Vol.2021-IOT-54 No.1
2021/7/9

大学における新型コロナウィルス感染拡大防止のための健康調査入力簡易化システムの開発の試み

山之上 卓[1]　成瀬 悠朔[1]　尾関 孝史[1]

概要：福山大学では新型コロナウィルス感染拡大防止のための健康調査が毎日行われているが、その入力のために多くの学生教職員が煩わしさを感じており、その煩わしさは健康調査入力率の低下にもつながっている。この煩わしさを低下させるため、入力の手間の大部分を省略するためのシステムを開発している。このシステムの試作について述べる。

キーワード：双方向，デジタルサイネージ，スマートフォン

Experimental Development of an Easy Input System in the Health Survey for Preventing COVID-19 Infection in a University

Takashi Yamanoue[†1] Yusaku Naruse[†1] Takashi Ozeki[†1]

Abstract: Every Students, faculty members and officers of Fukuyama university is asked to answer questions of health survey every day, for mitigating spreading of COVID-19 infection. However, many students and staff were feeling annoying. In order to mitigate this feeling of annoying, we are developing an easy input system in the health survey. This paper discusses the experimental development of the input system.

Keywords: Intrusion detection, Mining malware, botnet

1. はじめに

福山大学では新型コロナウィルス感染拡大防止のための健康調査が毎日行われている。この調査と、対面授業の３密回避と座席指定、スクールバスの３密回避、建物入口のアルコール消毒剤の設置、パソコン室等への消毒ティッシュの設置、高濃度二酸化炭素検知・警告装置設置などの対策と組み合わせることにより、学内での感染拡大防止に成功している。

しかしながら、調査が長期化するに従い、毎日の健康調査の入力については多くの学生教職員が煩わしさを感じており、その煩わしさは健康調査入力率の低下にもつながっている。健康調査入力率の低下は、感染拡大につながる可能性がある。

この煩わしさを低下させるため、入力の手間の大部分を省略するためのシステムを開発している。このシステムの試作を行ったことについて述べる。

2. 健康調査システム

新型コロナウィルス感染拡大は 2020 年以降の大学の活動に大きな影響を与えた。2020 年の春、多くの大学は立ち入り禁止となり、対面授業の代わりにオンライン授業が実施された。その後、いくつかの大学では対面授業が始まった。

我々の大学は 2020 年５月まで授業開始を遅らせて、その後、オンライン授業を開始することになった。

新型コロナウィルスは、他との接触を絶った後、２週間、その兆候が表れていない場合、感染している可能性が低いことが知られている。我々の大学の危機対策本部は 2020 年 4 月 8 日に、対面授業を安全に実施し、感染拡大を緩和するため、健康調査を実施することを発表した。全ての学生教職員に対して、毎日、健康調査ページの各項目に記入し、新型コロナウィルスに感染した兆候があった場合、または、２週間の間に健康調査に未記入があった場合には、大学に来ないよう、要請することになった。この調査では、体温、その他の健康状況、訪問先、住所、関連するコメントも調査している。

この調査は、データを入力し、解析する労力を削減するため、Microsoft Office 365 を使って行われている[1]。

この調査によって、潜在的な新型コロナウィルスの感染を知り、早い時期に対応を取ることが可能になる。一人の学生が新型コロナウィルスに感染した場合、周りに感染者がいないか、調べることになるが、この調査と、座席指定の情報等を使って、他の学生・教職員の、潜在的な感染者の範囲を狭めることができる。

1 福山大学
Fukuyama University

Vol.2021-IOT-54 No.1
2021/7/9

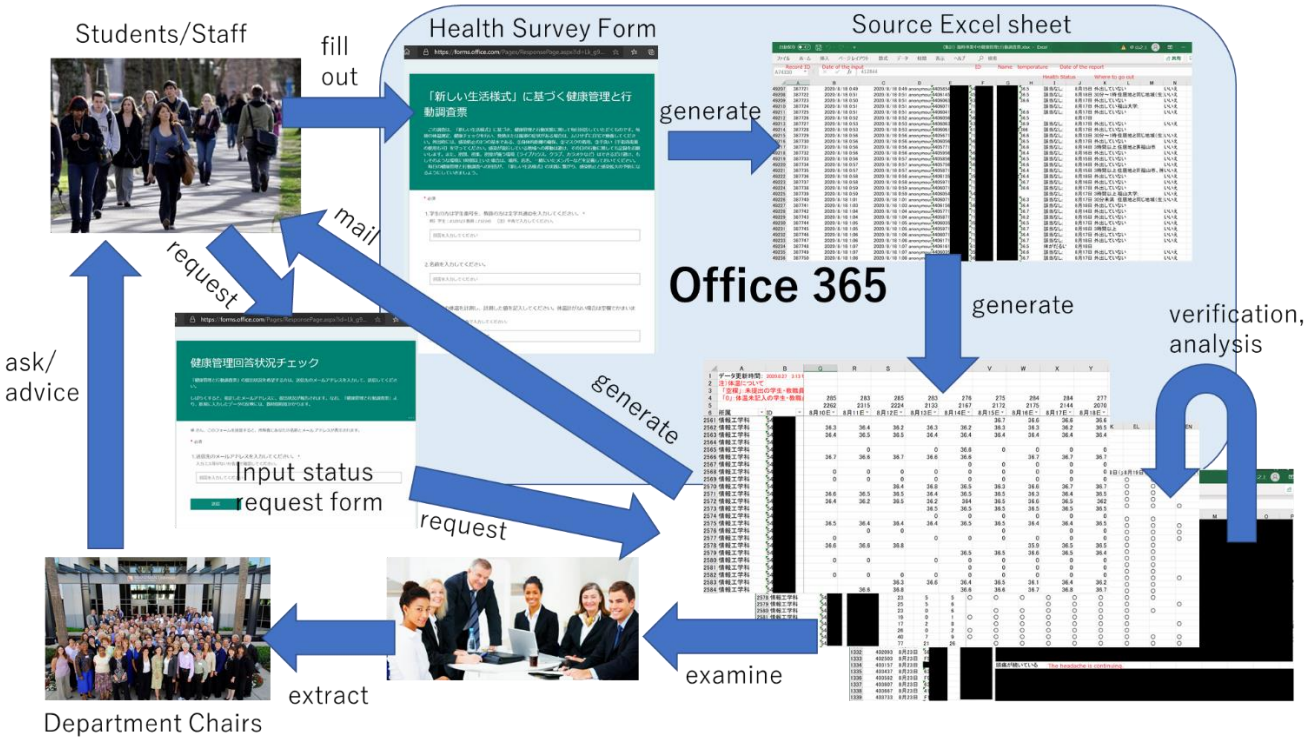

図 1　健康調査システムの構成とデータの流れ

図 2 健康調査票（Health Survey Form）の一部

この調査を含め、福山大学で実施された新型コロナウィルス感染対策は「福山市新型コロナウイルス感染症対策ガイドライン」に基づく認定証を受けた[2]。

図 1 に、健康調査システムの構成とデータの流れを示す。図 2 に学生・教職員が毎日健康調査の入力を行う健康調査票を示す。この調査票は Microsoft Office 365 の Forms を使って作成された Web ページである。2020 年度の入力項目は以下のとおりである。2021 年度になって、一部変更されている。

- 学生番号または教職員 ID
- 名前
- 体温
- 健康状態（選択-複数可）
- 報告日
- 外出時間
- 外出した時の場所（選択-複数可）
- 外出した場所が居住地や大学と異なる場合の場所
- 三密環境に 1 時間以上いた場合の有無
- その場合の場所
- 居住地（選択）
- 居住地の具体的な場所（登録住居以外の場合）
- なにか不安な点がある場合のコメント

調査票の項目の多くは選択式であり、スマホでも入力

可能であるが、それでも、入力にはそれなりの時間が必要であり、入力項目の多くは、健康でいつもの生活と変化がなくても、いつも同じ入力を行うことになる。このため、学生・教職員が入力に煩わしさを感じる場合があり、全学生・教職員が毎日入力を実施できていない。体温計を持っていない場合も考慮して、体温は空白でも良い、となっており、また、入力を忘れていた場合、過去の入力も可能としているが、それでも、学生・教職員全員が毎日入力を行うまでに至っていない。

3. 健康調査入力簡易化システムその１

対面授業の受講者は、健康調査を過去２週間にわたって実施していることが求められているが、どうしても入力忘れが発生する。大学に来ている場合は自分自身が健康であると思っているはずであり、その場合の健康調査の入力項目の多くは、いつも同じになる。

新型コロナウィルスは複数の人が共に触る物を通じて感染する可能性があるので、共通の機器で入力する場合、非接触で入力できると良い。

これらを踏まえ、対面授業直前でも、受講者がすぐに健康調査入力を非接触で行うことができるよう、健康調査入力簡易化システムを試作した。図 3 に健康調査入力簡易化システムの外観を示す。

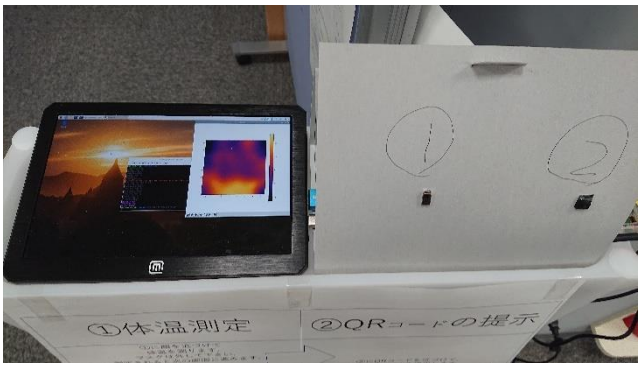

図 3. 健康調査入力簡易化システムの外観

図 4. QR コードの例
(fukuyama-u;id=F0000; name=福山　太郎)

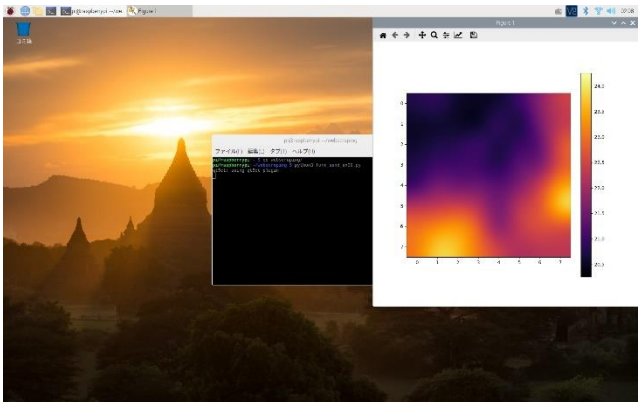

図 5 推定体温の入力画面

このシステムは、体温をサーモグラフセンサで計測し、学生番号・教職員 ID、氏名を、QR コードを使って、非接触で入力し、健康調査システムの健康調査票に、他の、大学に来ていれば共通になることが自明な項目と一緒に、自動的に入力するものである。図 4 に QR コードの例を示す。

このシステムを利用するときは、

1. 顔を図 3 の①のサーモグラフィセンサに近づけて体温の推定値を測定
2. 事前にスマートフォンなどに取り込んだ、fukuyama-u;id=<学籍番号または教職員 ID>;name=<名前> をコード化した QR コードを図 4 の②のカメラに近づけてコードを読み取る。

を利用者が行う。２を実施した後、健康調査票を入力するための Web ブラウザが表示されて、自動的に入力が行われる。

3.1 システムのハードウェアとソフトウェア

このシステムのハードウェアは、

- Raspberry Pi 3 又は Raspberry Pi 4
- AMG8833 搭載 I2C サーモグラフセンサ
- Raspberry Pi カメラ

を組み合わせて作成した。

このシステムのソフトウェアは、Python 3 の上で、

- 体温入力クラス AMG8833
- QR コード入力クラス QR_Reader
- 健康調査票自動入力クラス Browser_Controller
- 全体の制御部

の４つの部分でできている。

3.2 体温入力クラス AMG8833

図 6 に、体温入力クラス AMG8833 の定義を示す。このクラスは、@tm_nagoya 氏の Qiita のページ[3][4]を参考にして、Adafruit 社から提供されている CircuitPython パッケージと、matplotlib パッケージを使用して、作成した。

```python
class AMG8833:
    def __init__(self):
      # I2Cバスの初期化
      self.i2c_bus = busio.I2C(board.SCL, board.SDA)
      # センサーの初期化
      self.sensor = adafruit_amg88xx.AMG88XX(self.i2c_bus, addr=0x68)

      # センサーの初期化待ち
      time.sleep(.1)

      # 8x8ピクセルの画像とbicubic補間をした画像を並べて表示させる
      #plt.subplots(figsize=(8, 4))
      plt.subplots(figsize=(8,8))
      self.btemp=0;

    def mloop(self):
      # ループ開始
      while True:
        # データ取得
        sensordata = self.sensor.pixels
        #print(sensordata)
        mx=-1.0;
        fac=0;
        th01=27.0;
        for r in sensordata:
            for c in r:
                if(c>th01):
                    fac=fac+1
                if(c>mx):
                    mx=c
        if(fac>30 and fac<45):
            #print(mx)
            print(mx+5.5)
            self.btemp=mx+5.5
            plt.close()
            return
        # 8x8ピクセルのデータ
        #plt.subplot(1, 2, 1)
        #fig = plt.imshow(sensordata, cmap="inferno")
        #plt.colorbar()

        # bicubicのデータ
        #plt.subplot(1, 2, 2)
        plt.subplot(1,1,1)
        fig = plt.imshow(sensordata, cmap="inferno", interpolation="bicubic")
        plt.colorbar()

        # plt.showだと止まってしまうので、pauseを使用
        # plt.clfしないとカラーバーが多数表示される
        plt.pause(.1)
        plt.clf()
    def getBTemp(self):
        return str(self.btemp)
```

図 6. 体温入力クラス AMG8833

```python
class QR_Reader:
  def __init__(self):
    self.root = tkinter.Tk()
    self.root.title('QR reader')
    self.root.geometry('640x488')
    self.CANVAS_X = 640
    self.CANVAS_Y = 480
    self.canvas = tkinter.Canvas(self.root, width=self.CANVAS_X, height=self.CANVAS_Y)
    self.canvas.pack()
    self.cap = cv2.VideoCapture(0)
    self.cont_cap=True;
    self.xid=""
    self.xname=""

  def capture_code(self):

    if(not self.cont_cap):
      self.root.destroy()
      self.cap.release()
      return;
    ret, frame = self.cap.read()
    if ret == False:
        print("Not Image")
    else:
        image_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
        image_pil = Image.fromarray(image_rgb)
        image_tk = ImageTk.PhotoImage(image_pil)
        self.canvas.image_tk = image_tk
        self.canvas.create_image(self.CANVAS_X / 2, self.CANVAS_Y / 2, image=image_tk)

        decoded_objs = pyzbar.decode(frame)

        if decoded_objs != []:
            for obj in decoded_objs:
                print('Type: ', obj)

                str_dec_obj = obj.data.decode('utf-8', 'ignore')
                print('QR code: {}'.format(str_dec_obj))
                if(str_dec_obj.startswith("fukuyama-u;id=")):
                  cand=str_dec_obj.split(";")
                  idxx=cand[1]
                  print(idxx)
                  idxxx=idxx.split("=")
                  print(idxxx)
                  self.xid=idxxx[1]
                  print(self.xid)
                  namexx=cand[2]
                  print(namexx)
                  namexxx=namexx.split("=")
                  print(namexxx)
                  self.xname=namexxx[1]
                  print(self.xname)
                  self.cont_cap=False
                left, top, width, height = obj.rect

                self.canvas.create_rectangle(left, top, left + width, top + height, outline=
                self.canvas.create_text(left + (width / 2), top - 30, text=str_dec_obj, font=

    self.canvas.after(10, self.capture_code)
  def main(self):
    self.capture_code()
    self.root.mainloop()
  def getXid(self):
      return self.xid
  def getXname(self):
      return self.xname
```

図 7. QR コード入力クラス QR_Reader

```python
class Browser_Controller:
    def __init__(self):
      #self.chromepath='/usr/lib/chromium-browser/chromedriver'
      #self.browser = webdriver.Chrome(self.chromepath)
      self.browser = webdriver.Chrome()
      self.url = "https://forms.office.com/pages/responsepage.aspx?id=Lk_g9n8Cr
      self.browser.get(self.url)

    def input(self, xid,xname,xtemp):
      print("xid="+xid)
      print("xname="+xname)
      print("xtemp="+xtemp)
      sleep(3)
      uid = self.browser.find_element_by_xpath("//input[1]")
      uid.send_keys(xid)
      uname = self.browser.find_element_by_xpath("//*[@id=\"form-container\"]/d
      #uname = self.browser.find_element_by_xpath("//input[2]")
      uname.send_keys(xname)
      btemp = self.browser.find_element_by_xpath("//*[@id=\"form-container\"]/d
      btemp.send_keys(xtemp)
      chose_01 = self.browser.find_element_by_xpath("//*[@id=\"form-container\"
      chose_01.click()
      now = datetime.datetime.now()
      print(now)
      today=now.strftime('%Y/%m/%d')
      print(today)
      date= self.browser.find_element_by_xpath("//*[@id=\"form-container\"]/div
      date.send_keys(today)
      xtime=self.browser.find_element_by_xpath("//*[@id=\"form-container\"]/div
      xtime.click()

      xplace=self.browser.find_element_by_xpath("//*[@id=\"form-container\"]/di
      xplace.click()

      xsanmitsu=self.browser.find_element_by_xpath("//*[@id=\"form-container\"]
      xsanmitsu.click()

      xresidence=self.browser.find_element_by_xpath("//*[@id=\"form-container\"
      xresidence.click()
      #send_button=self.browser.find_element_by_class_name('button-content')
      send_button=self.browser.find_element_by_xpath("//*[@id=\"form-container\
      send_button.click()
      time.sleep(2)
      #dmy=input()
      self.browser.quit()

while True:
  amg=AMG8833()
  amg.mloop()
  reader=QR_Reader()
  reader.main()
  bx=Browser_Controller()
  bx.input(reader.getXid(),reader.getXname(),amg.getBTemp())
```

図 8. 健康調査票自動入力クラス Browser_Controller と全体の制御部

サーモグラフィは非接触で物体の表面温度を計測することができるが、できるだけセンサから顔までの距離が一定となるところで顔の温度を計測した方が良い。そこで、サーモグラフで得られた８x８の温度分布において、顔の温度を計測している場合はその部分は 27 度 C 以上あるであろう、と推測し、顔の部分が 8x8 のうちの 30 画素より多く、45 画素より少ない場合の、最大の温度を計測することにより、センサから顔までの距離がだいたい同じ場合の温度を計測している。また、実験により、その最大温度に 5.5 を加えた値が体温に近いであろうと推測して、その値を体温の推定値として採用している。

3.3 QR コード入力クラス QR_Reader

図 7 に、クラス QR_Reader の定義の大部分を示す。このクラスは、@PoodleMaster 氏の Qiita のページ[5]を参考にして、OpenCV モジュール、pyzbar モジュールを使用して、作成した。

図 7 の行番号 111 の if 文において、QR コードの解析結果が、”fukuyama-u;id=”で始まるものだけについて、行番号 112 から 125 までの処理を行っている。この範囲内で、学籍番号または教職員番号と、氏名を取り出している。

3.4 健康調査票自動入力クラス Browser_Controller と全体の制御部

図 8 に、健康調査票自動入力クラス Browser_Controller

と全体の制御部のコードを示す。

健康調査票の入力は Web ページで行われるため、その自動入力には、Python の Selenium を使っている。Qiita の @Brutus 氏のページ[6]や「分かりやすい技術ブログ」の「Raspberry pi で selenium がやっと動かせた」のページ[7]等を参考にして作成した。

3.5 健康調査入力簡易化システムその１の問題点

大学に来た学生に、健康調査入力簡易化システムその１を使って実際に入力をしてもらったところ、体温が高い自覚がない学生の推定体温が異様に高い場合が頻発した。このとき、別の非接触体温計で計測した場合に平温であることを確認できた。同じ学生でも、センサに顔を近づける方法によって、推定値が異なる場合があった。このため、学生から、自覚がないのに高い体温の推定値が得られた場合、入力のやり直しができるようにして欲しい、との要望が得られた。

4. 健康調査入力簡易化システムその２

健康調査入力簡易化システムその１の問題点を解消し、学生の要望に応えるため、体温の推定値を計測したとき、その値がおかしかった時に、計測しなおしたり、何度計測してもおかしい時は、空白を入力したりできるようにするため、健康調査入力簡易化システムその１を拡張して、健康調査入力簡易化システムその２を試作した。

計測した体温を表示して、利用者に見せることは容易であるが、その後、その値によって、利用者が、そのまま継続するか、再入力するか、空白を入力するか判断し、システムに入力する必要がある。光センサや超音波距離センサを使って、非接触でこの入力を行うことも考えられるが、フットペダルを使うことにより、完全な非接触ではないが、感染防止としては問題なく簡単に入力を行うことができる。

そこで、この入力の部分を、フットペダルを利用することにより、健康調査入力簡易化システムその２を試作した。利用したフットペダルは Raspberry Pi と USB ケーブルで接続され、Raspberry Pi から見ると、キーボードとして認識されるので、ソフトウェアは健康調査入力簡易化システムその１を少し変更するだけで実現することができた。

図 9 に、健康調査入力簡易化システムその２の外観を、図 10 に，この入力システムを利用している様子を示す。図 11 に、システムが推定体温を表示し、利用者にフットペダルでの入力を促しているとことを示す。

図 12 にフットペダルを使うために変更した全体の制御部を示す。３つのフットペダルの左から、”a”, “b”, “c”の文字が割り当てられている。Raspberry Pi でフットペダルを踏んだ時に１文字だけ入力するために、readchar モジュールを利用している。

図 9. 健康調査入力簡易化システムその２の外観

図 10 健康調査入力簡易化システムその２を利用しているときの様子.

```
qr_generator.py
qr_read_ex01.py
qrcode_A.png
qrcode_AB.png
qrcode_B.png
pi@raspberrypi:~/python $ python3 form_send_ex09.py
①のサーモセンサに顔を5cmくらいに近づけて、体温を計測してください。
37.25
↑があなたの推定体温です．
これで良ければ左のペダルを
やり直す場合は中央，空白で提出する場合は右のペダルを
踏んでください．
```

図 11. 推定体温計測後のフットペダル入力を促す表示

```
while True:
    print('①のサーモセンサに顔を5cmくらいに近づけて、体温を計測してください。')
    amg=AMG8833()
    amg.mloop()

    print('↑があなたの推定体温です．\nこれで良ければ左のペダルを\nやり直す場合は中央，空白で提出す
    while True:
        x=readchar.readkey()
        print(x)
        if x=='a':
            xtemp=amg.getBTemp()
            break
        if x=='b':
            break
        if x=='c':
            xtemp=""
            break
    if x=='a' or x=='c':
        print('fukuyama-u;id=<学籍番号>;name=<氏名> をコード化したQRコードを②のカメラに近づ
        reader=QR_Reader()
        reader.main()
        bx=Browser_Controller()
        bx.input(reader.getXid(),reader.getXname(),xtemp)
```

図 12. フットペダルを使うために変更した全体の制御部

2021 年度より、健康調査票のページに少し変更が施された。健康調査入力簡易化システムその２は、これに対応するための修正も加えられている。

5. 関連研究

5.1 赤外線サーモグラフィによるパンデミック対策事例

太田は、赤外線サーモグラフィで計測した人間の体表温度が、環境温度によって左右され一様では無く課題も多くあることに対して、発熱者を精度良く検知するための製品動向と利用技術の向上を目指した取り組みを紹介している[8]。本システムは、現時点で太田が紹介したこれらの know how を利用していない。本システムの推定体温の精度向上のためには、このような know how を利用する必要がある。

5.2 ウェアラブルセンサを用いた深部体温推定

濱谷らは人体の熱産生,熱移動を定式化した生体温熱モデルを用いて網羅的なシミュレーションを行うことで深部体温と対応するセンサ値組を算出し、深部体温の確率密度分布を事前に生成し,生成した分布のうち,実際に観測したセンサ値組に最も近い場合に対応する深部体温の確率密度分布に基づき,現在の深部体温を推定する方法を示している[9]。

また、提案手法の実現可能性を検討するため実データを収集し,センサ値組と深部体温の関係を確認したところ,深部体温と心拍数の値に相関 (R = 0.68) が見られ,さらに複数のセンサ情報を組み合わせることで,よりばらつきの少ない深部体温の分布を得られることを確認している。

ウェアラブルセンサを用いた体温測定は，濱谷らの方法などを使って，個々の体温を高い精度で，長期にわたって確実に計測することを可能にする。しかしながら、被測定者個々にウェアラブルセンサを装着する必要があり，大学全体や大学の学科全体でこれを実施する場合は、被計測者の心理的な負担が発生し、また、費用も高くなる。

本システムは、精度におおきな問題があるが、被計測者全員にセンサを装着する必要がなく、心理的負担が少ない。また、費用も少なく済む。

6. おわりに

Raspberry Pi とサーモグラフセンサとカメラとフットペダルを組み合わせて、実質的に非接触で操作できる、健康調査入力簡易化システムを試作したことについて報告した。現在、計測した推定体温の精度が悪いなどの問題がある。また、精神的負担間の問題もあるが、利用者の賛同が得られれば、ウェアラブルセンサと IoT 技術を用いて、本人はなにもしなくても自動的に入力が行われるようなシステムも検討したい。このとき、IoT 技術については、すでに[10][11][12]のような研究を行っているので、これを利用したい。

謝辞 健康調査システムを開発した本学片桐助教と危機管理対策本部の平副学長、健康調査入力簡易化システムその１の問題点を指摘し、修正のための要望を出してくれた本学大学院情報処理工学専攻の山本君、本システムを利用している場合の撮影に協力してくれた山上君、本システムを実際に利用してくれた学生諸君他、関係した学生・教職員に感謝します。

参考文献

[1] Shigekazu Katagiri, Takashi Yamanoue, Kazuki Yoshizu, Shinji Hira, "Preventing COVID-19 Infection in a University Using Office 365," SIGUCCS '21: ACM SIGUCCS Annual Conference, March 2021 Pages 60–65, https://doi.org/10.1145/3419944.3441219

[2] 「福山市新型コロナウイルス感染症対策ガイドライン」に基づく認定証, https://www.fukuyama-u.ac.jp/news/39203/ , 2021/6/13 閲覧

[3] @tm_nagoya, “RaspberryPi で赤外線アレイセンサ AMG8833(Grid-EYE)からデータ取得”, https://qiita.com/tm_nagoya/items/904ba8a23868ddcdcc54, 2021/6/13 閲覧

[4] @tm_nagoya, “赤外線アレイセンサ AMG8833(Grid-EYE)のデータを表示”, https://qiita.com/tm_nagoya/items/32d7e5becf73ba8a6110, 2021/6/13 閲覧

[5] @PoodleMaster, “【簡単】QR コードの作成と読み取り in Python”, https://qiita.com/PoodleMaster/items/0afbce4be7e442e75be6, 2021/6/13 閲覧

[6] @Brutus、” Raspberry Pi によるスクレイピング 環境構築（Selenium + Chromium ドライバ）”, https://qiita.com/Brutus/items/7381a13fa395f9b73855, 2021/6/13 閲覧
[7] 「Raspberry pi で selenium がやっと動かせた」, 分かりやすい技術ブログ, https://sun0range.com/information-technology/raspberry-pi-selenium/, 2021/6/13 閲覧
[8] 太田 二朗, “赤外線サーモグラフィによる パンデミック対策事例の紹介”, NEC 技報, Vol.63 No.3 2010 年 9 月 パブリックセーフティを支える要素技術・ソリューション特集.
[9] 濱谷尚志, 内山彰, 東野輝夫, “ウェアラブルセンサを用いた深部体温推定に関する一検討”, 情報処理学会研究報告. MBL, [モバイルコンピューティングとユビキタス通信研究会研究報告] 2014-MBL-72(4), 1-5, 2014-08-20
[10] Yamanoue, T., Oda, K., Shimozono. K. 2013. “An Inter-Wiki Page Data Processor for a M2M System, “ 2013, In Proceedings of the 4th International Conference on E-Service and Knowledge Management (ESKM 2013), Advanced Applied Informatics (IIAIAAI), 2013 IIAI International Conference on.(Matsue, Shimane, Japan, 31 Aug- 4 Sep. 2013) IEEE, Los Alamitos, CA. 45-50. DOI= https://doi.org/10.1109/IIAI-AAI.2013.48
[11] Yamanoue, T., Oda, K., Shimozono. K., “Experimental Implementation of a M2M System Controlled by a Wiki Network,” 2014, In Applied Computing and Information Technology, Studies in Computational Intelligence, Springer, Vol.553, 121-136.
[12] Takashi Yamanoue,"An Attempt of Automatic and Flexible Operation of Campus Equipment Using Bot Computing," SIGUCCS '21: ACM SIGUCCS Annual Conference, March 2021 Pages 30–35, https://doi.org/10.1145/3419944.3441163