

キムワイプ卓球の台の割り当てアルゴリズムの比較

小嶋明

東京大学キムワイプ卓球会

a.kojima8924@gmail.com

1 概要

東京大学キムワイプ卓球会は駒場祭においてキムワイプ卓球体験会を開催している。キムワイプ卓球体験会では複数の卓球台が用意され、会場に来たプレイヤーはキムワイプ卓球を行うことができる。プレイヤーは到着時に使用されていない台があり、他に待機しているプレイヤーがいない場合は即座にキムワイプ卓球を始めることができるが、全ての台が使用されている場合は台が解放されプレイヤーに割り当てられるまで待つ必要がある。キムワイプ卓球会員は必要に応じていくつかの台で行われるキムワイプ卓球を終了させて台を解放し、待機しているプレイヤーに空いている卓球台を割り当てる。台を割り当てる順序は到着順で行われてきたが、到着

順ではプレイ時間が長いプレイヤーの直後に到着したプレイヤーの待ち時間が増大し公平性が失われる場合がある。

本研究ではキムワイプ卓球体験会で待機しているプレイヤーに卓球台を割り当てる方法に CPU スケジューリングの手法を適用し、各手法の平均待ち時間と公平性を評価する。さらにキムワイプ卓球と CPU スケジューリングの差異を考慮し、キムワイプ卓球に適する台の割り当てアルゴリズムを考案する。

2.1 キムワイプ卓球の台の割り当てと CPU スケジューリング

キムワイプ卓球体験会では、キムワイプ卓球会員が待機しているプレイヤーの集合から次に台を割り当てる

プレイヤーを選択し、選択されたプレイヤーにキムワイプを配付して台に案内する。CPU スケジューリングでは、CPU スケジューラが実行可能状態にあるプロセスの集合から次に CPU を割り当てるプロセスを選択する。さらに、ディスパッチャが選択されたプロセスに CPU を割り当てる。このように、キムワイプ卓球の台の割り当てと CPU スケジューリングは類似しており、プレイヤーとプロセス、卓球台と CPU、キムワイプ卓球会員と CPU スケジューラ及びディスパッチャがそれぞれ対応する。

CPU スケジューリングには、横取り可能な(preemptive)スケジューリングアルゴリズムと横取り不可能な(non-preemptive)スケジューリングアルゴリズムがある。横取り可能なアルゴリズムでは、実行中のプロセスから CPU を奪い、他のプロセスに割り当てることができる。同様にキムワイプ卓球の台の割り当てでも横取り可能な割り当てアルゴリズムと横取り不可能な割り当てアルゴリズムを考

えることができる。横取り可能なアルゴリズムではキムワイプ卓球会員が実行中のキムワイプ卓球を中断させ、台を他のプレイヤーに割り当てることができる。

2.2 CPU スケジューリングの諸手法

前述したようにキムワイプ卓球会員の台の割り当てと CPU スケジューラの CPU の割り当てが対応するため、後者の手法を前者に応用することができる。代表的なスケジューリングアルゴリズムとして到着順(FCFS, FIFO)スケジューリング、最短時間順(SJF)スケジューリング、優先度(priority)スケジューリング、ラウンドロビン(RR)スケジューリングを挙げる。

2.2.1 到着順スケジューリング

到着順スケジューリングでは、実行可能状態となったプロセスは FIFO（先入れ先出し）キューに入れられ、キューの先頭のプロセスから実行さ

れる。この方法では、処理時間の長いプロセスが先に到着すると平均待ち時間が増加する。キムワイプ卓球でも同様に、プレイ時間の長いプレイヤーが先に到着すると平均待ち時間が長くなる。

2.2.2 最短時間順スケジューリング

最短時間順スケジューリングでは、最も CPU 処理時間の短いプロセスに対し CPU が割り当てられる。横取り不可能の場合は CPU を割り当てられたプロセスは処理が終わるまで横取りされることはない。横取り可能の場合は実行中のプロセスの残りの処理時間より小さい処理時間を持つ実行可能プロセスがあるとき、処理時間の小さいプロセスに CPU が再割り当てされる。この方法では各プレイヤーのプレイ時間が変更されない場合の平均待ち時間が最小化されるが、実際の CPU スケジューリングでは各プロセスの CPU 処理時間を厳密に知ることはできないので処理時間の推定値が

用いられる。キムワイプ卓球では待機しているプレイヤーに希望するプレイ時間を尋ねることができるため、厳密な最小時間順の割り当てができる。

2.2.3 優先度スケジューリング

優先度スケジューリングでは、各プロセスに優先度の値が定められ、優先度の最も高いプロセスが実行される。横取り不可能の場合は実行中の低優先度のプロセスは横取りされないが、横取り可能の場合は実行中のプロセスの優先度より高い優先度を持つ実行可能状態のプロセスがあるとき、CPU は後者に再割り当てされる。この方法では、低優先度のプロセスが後回しにされ無限に待たされてしまう問題（飢餓, starvation）がある。この対策として、待ち時間に応じて優先度を徐々に高くする方法（エージング, aging）がある。キムワイプ卓球では、優先度として各プレイヤーの組の構成人数や所属を優先度として定めることができる。

2.2.4 ラウンドロビンスケジュールリング

ラウンドロビンスケジュールリングでは、実行可能状態にあるプロセスはFIFO キューに入れられる。キューの先頭にあるプロセスは実行されるときにタイムクォンタムと呼ばれる一定の処理時間が与えられ、処理時間がタイムクォンタムに達するとプロセスは中断されキューの最後尾に移動する。タイムクォンタムが十分小さいとき CPU は各プロセスに均等に処理能力を与えているとみなすことができるが、プロセスの切り替えのオーバーヘッドは増大する。一方、タイムクォンタムが十分大きい場合は到着順スケジュールリングと同じとなる。キムワイプ卓球では、タイムクォンタムを10分としてラウンドロビン方式で行われたことがある。

2.3 キムワイプ卓球と CPU スケジュールリングの相違点

前述したように、CPU スケジュールリングではプロセスの処理時間を正確

に知ることはできないが、キムワイプ卓球ではプレイヤーにプレイ時間を尋ねることができる。そのためキムワイプ卓球では最短時間順のスケジュールリングを行うことが容易である。しかしキムワイプ卓球のプレイヤーはプレイ時間を自由に決めることができるので、台の割り当て順序が最短時間順となっているとき、プレイヤーは短いプレイ時間のキムワイプ卓球を複数回行うことを選択する場合がある。例えば多数のプレイヤーが20分程度のプレイ時間のキムワイプ卓球を行っているとき、合計30分間キムワイプ卓球を行うことを望む合理的なプレイヤーは、プレイ時間が30分のキムワイプ卓球を1回行うのではなくプレイ時間が15分のキムワイプ卓球を2回行うことを選択すると考えられる。このようにプレイヤーが自主的に複数回のキムワイプ卓球を選択すると、合計のプレイ時間は等しいにもかかわらず台の切り替えが多くなりオーバーヘッドが増大する。さらに、

長いプレイ時間を選択したプレイヤーの待ち時間が非常に大きくなる。

その上、プレイヤーが複数回並び直して台の再割り当てを受けることを煩わしく思い、最短時間順やラウンドロビン方式で並び直しが必要とされる場合に本来の希望するプレイ時間より短いプレイ時間でキムワイプ卓球を中断する可能性がある。

2.4 公平性

キムワイプ卓球では、横取り不可能かつプレイヤーが自主的に複数回のプレイを選択しない割り当てアルゴリズムが好ましい。最短時間順の割り当てはプレイヤーがプレイ時間を変更しない場合の平均待ち時間が最小化されるが、実際はプレイヤーが複数回のキムワイプ卓球を選択し台の割り当てのオーバーヘッドが増大する。これを防ぐためには、待ち時間がプレイ時間と比例するように台の割り当てを行えば良い。ここで*i*番目のプレイヤーのプレイ時間を $t_{p,i}$ 、待ち時間を

$t_{w,i}$ として、プレイ時間に対する待ち時間の比を

$$r_i = t_{w,i}/t_{p,i}$$

と定める。 $t_{p,i}$ の大小にかかわらず r_i が一定の値となれば、プレイヤーの合計待ち時間は合計プレイ時間に比例し、キムワイプ卓球の回数に依存しない。ここで、 r_i の値のばらつきの小ささを公平性と定義する。公平性が最大するとき r_i は一定である。

2.5 公平性の高い割り当てアルゴリズム

公平性を最大化するには、 r_i の分散、標準偏差、最大値と最小値の差などを最小化するアルゴリズムを採用すればよい。しかし、最大値と最小値の差などで評価すると、最小値を大きくすることで公平性が最大化される場合がある。すなわち、公平性のためにそれぞれの r_i が大きくなり、平均待ち時間が増大する場合がある。本研究では平均待ち時間をと計算の簡略化のために、 r_i の最大値の最小化を目標とする。 r_i は待ち時間に比例して増加する

ので、最大値を抑えることで公平性は向上する。

公平性を高めるアルゴリズムを3種類挙げる。以降では台の数を c 、待機しているプレイヤーの人数を n とする。

2.5.1 貪欲法

r_i の最大値の最小化の方法として貪欲法による近似アルゴリズムがある。貪欲法では台が空いたときに待機している全てのプレイヤーに対しプレイ時間に対する待ち時間の比 r_i ($i = 1, 2, \dots, n$)を計算し、最も値が大きいプレイヤーに台を割り当てる。計算量は $O(n)$ である。

台が空いた時間を $t = 0$ として横軸を時間 t 、縦軸を比 r としたグラフを考えると、各プレイヤーの t, r の関係は一次関数で表される。 t 切片が最小であるプレイヤーを選ぶ方法が到着順、傾きが最大であるプレイヤーを選ぶ方法が最短時間順、 r 切片が最大であるプレイヤーを選ぶのが貪欲法である。

2.5.2 台が空く時刻を推定する方法

この方法では、待機している n 人のプレイヤーに台を割り当てる順序の中で、各プレイヤーに台が割り当てられた時刻での r_i ($i = 1, 2, \dots, n$)の最大値が最小であるものを探す。 n 人のプレイヤーに台を割り当てる順序は $n!$ 通りあり全て列挙して計算するのは n が大きいとき困難であるので、台が空く時刻をあらかじめ推定し、プレイヤーに台を割り当てる順序によって台が空く時刻が変化しないと仮定して考える。すなわち、待機しているプレイヤーに割り当てられた台が空くことによって台が空く時刻が変わることを無視して考える。

それぞれの台にいるプレイヤーに台が割り当てられた時刻とプレイ時間が記録されている場合は、台が空く時刻の最初の c 回を予測することができる。台にいるプレイヤーの情報が記録されていない場合や c 回目より後に台が空く時刻は、プレイヤーのプレイ時間の平均値から推測する。プレイ時

間の平均値を $\bar{t}_p$ 、台の割り当てにかかる時間を d とすると、台が空く時間の間隔はおよそ $(\bar{t}_p + d)/c$ となる。このようにして台が空く時刻を n 回推定する。

i 番目のプレイヤーが j 回目に台が空いたときに台を割り当てられた場合のプレイ時間に対する待ち時間の比を $r_{i,j}$ ($i, j = 1, 2, \dots, n$)としてそれらを計算すると、 n^2 個の値が得られる。ここで、 $r_{i,j}$ を昇順でソートしたとき k 番目の $r_{i,j}, i, j$ の値をそれぞれ R_k, I_k, J_k ($k = 1, 2, \dots, n^2$)とする。また、ある k, i について、 $R_k \geq r_{i,j}$ であるような $r_{i,j}$ の j の最大値を $M_{k,i}$ （ただし条件を満たす $r_{i,j}$ が存在しないとき0）とすると、

$M_{k,i} = \max\{J_l \mid l \leq k \text{ かつ } I_l = i\} \cup \{0\}$ となる。すると、ある k について、

$\forall j \leq n \quad |\{M_{k,i} \mid M_{k,i} \geq j\}| > n - j$
 $\Leftrightarrow$ 最小化された r_i の最大値が R_k 以下となる。言い換えると、 R_k より小さい r_i の値で j 回目に台の割り当てを受けることのできるプレイヤーの人数が $(n - j + 1)$ 人以上であれば r_i の最大値

が R_k 以下となるようにプレイヤーの順序を決めることができる。上記の条件を満たす最小の k は二分探索で求めることができ、 r_i の最小化された最大値とその場合に台の割り当てを受けるプレイヤーおよび時刻が決定される。すでに割り当ての時刻が決定されたプレイヤーを除いて二分探索を n 回繰り返すことで、全てのプレイヤーの台の割り当て順序を決定できる。計算量は $O(n^2 \log(n))$ である。

2.5.3 厳密解を求めるアルゴリズム

待機している n 人のプレイヤーの台の割り当て順序について、台が空く時刻の推定をせずに厳密に考える。それぞれの台にいるプレイヤーに台が割り当てられた時刻とプレイ時間、待機しているプレイヤーが到着した時間とプレイ時間、待機しているプレイヤーが台の割り当てを受ける順序の全てが分かっているとき、待機しているプレイヤーが台の割り当てを受け

る時刻とその場合の r_i が計算でき、計算量は $O(n)$ である。

n 人のプレイヤーの台の割り当て順序は $n!$ 通りである。しかし、待機しているプレイヤーの中の2人のプレイヤーA, B について到着順と最短時間順が一致するとき、すなわちAの到着した時刻がBの到着した時刻より早く、Aのプレイ時間がBのプレイ時間より短いとき、Aは必ずBより早く台の割り当てを受ける。これはBがAより早く割り当てを受けた場合の r_i の最大値がAが先に割り当てを受けた場合の r_i の最大値以上になるためである。ここで各プレイヤーを頂点として、BよりAの方が到着した時刻が早く、プレイ時間が短いという条件を満たす全てのプレイヤーA, Bの組についてAからBに有向辺を置くと、グラフは有向非巡回グラフ（DAG）となる。台の割り当て順序としてあり得る順序はグラフがトポロジカルソート（各頂点を各有向辺が示す順序を満たしながら並べること）された順序である。

各プレイヤーのプレイ時間をランダムとして、 $(n-1)$ 人のプレイヤーがトポロジカルソートされた順序に対し、最も到着順が遅い n 番目のプレイヤーを挿入する場合の可能な挿入箇所の平均は $(n+1)/2$ 個以下であるので、 n 人のプレイヤーがトポロジカルソートされた順序の数の平均は $(n+1)!/2^n$ 以下である。よって、トポロジカルソートされた順序の全てに対して r_i の最大値を計算し、 r_i の最大値を最小化するプレイヤーの順序を探す計算の平均計算量は $O(n \cdot (n+1)!/2^n)$ 程度となる。

2.6 割り当て方法の平均待ち時間の評価

c 台の台の全てをプレイヤーが使用しており、 n 人のプレイヤーが待機している状況で最後に到着したプレイヤーAの平均待ち時間を計算する。プレイヤー全体の平均プレイ時間を $\bar{t}_p$ 、Aのプレイ時間を t_p 、1回の台の割り当てにかかる時間を d とする。また、平

均 $(\bar{t}_p + d)/c$ の時間間隔で 1 人のプレイヤーが到着すると仮定する。

到着順では、A がプレイを終えるまでに n 人の待機しているプレイヤーと c 人の台についているプレイヤーがプレイを終える必要があり、 c 人のプレイヤーは平均して $\bar{t}_p/2$ のプレイ時間を残しているので、合計でおよそ $(n + \frac{c}{2})\bar{t}_p$ の時間だけ台が使用される必要がある。台は c 台あり、台が使用されている時間と台の割り当てのために使用されていない時間の比は $\bar{t}_p : d$ であるので A が到着してからプレイを終えるまでにかかる時間は

$$\frac{n + c/2}{c} \cdot \frac{\bar{t}_p + d}{\bar{t}_p} \cdot \bar{t}_p$$

である。平均待ち時間は自分自身のプレイ時間 t_p を引き、

$$\frac{n + c/2}{c} \cdot \frac{\bar{t}_p + d}{\bar{t}_p} \cdot \bar{t}_p - t_p$$

となる。

最短時間順では、待機している n 人のプレイヤーをプレイ時間が短い順に並び替え、最もプレイ時間が短いプレイヤーを 0 番目としたとき A の順位を k ($k = 0, 1, \dots, n-1$)とすると、A

が台の割り当てを受ける前に最初に待機している n 人のプレイヤーのうち k 人が台の割り当てを受ける。 k 人の台の割り当てにかかる時間は $k(\bar{t}_p + d)/c$ 以下であり、この間に到着するプレイヤーの平均人数は k 人以下、その中でプレイ時間が A より小さいプレイヤーの平均人数は $k^2/(n-1)$ 以下である。同様に繰り返して、A が台の割り当てを受ける前に台の割り当てを受ける平均人数は

$$\sum_{i=0}^{\infty} k \left(\frac{k}{n-1} \right)^i = \frac{(n-1)k}{n-k-1}$$

以下であり、A の平均待ち時間は

$$\frac{\frac{(n-1)k}{n-k-1} + 1 + c/2}{c} \cdot \frac{\bar{t}_p + d}{\bar{t}_p} \cdot \bar{t}_p - t_p$$

以下である。特に、 $k/(n-1) = 1/2$ すなわち A のプレイ時間が最初に待機している n 人のプレイヤーの中央値であるとき、A の平均待ち時間は

$$\frac{n + c/2}{c} \cdot \frac{\bar{t}_p + d}{\bar{t}_p} \cdot \bar{t}_p - t_p$$

以下となり、到着順の平均待ち時間を超えない。平均待ち時間は k に対し単調増加する。

ラウンドロビン方式では、タイムクォンタムを q として、 $\bar{t}_p = q$ と仮定すると、台についているまたは待機しているプレイヤーの平均人数は $(n + c/2)$ 人となる。簡単のために c 台の台が $(n + c/2)$ 人のプレイヤーに処理能力を等しく配分しているとみなすと、Aの平均待ち時間は

$$\begin{aligned} & \frac{n + c/2}{c} \cdot \frac{q + d}{q} \cdot t_p - t_p \\ &= \left(\frac{n + c/2}{c} \cdot \frac{q + d}{q} - 1 \right) \cdot t_p \end{aligned}$$

と計算できる。平均待ち時間が t_p に比例しているため、公平性は最大となり、 $\bar{t}_p = t_p = q$ の場合、到着順の待ち時間と一致する。実際には q が小さいとき $(q + d)/q$ は大きくなり、台についているまたは待機しているプレイヤーの平均人数は $(n + c/2)$ 人より多くなるため平均待ち時間は増大する。一方 q が大きいとき台の処理能力が各プレイヤーに等しく配分されているとみなすことはできず、公平性は低下する。よって、平均待ち時間と公平性がトレードオフとなっている。

最後に、 r_i を評価する割り当て方法の平均待ち時間を計算する。 r_i のばらつきの小ささ（公平性）や平均値（平均待ち時間と比例する）は採用するアルゴリズムによって異なるが、公平性が最大つまり r_i の値が全て等しいと仮定する。台についているまたは待機しているプレイヤーの平均人数は $(n + c/2)$ 人であり、公平性が最大であることから各プレイヤーのプレイ時間に対する到着してからプレイを終えるまでの時間の比は一定で、台の割り当てにかかる時間を無視すると、 $r_i + 1 = (n + c/2)/c$ となる。よって、Aの平均待ち時間は

$$\begin{aligned} & \frac{n + c/2}{c} \cdot \frac{\bar{t}_p + d}{\bar{t}_p} \cdot t_p - t_p \\ &= \left(\frac{n + c/2}{c} \cdot \frac{\bar{t}_p + d}{\bar{t}_p} - 1 \right) \cdot t_p \end{aligned}$$

と計算できる。これもラウンドロビン方式と同様に平均待ち時間が t_p に比例し、公平性は最大となっているため仮定と一致する。この割り当て方法は横取り不可能であるので台の割り当て回数はラウンドロビン方式の回数以下である。よって $(\bar{t}_p + d)/\bar{t}_p$ は $(q +$

$d)/d$ 以下であるので、ラウンドロビン方式より平均待ち時間が小さい。

3. まとめ

キムワイプ卓球の台の割り当てにおいて、CPU スケジューリングの手法とプレイ時間に対する待ち時間の比によって評価する方法の計算量、平均待ち時間、公平性を評価した。キムワイプ卓球において実施が容易であるのは到着順、最短時間順、優先度順、ラウンドロビン方式、貪欲法による方法である。平均待ち時間に優れるのは横取り不可能の方法であり、プレイヤーが自主的に複数回のプレイをしない場合は最短時間順が優れる。公平性に優れるのはタイムクォンタムを小さくしたラウンドロビン方式とプレイ時間に対する待ち時間の比によって評価する方法である。

公平性が高い台の割り当て方法はキムワイプ卓球以外のサービスにも応用できる。キムワイプ卓球と同様にサービスを受ける時間と回数を客が決めることができ、サービスが無償で

提供され、混雑が長く続くようなサービスに対して応用しやすい。

キムワイプ卓球体験会は実施された回数が少なく、その台の割り当て方法はほとんどが到着順で行われた。今後のキムワイプ卓球体験会やその他のサービスでは到着順以外の台の割り当て方法を試し、各割り当て方法の実用上の問題を明らかにしたい。

SNS を活用したキムワイブの画像認識

崎下 雄稀

東京大学キムワイブ卓球会

E-mail: sakishita-yuki26@g.ecc.u-tokyo.ac.jp

概要

SNS 上で日々投稿される「キムワイブらしい」画像を機械学習で学習し、キムワイブらしさを定量化する識別器の作成を目指す。

1. 背景

SNS 上には「キムワイブ」と称して様々な画像が投稿されている。電車やバス、建物といった本来キムワイブとは全く異なる被写体であっても非常にキムワイブらしさを備えたデザインのもの数が多く存在する。単にキムワイブらしいと言っても人により評価は様々であり、色や形などキムワイブらしさを決定づける要因は判然としない。そこで、キムワイブらしさを客観的に測る指標として機械的なアルゴリズムによる評価を提案する。SNS からキムワイブらしいと思われる画像を収集し、機械学習により識別器を作成することで個々の画像がどれだけキムワイブらしいのかを定量的に判定し、本質的なキムワイブらしさを決定する要因の探求を目指す。

2. 手法

Twitter で収集したキムワイブらしい画像、及びキムワイブそのものの写真を総じて 426 点、さらに適当にキムワイブでない画像を同数程度集め、キムワイブかそうでないかのラベルをつけて学習する。

計算は Python の機械学習用ライブラリ、scikit-learn を用いた。そもそも機械学習とはコンピューター上で人間の学習に当たる働きを実現する技術の総称であり、近年話題の多いディープラーニングなどもこの 1 つである。

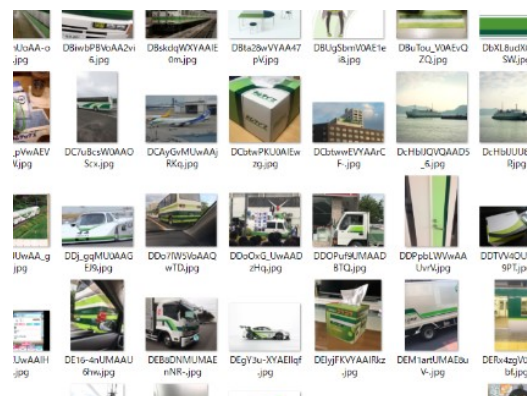


図 1: 図：収集したキムワイブらしい画像

機械学習についての概略と今回用いた手法の簡単な説明、そしてその結果をご報告したい。

3. 課題

今回は学習にあたってのパラメータ調整の難易度から深層学習の適用は見送ったが、画像認識は深層学習の得意とするところであり、これや他の分析手法を取り入れより性能の高い識別器の可能性に期待したい。

また、今回はキムワイブそのものについての適用であったが、キムワイブ卓球への機械学習の導入も模索していきたい。

参考

Sebastian Raschka, 福島 真太郎 訳 (2015) 『Python 機械学習プログラミング 達人データサイエンティストによる理論と実践』 株式会社クイープ

バーチャルキムワイプ卓球の実現及び VR体験がもたらす可能性

濱田 柚子
VRアカデミー3期
Twitter @citron_114

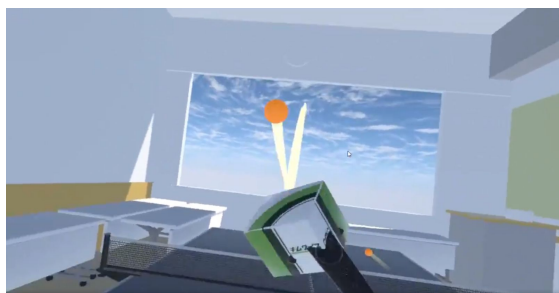
技術提供
宮原 啓太
東京農工大学 情報工学部 情報工学科 2年

概要

サイエンティフィックスポーツであるキムワイプ卓球をVR空間内で実現し、物理挙動のシミュレーション及び、キムワイプの質感を3DCGで再現。
VRキムワイプ卓球体験がもたらす可能性について提案する。

目的と課題

理系大学や研究室などで見られるサイエンティフィックスポーツであるキムワイプ卓球文化をVR空間に進出させ、キムワイプ卓球の知名度の向上を図るとともに、バーチャルキムワイプ卓球という新体験をより多くの人々に届け、楽しんでもらうためである。



▲開発中のVRキムワイプ卓球

上記の目的で私はVRキムワイプ卓球を開発してきたが、現状私の開発していたキムワイプ卓球には問題点がいくつかある。

VIVEコントローラを介してキムワイプをもつという行為では、キムワイプを実際にもっている感覚があまりない。

初めて体験する人の球の打ちやすさを重視するあまり、球の物理挙動がふわふわとしていて、現実味を帯びていない。

過去に自作したキムワイプS-200の3Dモデルは、テクスチャや形状など非常に不格好なものである。テクスチャやノーマルマップを駆使し、特徴的な容器とワイパー部分の材質を3DCGで忠実に再現していきたい。

方法

【使用ソフト】

- ・ Unity 2018.3.12f1
- ・ Autodesk Maya 2017
- ・ Substance Painter 2018.2.1
- ・ CLIP STUDIO PAINT

【使用器具】

- ・ キムワイプ(S-200、M-150)

- ・ HTC VIVE
- ・ G-Gear ノートPC
(CPU Core i7-7700HQ プロセッサー
GPU NVIDIA GTX1060
メモリ 16GB
SSD 512GB
Windows 10 pro)

アプローチ

本研究では、上記に挙げた課題を改善し、ローカルなVR空間上で1対1のキムワイプ卓球を実現する方法を模索していく。

サイエンティフィックで面白みのあるVRならではの新体験を交え、より良いバーチャルキムワイプ卓球体験を目指していくこととする。

参考文献

[1] <https://www.iktta.org/index.ja.html>

[2] Iwatsuki, K. (2016). Laws of Kimwipe Table Tennis. Scientific Sports, 1(1), 1–4.