿勢全体の判定に関しては、95.6% の高い精度で判断が可能である。更に、各項目の判定精度を見ると、レスト側の腕の伸び判定では 94.0%、後方の肘の角度が直角に近い角度で構えているかの判定では 93.6%、そして、脇の締め判定では 86% の精度であった。この結果、我々のシステムは姿勢全体での判定は現在でも実用に耐えらると思える。

我々のシステムをさらによくするために、以下のような今後の課題が残されている。

(1) 新たな判定方法の検証

今回、撞球時に脇が締まっていることを判定するために、後方の手首と腰との距離を利用した。しかし、判定制度はあまりよくなかった。そのため、新たな判定方法が必要であると思える。

(2) Kinect を 2 台使った姿勢判定

本研究では、Kinect を 1 台だけ使用して姿勢判定を行った。そのため、我々のシステムを使うためには、Kinect に全箇所の関節が写る位置からの撞球だけが許され、毎回 Kinect を移動させなければならず、制限の厳しい物であった。また、1 台ですべての関節を検出しなければならないため、マークが隠れることもあった。よって、今後は Kinect

をもう1台増やして、被験者の正面と背面を撮影することで、正面で撮影したときに顔に隠れてしまうこともあった肩のマーカーを背面から撮影することで、検出精度を向上させることができると考える。

(3) 適切な姿勢へのアドバイス

現在のシステムには、誤り箇所を発見するまでの機能しか持っていない。よって、正しい撞球姿勢にするためにはどのように修正すべきかをプレイヤーへアドバイスする機能を追加する。

参考文献

- [1] N. Watanabe and A. Suganuma: “Development of Billiards Support System for Beginners by Image Processing,” International Technical Conference on Circuits/Systems, Computers and Communications, 2013.
- [2] F. Takata and A. Suganuma: “A Flicker-less Method Detecting the Cue Stick for a System Predicting the Path of the Billiard balls,” Proc. of the 30th International Technical Conference on Circuits/Systems, Computers and Communications, 2015.
- [3] H. Uchiyama and H. Saito, “AR Supporting System for Pool Games Using a Camera-Mounted Handheld Display,” Advances in Human-Computer Interaction, vol. 2008, Article ID 357270, 13 pages, 2008. doi:10.1155/2008/357270
- [4] Jack H. Koehler: “THE SCIENCE OF POCKET BILLIARDS” sportology publications the United States of America, 1995.
- [5] J. Shotton, A. Fitzgibbon, A. Blake, A. Kipman, M. Finocchio, R. Moore, T. Sharp, & “Real-Time Human Pose Recognition in Parts from a Single Depth Image, 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2011.
- [6] D. Herrera C., J. Kannala, J. Heikkila, & “Joint Depth and Color Camera Calibration with Distortion Correction,” IEEE Transactions on Pattern Analysis and Machine Intelligence Year: 2012, Volume: 34, Issue: 10, pp.2058 - 2064, 2012.
- [7] D. Herrera C., J. Kannala, J. Heikkila, & “Accurate and practical calibration of a depth and color camera pair,” CAIP 2011, Part II, LNCS 6855, pp. 437 - 445, 2011.
- [8] OpenCV programming book production team (Author): “OpenCV programming book.