

〔論文〕

WEB カメラを用いたマークシート式成績入力システムの試作

千田 陽介*¹

A Grade Input System Using Camera Images in an Optical Mark Recognition Sheet

Yosuke SENTA*¹

Abstract

It is difficult to properly manage student achievement and/or attendance in large classrooms. The author believes that the most important step toward addressing this issue is to digitize student information because digital data can be calculated accurately using some kinds of scripts or spreadsheet software. One such method is optical mark recognition (OMR), which requires high-tech equipment and is generally expensive. For a seamless application of such a system, the author developed an OMR set-up using a web camera. With a view to improving the camera's ease of use, it has been identified that the relative position between the camera and the OMR sheet and the corresponding brightness are not constant. Therefore, it is necessary to develop the system through stable marker detection and a correction algorithm that is on the detected markers. This paper describes these algorithms in detail, as well as their performance when the prototype system was first applied to the class of 2018. The result is a data-reading rate of approximately 3.5 sec per sheet, with about 5 percent of the sheets incompletely read.

Key Words : Optical Mark Recognition, Web camera, Geometric interpolation

1. はじめに

筆者は現在、情報ネットワーク工学科の「電気回路Ⅰ」を担当している。これは直流電気回路を題材に、数学を工学的道具として扱えることを目標にした一年の必須科目である。受講者数は一年全員及び再履修を含めおおよそ130名である。一方で筆者は「数学や物理は紙と鉛筆を用いて図を描いたり式変形を一行ずつ行ったりすることで理解を深めることができる」という教育理念を持っている。この理念が正しいかどうかはさておき、この理念の実行のため「電気回路Ⅰ」では毎回、紙による演習を行い、解答用紙の回収をもって出席としている。

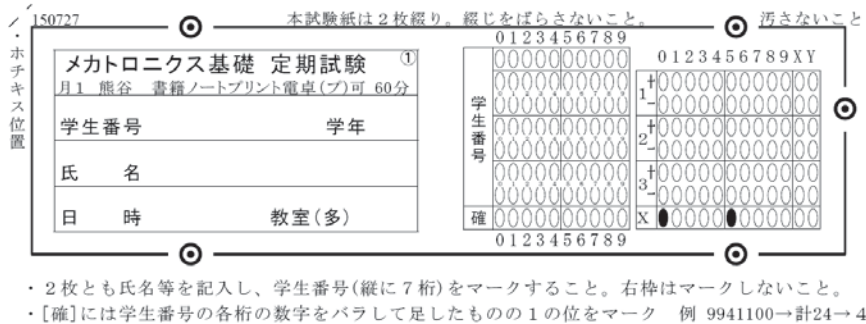
昨年（2017年）度、解答用紙から出席データへの変換はマウスとキーボードによる手動登録で行っていた。百人を越す数を間違いなく入力するためかなりの時間と集中力を要していた。そこで解答用紙の中に図1のようなマークシート部分を設けWEBカメラを用いて読み取るシステムを試作し、2018年度の授業に適用してみた。本報告はこのシステムの詳細と効果について述べるものである。

学生番号	日付		0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	
氏名		X	○	○	○	○	○	○	○	○	○	○	h	○	○	○	○	10
		(学生番号)	○	○	○	○	○	○	○	○	○	○	l	○	○	○	○	0-4
		Y	○	○	○	○	○	○	○	○	○	○	c	○	○	○	○	5-9
			○	○	○	○	○	○	○	○	○	○	e	○	○	○	○	⊗

・ 太く囲った枠内のみ記入してください
 ・ X は学生番号を上から順に塗りつぶしてください
 ・ Y は学生番号を 43 で割った値の小数点第 2 位
 例) 123456 なら 123456÷43=2871.069767
 だから 6 を塗りつぶす

Fig. 1 OMR sheet form of this study

*¹ 情報ネットワーク工学科
平成30年9月25日受理



- ・ 2枚とも氏名等を記入し、学生番号(縦に7桁)をマークすること。右枠はマークしないこと。
- ・ [確]には学生番号の各桁の数字をバラして足したものの1の位をマーク 例 99411100→計24→4

Fig. 2 OMR sheet form of Kumagai Lab. (Tohoku Gakuin Univ.)⁽⁴⁾

2. マークシート読み取りシステム

マークシート (Optical Mark Recognition: OMR Sheet) は解答の自由度を制限することで、高速かつ確実な電子化を可能にした入力法で、各種試験やアンケート集計においてよく用いられている。以前は専門の用紙と読み取り機器を要したが、スキャナとプリンタを用いて簡単に導入することができるようになり様々なソフトウェアが発表されている^(1,2)。

一方大学での授業、特に演習のような形態において、マークシートの利点はあまりないと考える。マークシートは性格上結果のみ記述するものであり演習に不向きであること、また特に近年は学内ネットワーク環境の整備に伴い Moodle⁽³⁾のような e-Learning を用いた方が運用や効果ですぐれているためである。しかし、現状の e-Learning システムはまだ試行錯誤的に図を描いたり式変形を行ったりすることは不向きなため、「電気回路 I」は電子化不可能と考えていた。そのような折、東北学院大学ロボット開発工学研究室 (熊谷研) の WEB ページ⁽⁴⁾で図 2 のような過去問を見かけた。図のように学生番号および付加的な情報をマーキングし光学的に読み取るよう設計されている。解答全体でなく、このように一部だけを電子化する方式は上述した紙と鉛筆を用いた学習を生かしつつ、数百人分の出席や採点結果を高速かつ正確に電子化するには有効だと考えた。

作者の熊谷によると該システムは、評価のエビデンスを取ることを主目的とし解答用紙全体を自動送り機能付きドキュメントスキャナ (Fujitsu ScanSnap) で取り込み、マークシート部分の解析結果に応じたディレクトリ・ファイル名で画像保存する運用であった。この運用はたしかに魅力的である。しかし自動送り機能のついたドキュメントスキャナは比較的高価であり、どの教員でも気軽に導入するシステムにならない。さらに筆者の経験では、演習プリントといった消しゴムを多用した紙は紙詰まりが発生しやすく、一回の授業分を取り込む間に何度も紙詰まりを解消するため逆に時間がかかってしまう。そこで出席や採点結果の記録のみに特化し、ドキュメントスキャナの代わりに WEB カメラを用いて、手動で一枚ずつマークシート記述部分を写して読み取るシステムの構築を試みた。

WEB カメラは安価で手軽に活用できる反面、紙面までの距離や角度の自由度が大きく、毎回同じ条件でマークシートを撮影することが難しい。斜め方向から撮影された画像は遠近法により歪んで見える。マーク状態を正しく読み取るには歪みを補正しなければならない。厳密に歪みを補正するためにはカメラの内外部に関する幾つものパラメータを正確に取得するキャリブレーション作業が必要である。カメラキャリブレーションに関する研究は幾つも行われているが^(5,6,7)、いずれも数枚の画像を取得する必要があると決して手軽とは言えない。当システムは手軽に使えることを目指しており、マークシート読み取りに特化することでキャリブレーション作業を必要としない歪み補正を実現した。

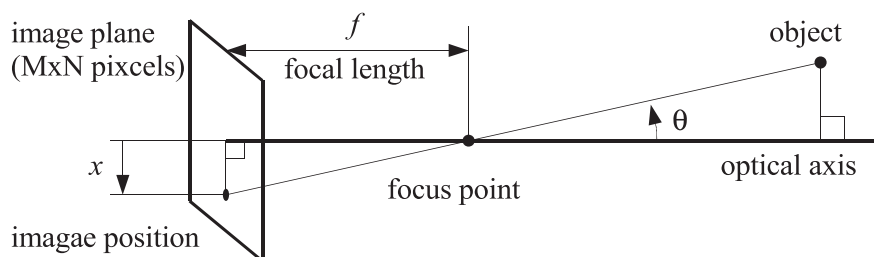


Fig. 3 Pinhole camera model

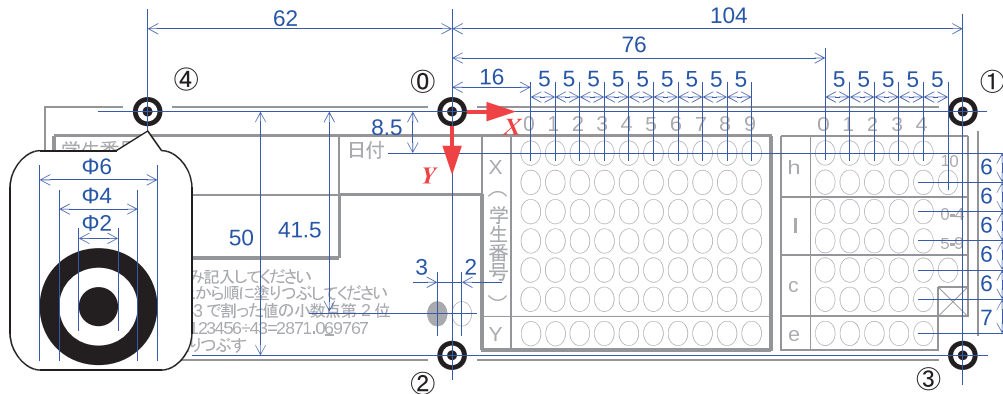


Fig. 4 Dimension of the Fig. 1's OMR sheet

3. カメラモデルと画像処理

読者の理解を助けるため、この章では画像処理の基礎的な事項を解説する。図3にピンホールカメラのモデルを示す。被写体は焦点を介して撮像素子に投影される。その関係は幾何学的であり、光軸から角度 θ 離れた物体が投影面に写る位置 x は焦点距離 f が分かれば簡単に求めることができる。投影面は $M \times N$ の受光素子が二次元に並んでおり個々の受光素子をピクセルと呼ぶ。ソフトウェアはピクセル単位でデータを取得する。そのため1ピクセルが何mm分かからなければ x の真値を知ることができない。ここで1ピクセル当たりの長さも焦点距離 f も同じ比例係数であり $f = 1$ と考えると θ を求めるのに必要なパラメータは一つで済む。また光軸から被写体、もしくは被写体間の距離を得るためには θ だけでなく焦点から被写体までの距離も必要である。しかし撮影対象が都合よく拡張されたと解釈できる類の問題設定の場合、ピクセル当たりの長さ、焦点距離、被写体までの距離といったパラメータを考えず距離を無次元化できる。本稿の場合も距離を絶対的な値でなく相対的な値として扱うため距離を無次元化できる。以後の議論やグラフで単位を示していないのはそのためである。

一般的なカメラでは $M \times N$ の受光素子それぞれに赤(R)、緑(G)、青(B)の輝度が0~255の値で計測される。RGB三次元のままだと情報過多で取り扱いが難しい。そのため

$$Y = 0.299R + 0.587G + 0.144B \quad (1)$$

という式⁽⁸⁾を用いて一次元0~255の輝度情報(グレースケール)に変換して活用するのが一般的であり本稿でもこの手法を用いている。

近年、画像処理ライブラリが充実しており、研究用のソフトウェア開発でもフリーのOpenCV⁽⁹⁾を使うのが普通となってきている。しかしOpenCVは様々な状況で使えるように完璧な動作を目指しており規模が大きい。本研究のように比較的単純なアルゴリズムを組み込む際は、逆に導入の手間といった問題の方が大きくなると考え活用しなかった。

4. マークシート読み取りアルゴリズム

4.1 全体的な手順

マークシートを読み取るには用紙上の塗りつぶし可能箇所が塗りつぶされたかどうかを調べればよい。塗りつぶし可能箇所は図4のようにあらかじめ決まっているため、全位置をデータベース化し、カメラ画像上のそれらの位置(ピクセル)すべての濃淡(輝度)を調べれば入力情報を取得できる。紙面とカメラ画像の位置関係はカメラ画像上における紙面原点 O の位置や x 軸、 y 軸の方向を始めとする状態量で決定する。その情報を得るためシート上には①~④の5つマーカを設置してある。

読み取りは、任意の解像度で取り込んだカメラ画像に対し(1)中央部 640×480 部分を抽出しグレースケール化してから、(2)マーカを検出。(3)①~④の対応付けを行った後、(4)紙面上の任意の位置とカメラ画像上の位置との対応関係を得て、(5)塗りつぶし可能箇所全ての濃淡を調べ、(6)学生番号等の情報に変換した後エラーチェックを行い(7)保存という流れで行う。以後の節ではこのうち(2)~(6)について詳細を述べていく。

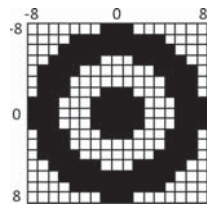


Fig. 5 Template of marker

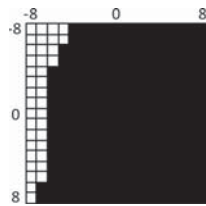


Fig. 6 Sample of failed

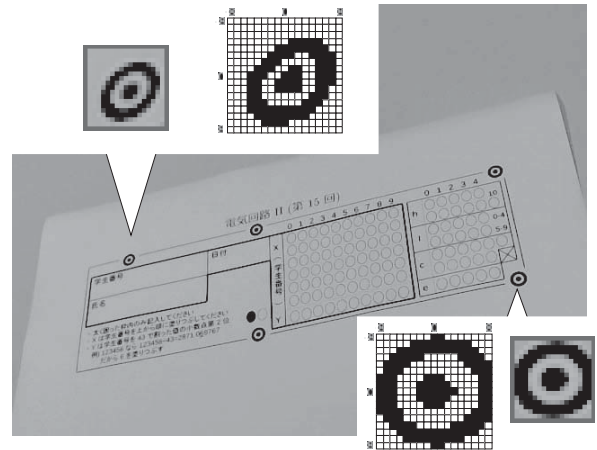


Fig. 7 Sample of camera image and binarized marker

4.2 マーカ検出

図4左下に示したようにマーカは直径比 1 : 2 : 3 の白と黒の円環とした。円形にしたのは紙面の向きの影響を受けないためである。これを図5のような17×17の二値のテンプレートを用いて検索する。検索は四隅8ピクセルを除いた $(640 - 16) \times (480 - 16) = 289536$ 点すべてに対して、次式で計算される類似度が大きいものを探すことで行った。

$$\begin{aligned}
 S_{XY} = & \sum_{x=-8}^8 \sum_{y=-8}^8 M_{XY}(x, y) \\
 & + \sum_{x=-8}^{-1} \sum_{y=-8}^8 M_{XY}(x, y) M_{XY}(-x, y) + \sum_{x=-8}^8 \sum_{y=-8}^{-1} M_{XY}(x, y) M_{XY}(x, -y) \\
 & + \sum_{x=-8}^7 \sum_{y=-8}^{-x-1} M_{XY}(x, y) M_{XY}(-y, -x) + \sum_{y=-7}^8 \sum_{x=-8}^{y-1} M_{XY}(x, y) M_{XY}(y, x)
 \end{aligned} \tag{2}$$

なお $M_{XY}(x, y)$ は、画像上の点 $(X+x, Y+y)^T$ とテンプレートの点 $(x, y)^T$ の明暗を比較し、同じなら1、異なるなら0を返す関数である。この式において、第一項は通常のテンプレートマッチングによる類似度を計算している。さらにマーカが対称な形状であることを利用し、第二項と第三項で縦横に、第四項と第五項で斜めに対称かどうか類似度の一部とした。式(2)の最大値は $17 \times 17 + 17 \times 8 \times 4 = 833$ である。明暗の判定は $(X \pm 8) \times (Y \pm 8)$ の輝度平均に対し、輝度が32以上大きいものを明、32以下のものを暗としている。これにより白紙部分等、単調の17×17ピクセル領域は明とも暗とも判定されないため類似度が0となる。しかし仮に机の色が暗色系だった場合、用紙との境界部分では明と暗が明確に分かれてしまう上、対称な形状のため類似度が400を超える時がある(図6)。そのような条件下でも正しくマークを検出するためにはこれよりも大きい値を閾値としなければならない。

このようなテンプレートとの類似度を用いた検索はスキャナで取り込んだり真上からカメラ撮影した画像に対してはかなり有効である。しかし前章で述べた通り今回はカメラの位置を厳密に定めなくても読み取れるシステムをめざしている。そのため少し斜めから撮影することもあり、その場合遠近法により手前のマーカと後方のマーカの大きさが異なる。図7にその一例を示す。図のように右手前のマーカと左奥のマーカは奥行きが異なるため、17×17で切り取った画像ではマーカの見え方が異なる。右手前のマーカの類似度が627点なのに対し、左奥のマーカは384点しかない。これは図6よりも小さな類似度であることから、このマーカを式(2)だけで検出することは不可能である。

そこで今回はマーカ形状が同心円であることを利用し、

1. 中心地点 $((X, Y)^T)$ が暗である
2. 中心から上下左右斜めの8方向において、7方向以上について、明⇒暗の切り替えが過不足なく1つ存在する

という二つの条件を満たしていない場合、式(2)の演算を行わず類似度を0にすることにした。画面内は当条件を満たしていない画素がほとんどのため、マーカ検出の精度を上げるだけでなく計算負荷を下げることにもなる。

用紙内にはマーカが5つ存在している。画面上289536点すべての類似度を計算し上位5位をマーカ位置とする。ただし同一マーカに対し、隣接するピクセルで上位を独占しないよう、二位以下のマーカは上位のマーカと20ピクセル

以上離れているものとした。さらに上位5位とはいえあまりにも類似度が低い場合、その画面内にはマークは発見できなかったことにして以後の処理は行わない。運用ではその閾値を250点とした。この値でほぼ誤検知なくマークが検出できている。

4.3 マーク対応

前節で得られたカメラ画像上の5つのマークはそれぞれ紙面上のマーク①～④のどれかに対応しているはずである。図4の赤い矢印で示した座標系において各マークの位置は $(0, 0)^T$, $(104, 0)^T$, $(0, 50)^T$, $(104, 50)^T$, $(-62, 0)^T$ と表現できる。前節では類似度の高い順にマーク検出したため何位のマークが何番のマークに対応しているかは不明である。しかし組み合わせはただか $5! = 120$ 通りなので、この中からどの組み合わせが一番適切か検索すればよい。

検索方法は各組み合わせにおいて、①～④と仮対応したカメラ画像上の位置を $M_0 \sim M_4$ として、まず

1. ①に対し③は①の右にある。すなわち $(M_3 - M_0) \times (M_1 - M_0) > 0$
2. ①に対し②は③の右にある。すなわち $(M_3 - M_0) \times (M_2 - M_0) > 0$
3. ①に対し④は②の右にある。すなわち $(M_2 - M_0) \times (M_4 - M_0) > 0$
4. ①から①, ②まで適度に距離がある。すなわち $|M_1 - M_0|^2 > 147^2$, $|M_2 - M_0|^2 > 70^2$
5. ①, ①, ④はほぼ同じ直線にある。すなわち $|M_4 - M_0|^2 - \{(M_1 - M_0) \cdot (M_4 - M_0)\}^2 / |M_1 - M_0|^2 < 17^2$

という幾何学条件が満たされているか確認する。条件1～3における演算子“ \times ”及び条件5における演算子“ \cdot ”は一般に二次元ベクトルの外積、内積と呼ばれているもので、それぞれ $(x_1, y_1)^T \times (x_2, y_2)^T = x_1 y_2 - x_2 y_1$, $(x_1, y_1)^T \cdot (x_2, y_2)^T = x_1 x_2 + y_1 y_2$ という計算を行うものである。条件4での閾値147と70は、直径6のマーク(図4)がほぼ17ピクセルに写る(図5)ことから、マーク間の距離104, 50はそれぞれ294, 141ピクセルになると計算でき、それに安全率0.5をかけた値としている。また条件5の左辺は①, ①を結んだ直線と④との距離を示しており、三点が同一直線上にあれば0になる。0に近い適当な値としてマーク直径のピクセル値17を閾値とした。4及び5の計算において二乗値で比較しているのは不要な平方根演算を省くためである。

これら条件を満たした組み合わせの中から、マーク④が①を介して①の反対側に位置する条件を用いて

$$L = \left| (M_4 - M_0) + \frac{62}{104} (M_1 - M_0) \right|^2 \quad (3)$$

を求め、この値が一番小さかったものを正しい組み合わせとしている。

4.4 紙面位置とカメラ画像位置との対応

4.4.1 線形補間

マークシート用紙をスキャナで取り込む場合、紙面の位置 $p = (p_x, p_y)^T$ と取り込み画像の位置 $d = (d_x, d_y)^T$ とは線形的な関係となる。そのため位置既知のマーク①: $(0, 0)^T$, ②: $(104, 0)^T$, ③: $(0, 50)^T$, ④: $(104, 50)^T$ が $M_0 \sim M_3$ に射影された事実を利用することで次のように簡単に演算できる。まず紙面上の点 $(p_x, 0)^T$, $(p_x, 50)^T$ はそれぞれ

$$d_u = M_0 + \frac{p_x}{104} (M_1 - M_0), \quad d_d = M_2 + \frac{p_x}{104} (M_3 - M_2) \quad (4)$$

に射影される。これより $p = (p_x, p_y)^T$ は

$$d = d_u + \frac{p_y}{50} (d_d - d_u) \quad (5)$$

に射影されることが分かる。この演算は線形補間と呼ばれるもので、コンピュータプログラムにおいてテーブル参照等によく使われる手法である。なおスキャナで取り込んだ画像はどの方向でも縮尺が等しく、かつ角度も正しく射影されるため、実際は M_0 と M_1 の位置さえ分かれば $p = (p_x, p_y)^T$ から $d = (d_x, d_y)^T$ への変換は可能である。しかし式(4, 5)のように M_2 や M_3 の位置も活用することで少し斜めから撮影したカメラ画像のように x 方向と y 方向の縮尺比が異なったり、垂直に交わらなくても対応できるようになる。

斜め方向から撮影したときの $p = (p_x, p_y)^T$ と $d = (d_x, d_y)^T$ の関係は、机板上での紙面原点の移動量 Δ_x, Δ_y , 回転角 r_x , 印刷もしくは撮影の縮尺 k に加え、カメラの視線と机板との角度 r_x , r_y やカメラ焦点と紙が置かれてある平面との距離 L といったパラメータにも関係する(図8)。その関係は媒介変数 $D = (D_x, D_y, D_z)^T$ を用いて、

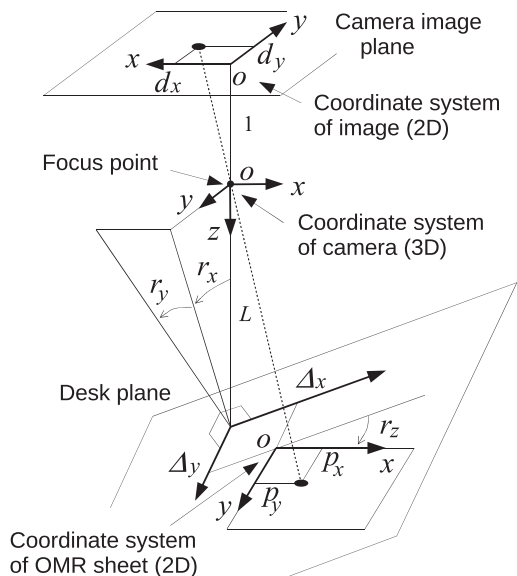


Fig. 8 Parameters of pinhole camera model

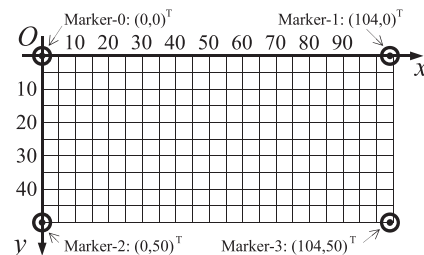


Fig. 9 Lattice pattern for simulation

$$D = R_x R_y (k R_z P + \Delta) \tag{6}$$

$$d = \frac{1}{L - D_z} \begin{pmatrix} D_x \\ D_y \end{pmatrix} \tag{7}$$

と表現できる。ここで $P = (p_x, p_y, 0)^T$, $\Delta = (\Delta_x, \Delta_y, 0)^T$ であり、また R_x, R_y, R_z はそれぞれ紙面座標系における x, y, z 軸周りの回転行列である。

図9のような格子状の模様を $\Delta_x = 20, \Delta_y = 10, r_x = 3^\circ, r_y = 5^\circ, r_z = -10^\circ, k = 0.85, L = 250$ という条件下で式(6, 7)に基づいて変換したもの(赤)と、式(6, 7)は $M_0 \sim M_3$ を求める時のみ使い、式(5)を用いて図形を変換したもの(青)を図10に示す。図のように真値(赤)に対して推定値(青)がずれているもののずれ量はわずかであり線形補間が正しく働いていることが見て取れる。一方図11はもう少し斜めから撮影したことを想定し、他の条件はそのままにして、 $r_x = 35^\circ, r_y = 20^\circ$ としたものである。図より推定値のずれが大きく、もはや正常にマークシートが読み取れる状態でないことがわかる。斜めからの撮影に強いシステムを実現するためにはより性能のよい補間アルゴリズムが必要である。

なお補間アルゴリズムを用いず図12のように用紙自体に補助的な目印を置く方法もある。このような用紙では②から③方向、もしくは①から③方向に向かい明暗が明確に切り替わる位置を検索すればよい。しかし個人的にはこのような補助目印は意匠的に美しくないと考えている。そこで次節に示すような性能のよい補間を考えた。

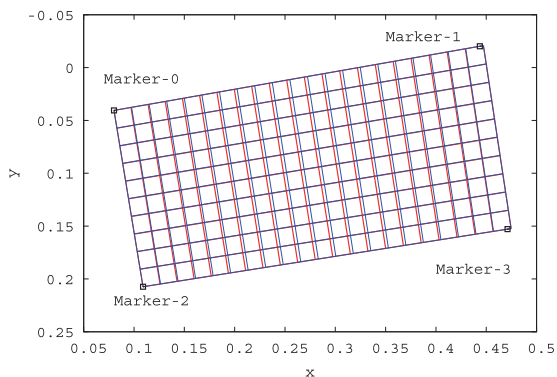


Fig. 10 View image of strict calculation (red) and linear interpolation (blue) under the condition of $r_x = 3^\circ, r_y = 5^\circ$

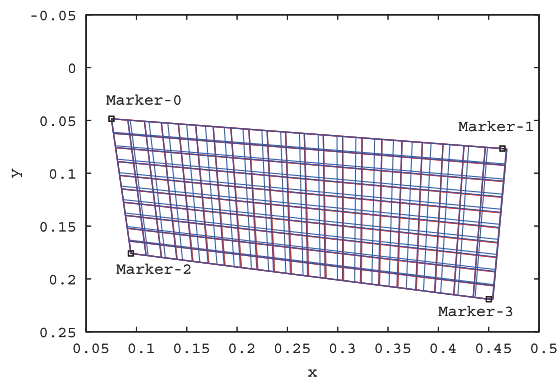


Fig. 11 View image of strict calculation (red) and linear interpolation (blue) under the condition of $r_x = 35^\circ, r_y = 20^\circ$

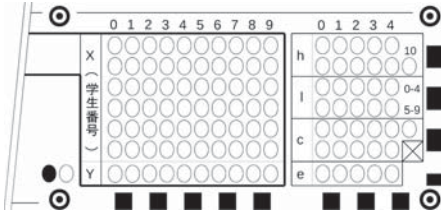


Fig. 12 OMR sheet with sub marker

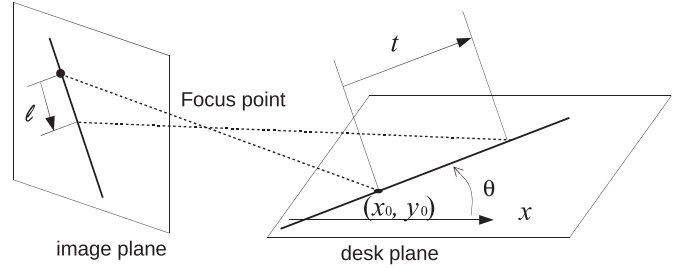


Fig. 13 Geometric relation between the actual length t and image length l

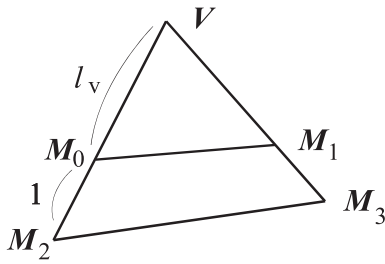


Fig. 14 Virtual vanishing point V

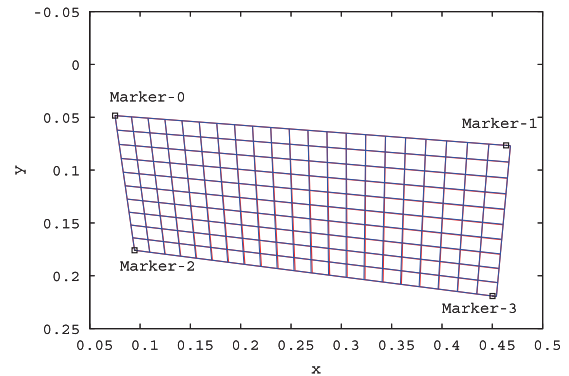


Fig. 15 Hyperbola interpolation (blue) under the same condition of Fig. 11

4. 4. 2 曲線補間

媒介変数 t を用いて紙面上の直線を $\mathbf{p} = (t \cos \theta + x_0, t \sin \theta + y_0)^T$ と θ, x_0, y_0 で表現し、式(6)にあてはめると、 $\mathbf{D} = (a_x t + b_x, a_y t + b_y, a_z t + b_z)^T$ という形になる。ここで $a_{\{x,y,z\}}, b_{\{x,y,z\}}$ は θ, x_0, y_0 はもちろん、図8で示した各種パラメータに依存する定数である。紙面の都合上具体的な式は省略するが、この \mathbf{D} に式(7)を適用すると

- 紙面上の直線は画面上の直線に変換される
- 紙面上における原点 $(x_0, y_0)^T$ からの距離 t と、画面上に投影された距離 l は以下のような形の式になる (図13)

$$l = \frac{t}{At+B} \tag{8}$$

ということが分かる。式(8)におけるパラメータ A, B は $a_{\{x,y,z\}}, b_{\{x,y,z\}}$ で表現される。すなわち上述した各種パラメータが分からなければ A, B の具体的な数値は求められない。しかしこれらのパラメータが分からなくても、位置関係が既知の点①~④に対する画面上の位置 $M_0 \sim M_4$ を用いることで A, B の値を推定することはできる。今回の場合、 xy 二次元の位置推定が必要なため、 x 方向のパラメータ A_x, B_x, y 方向のパラメータ A_y, B_y を個別に推定しなくてはならない。

このうち x 方向に関しては、まず M_0 から M_1 までの距離を 1 とした時の M_0 から M_4 までの正規化距離 $l = -|M_4 - M_0| / |M_1 - M_0|$ を求める。関数 $f(t) = t / (A_x t + B_x)$ は $f(0) = 0, f(104) = 1, f(-62) = l$ を満たすことから連立方程式を解けば、

$$A_x = \frac{1}{104+62} \left(104 + \frac{62}{l} \right), B_x = 104(1-A) \tag{9}$$

と解析的に A_x, B_x を求めることができる。

一方 y 方向は M_4 に相当する三点目が無い。そこで $M_0 \sim M_3$ の値を用いて消失点を推定し活用する。消失点の位置 V は M_0 と M_2 及び M_1 と M_3 を結ぶ直線の交点にあるため幾何的に求めることができる。しかし $M_0 \sim M_3$ の値は4.2節で述べた手法で取得しているため誤差を含み、厳密に V を計算する意義は薄い。そこで M_0 から V までの正規化距離 l_v を次のよう表すことができると仮定し計算を簡略化した (図14)。

$$l_v = \frac{|M_1 - M_0|}{|M_3 - M_2| - |M_1 - M_0|} \tag{10}$$

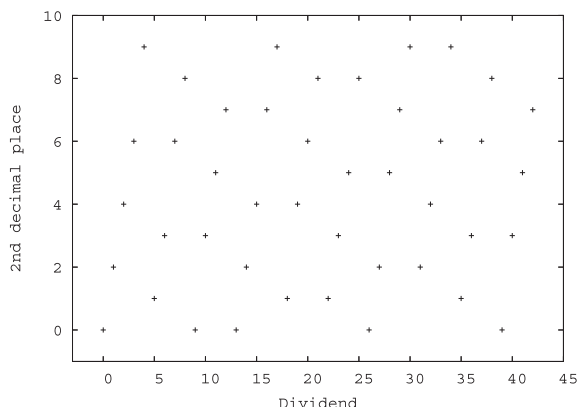


Fig. 16 Dispersion of the error detection code

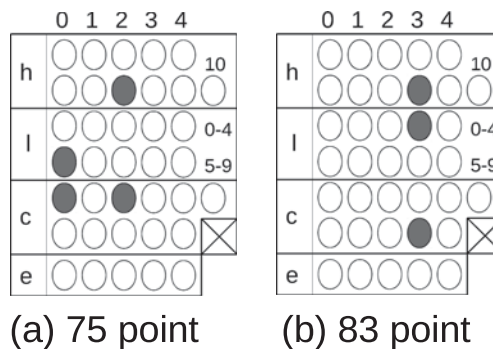


Fig. 17 Example of entering score and error detection code

関数 $g(t) = t / (A_y t + B_y)$ は $g(0) = 0, g(50) = 1, g(-\infty) = l_v$ を満たすことから, A_x, B_x は

$$A_y = 1 - L_v, B_y = 50L_v \tag{11}$$

と解析的に算出できる. ここで $L_v = 1/l_v + 1 = |M_3 - M_2| / |M_1 - M_0|$ である.

このようにして得た関数 $f(x)$ と $g(x)$ を用いることで紙面上の点 $(p_x, p_y)^T$ は,

$$\mathbf{d}_u = \mathbf{M}_0 + f(p_x) (\mathbf{M}_1 - \mathbf{M}_0), \mathbf{d}_d = \mathbf{M}_2 + f(p_x) (\mathbf{M}_3 - \mathbf{M}_2) \tag{12}$$

を用いて

$$\mathbf{d} = \mathbf{d}_u + g(p_y) (\mathbf{d}_d - \mathbf{d}_u) \tag{13}$$

に投影されると推定できる. ただし実装上は, 式 (10) の分母が 0 に近い時は当式を用いず式 (5) に切り替えている. 以上のアルゴリズムに基づいた図 11 と同じ条件での計算結果を図 15 に示す. 図のようにほとんど誤差が見えてないことが分かる.

式 (10) はあくまで近似であるため x 方向に比べ y 方向の推定は誤差が大きい. しかし y 方向はマーカ間の距離を短くし相対的に誤差が小さく見えるようにし, さらに塗りつぶし領域を縦長にして誤差を吸収しやすくする等用紙デザインを工夫することでその影響を小さくしている.

4.5 誤り検出

本システムは出席や採点結果という重要な情報を取り扱うため, マークシートの読み取りミス (記入ミスが原因のミスも含む) はできるだけ避ける必要がある. 読み取りミスを防ぐ手段に誤り検出符号がある. これは学生番号に対し一定のルールに基づいて算出した数値を学生番号と共にマーキングし, 読み取り時に検算するものである. 検算結果が異なれば読み取りミスが発生していることが分かる. 熊谷のシステムでは誤り検出符号として学生番号各桁の和の一行目を用いている. 和 (チェックサム) は学生がその場で数値を計算できる利点はあるものの, マークシートの性質上読み取りミスは +1 もしくは -1 となる横ずれが原因になることが多く学生番号とチェックサムの二か所同時に発生した場合発見できない. 誤り検出符号は学生番号との相関が見えにくく, 0 ~ 9 の数字が万遍なく現れ, その場で計算できることが理想である. そのような観点から学生番号を 43 で割った際の小数点二桁目の数字を使うこととした. 43 という数値自体は深い意味はなく, 35 ~ 45 人のクラスの中で規則性が見えにくいといった条件から適当に選定したものである. この計算で得られる符号は $100/43 = 2.32558 \dots$ ずつ増加される 43 周期の数であるが, 図 16 のように規則性が適度に隠れている. なお 6 桁の学生番号を 43 で割ることは手計算可能だが煩雑である. 幸い, 筆者が担当する「電気回路 I」の授業は関数電卓を活用するため, 電卓を用いて計算すればよい. またこの値は固定値なため初回は手計算なりスマートフォン等の電卓機能を用いるなりして数を算出し, 以降はその数を覚えておけばよく, 大きな問題とはならないと考えている.

図 1 のマークシート用紙において学生番号は X 欄に, 誤り検出符号は Y 欄にマークする. 同 h, l, c 欄は採点結果等を教官が記入するためのものである. h 欄は 100 を含む 10 の位, l 欄が 1 の位で 0 ~ 109 までの数値を記入できる. c 欄は h, l に対する誤り検出符号である. 教官がマーキングするうえで面倒な計算を必要とするような符号は望ましくない.

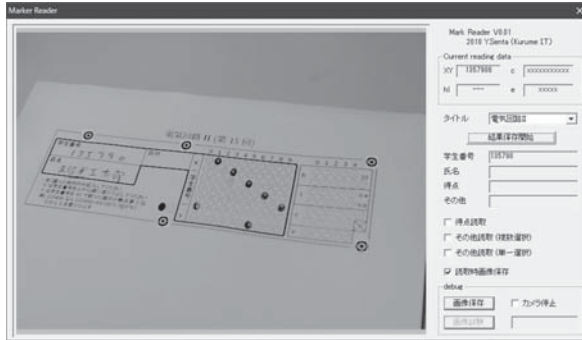


Fig. 18 Overview of prototype application

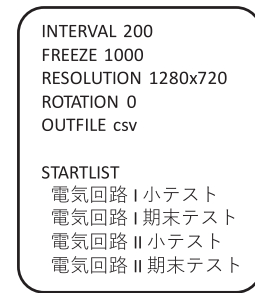


Fig. 19 Sample of config file

Table 1 Processing time of Optical Mark Recognition

	Average	min	max	SD
Capture Image	0.75msec	0.42msec	2.78msec	0.21msec
Change to 640 × 480 gray scale	3.08msec	1.43msec	4.30msec	0.48msec
Search 5 markers	33.1msec	17.5msec	47.2msec	6.2msec
Corresponding markers	5.44usec	3.20usec	10.88usec	1.38usec
Read and check the OMR	0.91msec	0.66msec	2.01msec	0.25msec

そこで単に上段は同じ列の h, l 欄が 1 つ塗りつぶされていた場合、下段は 2 つ塗りつぶされていた場合にそれぞれマークするという視覚で簡単に導き出せるルールとした。図17にこのルールに則ったマーキングの例を示す。

最後の e 欄は多目的に使える項目としている。この欄は誤り検出符号を実装してないので活用には注意が必要である。

5. 試作ソフト

以上のアルゴリズムを実装したアプリケーションを作成した。図18にその外見を示す。本プログラムは Microsoft Visual C++ 2008 Express Edition で作成した。プログラムは一定間隔でカメラ画像取得と画像解析を繰り返し、誤り検出符号による検算まで通過した場合（以下「読み取り完了」と表現）、操作者に視覚的に読み取り完了を伝えるため画面上下に緑もしくは赤の帯を表示するとともに画面を一定時間静止させている。また、「学生番号、氏名」という書式のファイル“StudentList.csv”を作っておくと、読み取り完了時に該当する学生番号の氏名を検索し画面に表示するようにしている。これにより操作者の視認による誤入力防止も可能となる。「結果保存開始」ボタンを押すと、読み取り完了時点で、読み取れた学生番号を Result ディレクトリ以下に csv もしくは txt ファイルとして保存する。また必要に応じ解析に用いた画像も jpg ファイルとして保存できる。保存画像内に手書きの氏名や学生番号が収まるようマーカの位置は工夫されている。

ソフトウェアの動作（カメラの読み取り間隔、読み取り完了時の静止時間、出力ファイル形式、タイトル候補等）は図19のような設定ファイル（MarkReaderCfg.txt）で指定できる。詳細はソフトウェア付属のドキュメントを参照されたい。なお設定項目 RESOLUTION はカメラの解像度である。プログラムはここで指定した解像度で画像を読み取り、中央 640 × 480 ピクセル部分を抽出して画像解析する。そのためこの解像度を大きくすればするほどズームしていることになるので遠隔から撮影することが可能となる。

プログラムの各所に QueryPerformanceCounter() 関数を設置し、カメラ画像の取り込み、640 × 480 のグレースケール画像へ変換、マーカ検出、マーカ対応、塗りつぶし状態の取得にかかる時間を計測してみた。144回の計測結果を表1に示す。ただし計測に用いた PC は Core i4-3930 (3.2GHz)、カメラは Logicool C922 (1280 × 720 モード) である。表から一回の処理にかかる時間は最大でも 57msec しかかかっておらず、実用十分な速度で処理が行えていることが分かる。なお、本結果によるとカメラ画像の取り込みはわずか 1 msec しかかかっておらず、使用したカメラの性能 (60fps) と一致していない。これは今回使用したウェブカメラライブラリ (EWCLIB⁽¹⁰⁾) が常時別スレッドでカメラ画像をバッファに取り込んでおり、カメラ画像取り込み命令 EWC_GetImage() は単にメモリコピーを行っているだけのためこのような値になったのではと推測している。

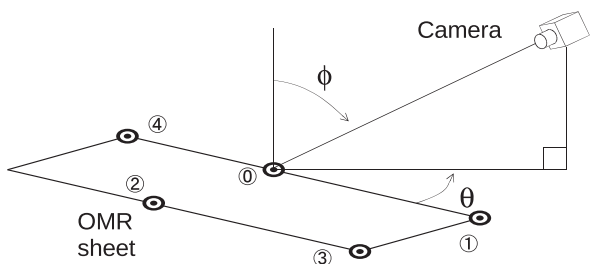


Fig. 20 Yaw angle θ and tilt angle ϕ

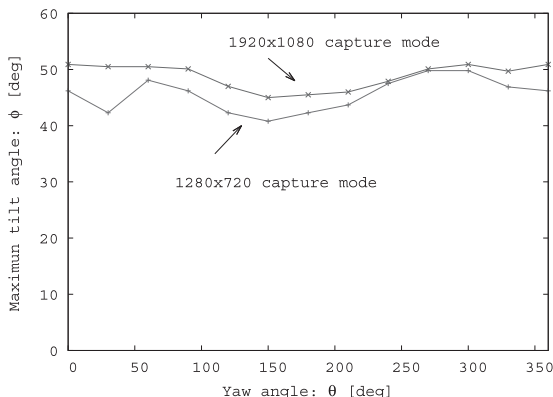


Fig. 21 Maximum readable tilt angle

これまで何度も述べてきたように当システムは、多少斜めから撮影した画像でも動作することが期待されている。その性能を評価するため、**図 20**のように紙面回転角 θ 毎の最大読み取り成功角度 ϕ を計測してみた。結果を**図 21**に示す。角度 ϕ を次第に大きくすると今まで成功していた読み取りができなくなる。読み取りできなくなる理由は遠方マーカの検出失敗であった。これは角度 ϕ が大きくなると遠近法により、遠と近のマーカの大きさの差が大きくなるためであろう。図 21においてカメラモードが 1280×720 より 1920×1080 の方が最大読み取り可能角度が大きい。これは解像度を大きくすることは望遠率を上げることを意味し、その分圧縮効果により遠近のマーカの大きさが均一になるからだと考えられる。いずれにせよ少なくとも見積もっても 40° という値は十分な大きさであり、だいたい真上から撮影するようにカメラ設置するだけでまず間違いなくマークシートを読み取ることができる。

6. 実証実験

当システムを 2018 年度前期の「電気回路 I」の授業に適用してみた。この授業では毎回授業開始時に前回の授業内容について A 4 一枚の演習プリントを配布し 15 分解決してから、解説、自己採点の後回収し出席としている。その紙に**図 1**のマーク記入欄を設けた。マークは X と Y 欄のみ使用し、h, l, c 欄は用いてない。**表 2**に全 15 回の読み取り結果を、**図 22**に一枚当たりにかかった読み取り時間と読み取りの成功率の推移を示す。図より成功率は最初は 80% 以下だったものが回数を重ねる毎に次第に上がっていき、最終的には 95% 前後を推移していることが見て取れる。第一回

Table 2 Performance evaluation result of the OMR system by Electric Circuit I lesson

Times	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Date	4/12	4/19	4/26	5/10	5/17	5/24	5/31	6/7	6/14	6/21	6/28	7/5	7/12	7/19	7/26
Total Num.	123	125	127	125	111	125	119	127	127	119	121	115	125	116	119
Read Time	10:16	7:26	8:54	7:19	5:35	6:40	6:46	7:06	7:32	6:28	6:52	6:03	6:46	6:40	5:39
Failure Num.	28	20	16	16	5	3	8	5	7	7	7	3	5	8	5

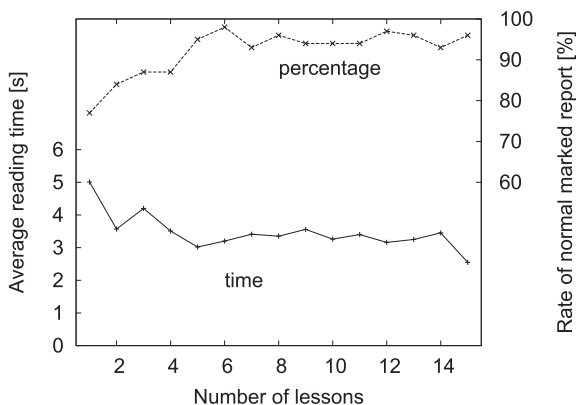


Fig. 22 Average reading times and mark reading success rate of the lesson

の読み取り成功率が悪いのは、どのようにマーキングすればいいかはマーク記入欄に文面として記載していることから説明しなかったためである。第一回目の成功率が予想より悪かったため二回目の授業ではマークの記入方法についてY欄の計算方法も含め説明し、また以降マーク不備だったプリントはどこがおかしいか指摘して返却することで次第に成功率が上がっていった。ただし常に5%前後はマーク不備で読み取れないものが存在した。不備の内容はY欄をマークしないもの、Y欄の値が間違っているもの、一つ左の数字をマークしているものなどが目立った。左をマークしてしまうのは、当書式は0から始まっているのを間違えて1から数えてマークしてしまったのではないかと推測している。

一方読み取り時間は一枚あたり3.5sec程度となった。前章で述べたようにシステム自体はマーカの読み取りに57msしか必要としない。そのためこの値は(1)未読の束から一枚取り出しカメラの画角に入れ、(2)マーカを読み取らせた後、(3)画面上に表示された学生の氏名と紙面上に書かれた氏名が一致しているか確認しつつ既読の束に重ねる、という一連の作業を人が行うための時間である。回毎に±1sec程度のバラツキがあるのは、マークが薄い等読み取りにくいプリントに対し光源の角度を調整してなんとか読み込ませようとした作業の時間だと考えている。

試作したソフトは汎用性を重視したため、読み込んだ学生番号をダンプするだけである。そのため運用ではこれを表計算ソフト等に取り込む必要がある。表計算ソフトに取り込む方法として様々な方法が考えられる。今年度筆者はcsv (comma separated values) ファイルで管理し、以下のようなcygwinのコマンドでデータを更新した。

```
$ sed 's/$/,A/' listA.csv | sort | uniq > a.csv (B, C も同様に)
$ cat a.csv b.csv c.csv | sort -n | join -t , -a 1 -e D -o auto currentlist.csv -> newlist.csv
$ rm currentlist.csv
$ mv newlist.csv currentlist.csv
```

なおlistA.csv, listB.csv, listC.csvは、回収したプリントの内容によってA, B, C 3つのグループに分けて保存したものである。またcurrentlist.csvは始めソートされた学生番号が並んでいるもので、以後これらのコマンドを実行するたびに、学生番号の後ろにカンマ区切りでA, B, CおよびD (欠席) が付与されていく。なお表計算ソフトとしてMicrosoftのExcelを使うのであれば、VBA (Visual Basic for Applications) を使うとより使い易くなるのではと考えている。

当システムが具体的にどのくらい時間が節約できるか確認するため、期末試験の結果(採点后h,l,c欄を記入)に関し、昨年同様手動でExcelに点数を入力した時の時間と当システムを用いて電子化した時間とを計測してみた。なお解答用紙は学生番号順に並んでおらず、手動入力ではExcelの学生リストの中から該当の学生番号を探し数値を入力した。その結果135人分の入力についてマークシートを用いると14分49秒かかったのに対し、手動入力では37分40秒かかった。本番の成績入力という緊張感が無かったためか思った程時間が短縮されなかったが、それでも3倍近い値となっている。

7. おわりに

回収したプリントの学生番号を電子化するために、WEBカメラを用いたマークシート読み取りシステムを試作し、半期にわたって運用してみた。WEBカメラの画角に一枚ずつ手動で入れるため、自動スキャナ等に比べ時間がかかる欠点はあるが、紙詰まり等のトラブルが発生せず、また安価に構築できる利点がある。カメラと紙面との正確な位置合わせを行う手間を省くため、5個のマーカから全塗りつぶし可能箇所の位置を推定している。推定アルゴリズムを工夫することで紙面鉛直軸から±40°斜めから撮像しても読み取れるようになった。

当システムを2018年度前期の「電気回路I」の授業で適当してみた所、全15回を通して読み取り成功率がほぼ95%に収束した。残念ながら100%にはならなかったが、センター試験でも受験番号の誤マークが7000件、全体の0.2%あるという報道もあり⁽¹⁾、ある程度の成功率で妥協するのが現実的だと考えている。ただし十分に方法を周知させれば98%くらいまでは上がるのではとも考えており、来年度の授業で確認したい。

半年間の試用を通しシステムの性能はおおむね満足しているが、唯一感じた欠点は学生の顔が見えにくくなったことである。昨年はExcelにA, B, Cを入力していたが、過去の結果を見つ「この学生はがんばっている」とか「この学生は手を抜くようになった」等、自然と氏名を覚え、授業中に目をかけていたような気がする。当システムにより効率化された分そのようなことが無くなった。これがいいことか悪いことは判断が付いてないが、今後考察を深めていく必要があるだろう。

最後に当システムは<http://sentlab.kurume-it.ac.jp/research/MarkReader/index.html>にて公開してい

る（ただしソフトウェアは学内限定で公開）。読者の教務の助けになれば幸いである。

文 献

- (1) らくらくマークシート, <http://www.geocities.jp/dnsjb294/easymark/e005-1.html>
- (2) マークシート読取君, <http://www.magnolia.co.jp/products/utility/mread4/index.htm>
- (3) Moodle, Open source learning platform, <https://moodle.org/>
- (4) 熊谷, 東北学院大学ロボット開発工学, <http://www.mech.tohoku-gakuin.ac.jp/rde/>
- (5) Z. Zhang, "A flexible new technique for camera calibration", IEEE Trans. on Pattern Analysis and Machine Intelligence, 22(11), pp.1330-1334, 2000
- (6) 田嶋, "カメラキャリブレーション手法の改良", 大分大学工学部技術部報告書 pp.66-79, 2014
- (7) 新妻, R. Prasanna, 金谷, "反復を要しない射影変換の高精度解析法", コンピュータビジョンとイメージメディア(CVIM) 2010-CVIM-170(55), 1-8, 2010
- (8) RECOMENDATION ITU-R BT.601-5, "Studio Encoding Parameters of Digital Television for Standard 4:3 and Wide-Screen 16:9 Aspect ratios", 1982-1995
- (9) OpenCV library, <https://opencv.org/>
- (10) EWCLIB <http://www.geocities.jp/in.subaru/ewclib/>
- (11) "センター試験「うっかり受験生」を23年間救済", 読売新聞, 2006. 5. 25