

令和6年 3月 25日

## 令和5年度 学生自主研究成果報告書

教 育 本 部 長 様

学生自主研究グループ名	コマ観戦研究会	
研究課題名	カメラを止めるな！	
研究代表者（学生）	学籍番号	B24M047
	氏 名	名苗 千諒
指導教員	学 科	機械工学科
	氏 名	鈴木 庸久

学生自主研究の報告書を別紙のとおり提出します。

## カメラを止めるな！

システム科学技術学部 機械工学科

2年 名苗 千諒

1年 染谷 浩志, 高橋 功太, 藤戸 信吾, 伊藤 武蔵, 岩城 俊哉, 田村 豪人

指導教員 システム科学技術学部 機械工学科

教授 鈴木 庸久, 准教授 野村 光由, 助教 藤井 達也

## 1. 緒言

## (1) 目的

令和5年10月21日に行われた、全日本製造業コマ大戦（特別）あきた場所において優勝することを目指し、企業と協力してコマの設計・製作を行う。大会実施後は出場したコマの特性を調べ、カメラを使ってコマの回転数を計ることを目的とする。

## (2) コマ大戦大会概要

このコマ大戦あきた場所は東北地方の学校の学生と企業が協力して製作したコマを戦わせる大会である。今大会では直径20mm以下、高さ60mm以下のコマが使用可能であり、コマの素材は無制限である。

## 2. 研究内容

コマ大戦において、戦っている二つのコマのどちらが優勢で、どちらが劣勢かを目視で判断することが困難であったことが多く、その優勢、劣勢をリアルタイムで判断することができれば、コマ大戦をより興味深いものにすることができると考えた。また、その優勢、劣勢はコマの回転数から判断できると考えた。ゆえにこの研究では、コマの回転数とコマの特性との関わりを調べる。以下にコマの回転数や諸特性の測定方法を示す。

## (1) 質量

コマの回転速度と質量には関係があると考えられ、質量が大きいほどコマの回転数が下がると考えられるので、質量は回転速度にどのように関係しているのかを調べるためにコマの質量を測定する。測定方法は、それぞれのコマを電子はかりで3回ずつ測定し、その平均値を算出する。

## (2) 回転数

コマの回転数と質量の関係を調べるため、下記の2通りの方法で回転が始まってから5秒後のコマの回転数を5回ずつ計測し、計測後にその平均値を算出する。

①専用のテープをコマに貼り付けて回転させ、それを認識することで回転数を計測する非接触の回転計を使う。

②突起がついているコマには突起を認識することで回転数を計測する計測器を使う。

## (3) カメラの fps の操作によるコマの回転数の測定

スマートフォンのアプリにおいて、スローモーション動画撮影が搭載されたスマートフォンを使い、カメラのフレームレイト (fps) を変更させることが出来るビデオカメラというものがある。今回の研究においてはこのアプリを、コマに印を付け、カメラの fps を変更させることで、コマの印がカメラ越しに止まったように見える fps を探すことでコマの回転数を測定するという使い方が出来るこ

とが考えられた。そこで、このスマートフォンのアプリであるビデオカメラを自分たちのパソコンでプログラミングにより再現することを研究対象とした。その内容は、プログラミング言語である Python を Anaconda という利用環境の下で、Python を利用する上で便利なソフトウェアである Spyder により、画像・動画に関する処理機能をまとめた OpenCV と組み合わせてプログラムするものである。実際にプログラムした2通りの内容を下記に述べる。

① カメラのフレームレイトを設定し、リアルタイムで撮影された映像をパソコンに表示する。

② 撮影した動画の出力するフレームレイトを調整し、その結果をパソコンに表示する。



図1 実際のプログラミングのようす(右)、アプリであるビデオカメラ(左)

また、(2)の回転数の測り方は、回転軸の外周面に反射テープを貼り、回転計から赤色可視光やLEDを出して反射マークで反射する光をカウントする反射マーク方式を用いているのに対し、(3)の方法は静止面の撮影周期と回転系の回転周期が一致すると回転系が静止しているように見え、そのタイミングがわずかにずれると、回転系がゆっくり回ったり、逆回転しているように見えるストロボ効果を利用している。それぞれの効果の概略図を下記に示す。

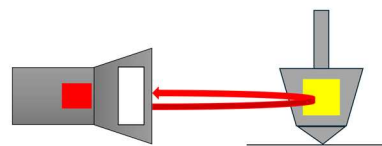


図2 反射テープ方式の様子

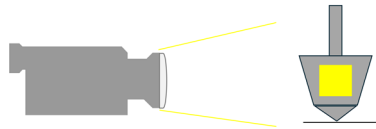


図3 ストロボ効果を利用している様子

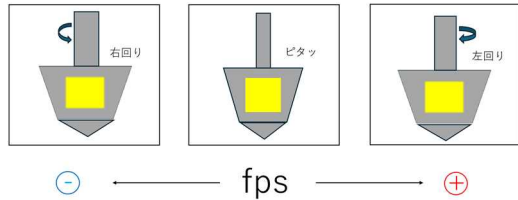


図4 ストロボ効果において、fpsを変えたときの様子  
(コマが紙面において右回りするとき)

### 3. 実験結果

以下にコマの特性についての計測結果を述べるが、対象とするコマは令和5年10月21日に行われた、全日本製造業コマ大戦（特別）あきた場所に出場したコマである。

表1に、コマの質量と回転数の測定結果を示す。なお、「2. 研究内容」の(2)における①を回転数1、②を回転数2としており、 $\sigma$ は得られたデータの標準偏差を示している。斜線部分はコマの形状によって回転数が測れなかったものである。

表1 コマの質量と回転数

No	重量[g]	回転数1[rpm] ( $\sigma$ )	回転数2[rpm] ( $\sigma$ )
1	52.5	2507.38(76.715)	
2	49.8		2504.8(98.30)
3	26.4		2426.0(226.2)
4	24.4	2582.16(162.80)	2579.8(161.7)
5	58.0	2122.64(150.03)	2123.8(152.7)
6	7.6	3479.22(709.99)	4291.8(313.9)
7	47.6	1961.32(255.28)	2214.0(222.5)
8	33.6	2614.72(194.89)	2593.8(212.0)
9	17.6	3408.72(175.22)	3436.8(196.0)
10	42.5	2333.30(182.20)	2260.0(78.32)
11	37.8	2265.14(320.29)	2269.8(318.3)
12	43.9	1961.16(164.53)	1957.2(152.0)
13	20.2	3589.30(150.63)	3588.6(138.2)
14	21.4	3609.94(140.56)	3695.0(152.0)
15	36.8	2826.92(197.57)	2837.0(198.7)
16	38.8	2769.12(217.74)	
17	35.3	2773.72(76.922)	2769.2(70.62)
18	5.8	3984.40(446.86)	3980.0(438.3)
19	31.2	3225.16(295.44)	3225.6(291.3)
20	42.9	2283.76(218.09)	2278.2(219.61)
21	30.5	2689.82(316.57)	
22	66.3	1842.24(112.27)	1830.8(103.4)

図5に、表1のコマの質量と回転数の関係を示す。

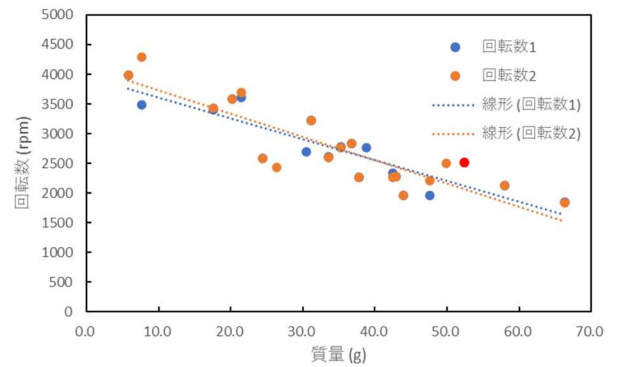


図5 コマの質量と回転数の関係

図5は横軸  $x$  に質量、縦軸  $y$  に回転数を示している。なお、赤い点は表1でNo1とされているコマのデータ的位置を表している。回転数1グラフの近似曲線は、

$$y = -35.061x + 3959.4 \quad (1)$$

となっているので、質量と回転数には強い負の相関があるといえる。

回転数2のグラフの近似曲線は、

$$y = -39.395x + 4128.2 \quad (2)$$

となっているので、質量と回転数には強い負の相関があるといえる。

図6に、「2. 研究内容」の(2)で述べた2通りの回転数の測り方の関係を示す。

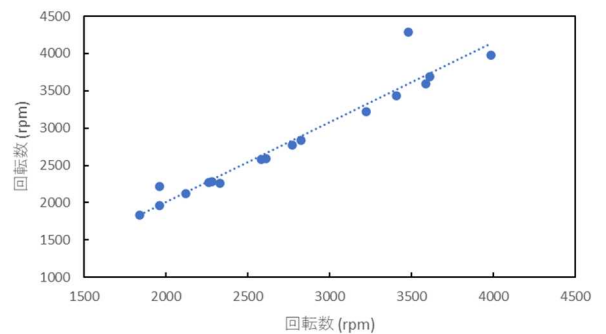


図6 2種類の回転計での回転数の関係

図6は横軸  $x$  に「2. 研究内容」の(2)の①で測ったときの回転数、縦軸  $y$  に「2. 研究内容」の(2)の②で測ったときの回転数を示している。このグラフの近似曲線は

$$y = 1.0712x - 133.4 \quad (3)$$

となっているので、ほぼ比例の関係があるといえる。

以下に、「2. 研究内容」の(3)で述べたPythonとOpenCVを組み合わせて、スマートフォンのアプリであるビデオタコメーターを再現した際のプログラミングの内容を示す。

表2に、「2. 研究内容」の(3)で述べた2通りのプログラムの内容を示す。なお、色がついている文字は今回のプログラミング環境であるSpyderを使った際に自動的に色付けされるものである。

表2 カメラ、動画の fps を調整した際のプログラム

カメラのフレームレイトを調整	動画のフレームレイトを調整
<code>import cv2</code>	<code>import cv2</code>
<code>import sys</code>	<code>cap = cv2.VideoCapture('data/temp/sample_video.mp4')</code>
<code>camera_id = 0</code>	<code>if (cap.isOpened() == False):</code>
<code>delay = 30</code>	<code>print("ビデオファイルを開くとエラーが発生しました")</code>
<code>window_name = 'frame'</code>	<code>while(cap.isOpened()):</code>
<code>cap = cv2.VideoCapture(camera_id)</code>	<code>ret, frame = cap.read()</code>
<code>if not cap.isOpened():</code>	<code>if ret == True:</code>
<code>sys.exit()</code>	<code>cv2.imshow("Video", frame)</code>
<code>while True:</code>	<code>if cv2.waitKey(25) &amp; 0xFF == ord('q'):</code>
<code>ret, frame = cap.read()</code>	<code>break</code>
<code>cv2.imshow(window_name, frame)</code>	
<code>if cv2.waitKey(delay) &amp; 0xFF == ord('q'):</code>	<code>else:</code>
<code>break</code>	<code>break</code>
<code>cv2.destroyAllWindows(window_name)</code>	<code>cap.release()</code>
	<code>cv2.destroyAllWindows()</code>

上記のプログラムで使用している `waitkey()` というものは、()内の数字が次のフレームを表示するまでの待ち時間を表しており、カメラの fps を調整する方のプログラムでは `delay` で `waitkey()` の引数を指定するようになっている。例えば、`delay = 1000` とすると、`waitkey(1000)` となり、1000 ミリ秒つまり 1 秒おきに 1 フレーム分の画像が表示される(概ね 1fps)。動画の fps 調整のプログラムでは、スロー再生や早送りは出来るもの、スマートフォンのアプリであるビデオタコメーターのような機能は再現できなかった。カメラの fps 調整のプログラムでは、`delay` の値を調整することにより、カメラの fps を変更し、コマが止まって見えるような映像を撮影することに成功した。下記に、`delay` の値と fps の関係を示す。

$$(\text{delay の値}) = \frac{1}{(\text{表示する映像の fps})} \quad (5)$$

#### 4. 考察

図 5 について、質量と回転数の関係にはそれぞれ強い負の相関があることがわかる。つまり、長い時間回ることが長所となる耐久型のコマにおいては、コマの軽量化が求められることがわかる。また、図 5 で赤く塗られているコマは、2 通りの回転計の測定において質量が大きいわりに回転数とその質量に近いコマよりも多いことがわかる。これは、このコマには、軸とコマの胴体の間に空洞が設けられており、中心から外側にかけての質量の分布を軸付近と外径付近で大きくし、その中間は 0 に近い値にすることで、コマを回転させる力が増大したためと考えられる。

図 6 について、「2. 研究内容」の(2)で述べた 2 通りの計測での回転数への影響を見ると、式(3)より、ほぼ比例の関係があるが、コマにシールを貼り回転数を計測したときの方が、突起を認識することで回転数を計測したときよりも 133rpm だけ回転数が全体的に大きいことがわかる。実際の計測中においても、コマにシールを貼り回転数を計測する方法では、手でコマを回すという方法においては、ありえない大きさの rpm の値を出すことが多かった。これは、形状の違いを認識することで回転数を計測する回転計に比べ、コマにシールを貼り回転数を計測する回転計が、赤色の可視光を対象物に発射し、シールから反射された光を読み取ることで回転数を測るため、今回はコマが金属製で

あり金属特有の光沢や周囲の蛍光灯から反射した光によって誤った認識をして、その分だけ回転数が多くなったのではないかと考えられる。

表 2 について、動画 fps を変える場合は、スマートフォンのアプリであるビデオタコメーターのように、回転している物体が止まって見えるような映像を表示させることは出来なかった。これは、動画の場合、プログラムの中にある `read()` の部分で次のフレームに進むようにできているため、必ずすべての撮影されたフレームが表示されるので、`waitkey()` というフレームを表示させる間隔を調整できるプログラムで動画のスロー再生や早送りは出来るが、フレームの一部を表示させないようにして、あたかも回転していないような映像をとることはできないと考えられる。だが、カメラの fps を調整するプログラムでは、リアルタイムで映像が入力されているため `read()` するしなないに関わらず映像は常に更新され、`read()` して読み込まれるものはそのタイミング以降に入力される最新のフレームとなる。つまり、`waitkey()` で待ち時間を調整すればカメラの fps を下げつつ、撮影されている映像を表示することが出来るので、動画の fps を調整するプログラムと違い、回転する物体が止まって見えるような映像を表示することが出来たと考えられる。また、カメラの fps を調整する方法では、表示できる映像の fps の上限値が撮影しているカメラの fps の上限値と同じなため、今回の使用したコマは最高で 4000rpm 近く回転するものがあって、約 70fps 以上で撮影が出来るカメラが必要になると考えられる。

カメラの fps を調整するプログラムにおいて、10fps で撮影した際にコマにつけた印が止まって見えるときがいくつも存在していた。これは、コマが 10fps つまり 600rpm で回転するところがいくつもあり、コマが回っている最中に回転の加速や減速を大きく繰り返しているわけではなく、コマが撮影している fps の自然数倍の回転になった際に止まって見えていると考えられる。例えば、コマが 30fps つまり 1800rpm で回転しており、そのコマを 10fps で撮影していたとすると、次のフレームを表示する際にコマが 3 回転してカメラに向けて前のフレームと同じ状態を見せるので、あたかもその瞬間に 10fps でコマが回転しているように見えると考えられる。

カメラの fps を調整するプログラムにおいて、コマが止まって見える映像を撮影することには成功したが、スマートフォンのアプリであるビデオタコメーターと比べると再現できない箇所が存在した。その一つにアプリの方では図 1 にある通り、カメラの rpm の値を映像を流しながらボタンを押すことで変更していたが、今回使用したプログラムでは、一度映像を切り `delay` の値を変えて再び撮影を開始しなければいけないようになってしまった。しかし、プログラミング言語である Python にはボタンを作成し、それを押すことでプログラムを実行できる機能があるのでそれをうまく使い、さらにアプリの内容と近くしていくことが次回からの研究の内容になるのではないかと考えられる。

#### 5. 結言

今回の研究では主に回転に関することを行った。これにより、質量が軽いコマは回転数が多くなることがわかった。また、プログラミング言語である Python を使って、コマが止まって見える映像をリアルタイムで撮影することができ、その時のコマの回転数がある程度把握することが出来るようになった。しかし、スマートフォンのアプリであるビデオタコメーターと比べて、映像を流しながらカメラの fps を変更することはできなかった。