

分散型画像処理教育支援システムの開発

古川 達也[†] 松尾 匡章[†] 宮崎 純生^{††}
堂 園 浩[†] 相知 政司[†]

画像処理技術は、様々な分野の研究開発の場において重要視されている。そこで筆者らは、基礎的な画像処理について体験学習することができる、Java RMI 技術を援用した処理分散型教育支援システムを開発した。また、このシステムは本学の学生実験に援用され、その際の受講者のアンケート結果から良好な評価を得ることができたので、これとあわせて報告を行う。

Development of Distributed Educational Support System for Learning Image Processing

TATSUYA FURUKAWA,[†] MASAKI MATSUO,[†] SUMIO MIYAZAKI,^{††}
HIROSHI DOZONO[†] and MASASHI OHCHI[†]

Image processing has become useful and important technology in various research and development fields. We have developed the distributed educational system based on Java RMI (Remote Method Invocation) technology to make it possible for a novice user to experience the practical digital image processing with ease. The system was adopted to a curriculum of a student laboratory in our Department of Electronics, Saga University. The questionnaires for the students, who engaged in the subject, were filled out and examined. First we will present the architecture of the present system, in which Java RMI technology has been introduced to implement the end user programming, evaluate the system and discuss based on students' questionnaires.

1. はじめに

一般的に視覚情報は、人間の外界に対する最大のインタフェースであり、これに多くの情報を付与できる画像処理技術は、映像技術だけではなく、医療、交通等高い信頼性が求められる分野にも応用されるに至っている。

したがって、今後、画像処理技術は、コンピュータ関連の研究開発に従事する人だけでなく、それ以外の様々な分野の研究者、技術者や学生にとって重要な技術であり、業務経験、予備知識や計算機環境等の制限なしに、本技術を体験・修得できる環境整備が必須である。

従前、画像処理の体験学習ができるシステムとして、日本では VIEW-Station¹⁾ が有望であったが、動作ハードウェアならびに OS が限定される、ソース・プログラム配布の限定使用ライセンス、1 ホスト 1 ユーザ

クライアントの動作制限事項、すでに公のサポートが打ち切られた等の理由で実用に供さない状況にある。一方、インターネットの普及とともに現れた Java 言語を用いた画像処理²⁾ や CAI システム³⁾ も提案されているが、例題画像の単発的な処理の例示にとどまっているにすぎない。

これらを鑑み、筆者らは画像処理の教育支援システム⁴⁾ を独自に開発してきた。本システムは『技術は耳で聴くより目と手で体験の方が習得速度が速い』という設計思想の基に、初学者ならびに非専門家向けに画像処理技術の体験学習を目的として開発された。

また、このシステムは、使いやすさや理解しやすさにも十分考慮したうえで、単発的な画像処理を行うだけのものではなく、種々の画像処理プログラムをカスケード化することによって画像抽出や合成等の実用的な連携処理を可能とし、さらに、利用者が自ら画像処理プログラムを実装できるような機能も備えている点で従来のシステム^{2),3)} とは大きく異なる。

本システムを実際に、平成 10 年度から本学部電気電子工学科の 2 年時学生実験に援用し、平成 11 年度の受講生対象にアンケートを実施し、いくつかの本シ

[†] 佐賀大学理工学部

Faculty of Science and Engineering, Saga University

^{††} 日本電信電話株式会社

NTT Corporation

システムに対する評価を得ることができたので、本論文では、これとあわせて本システムの概要について報告する。

2. 画像処理教育支援システム

教育支援システムに与えられた課題は、操作が簡単で初心者でも扱えるようなユーザ・インタフェースを実現すること、初学者が実際に利用するまでの段階で難解かつ煩雑な作業のため断念してしまうのを避けること、同時に多人数に対して動作させることができること、種々の計算機環境で動作すること、成功例だけでなく、失敗例も体験できること等があげられる。以下に、Java 言語で実装された本システムの特徴を示す。

2.1 ユーザ・インタフェース

本システムは図 1 のような GUI (Graphical User Interface) を採用しており、ボタン群は処理の種類や役割によってそれぞれまとめられているため、本システムの全容を理解しなくても、簡単な操作が可能となっている。そのため、コンピュータに不慣れな学習者にとっても問題なく使用できる。

また、UNIX や MS-DOS といったコマンドライン形式に慣れている利用者のために GUI だけでなく、図 2 のような CUI (Character User Interface) も実装しており、処理はメイン・ウィンドウ上部のテキスト・フィールドに書き込むことで行われる。

コマンドライン・モードの基本的な書式は以下のとおりである。

% [回数] <コマンド名> [オプション]

回数を省くと 1 回の処理が行われる。オプションは処理ごとに決まっており、不明なオプションの場合、そのオプションは無視して処理が行われる。また、回数が指定できることで複数回行う処理も一括して行うことができ、作業効率も向上するものと考えられる。

2.2 実装した画像処理

現在、本システムで実装されている画像処理は以下のような基本的なものである。

- ノイズフィルタ (平均移動法, Median filter, Max/Minimum filter, Edge preserved smoothing)
- 画像の抽出 (閾値処理, ラベリング, 抽出, 合成)
- 輪郭抽出 (Gradient, Laplacian, テンプレートマッチング, 細線化処理)
- FFT (FFT, Highpass filter, Lowpass filter, 逆 FFT)
- その他 (プレーン分割, 反転, モノトーン)

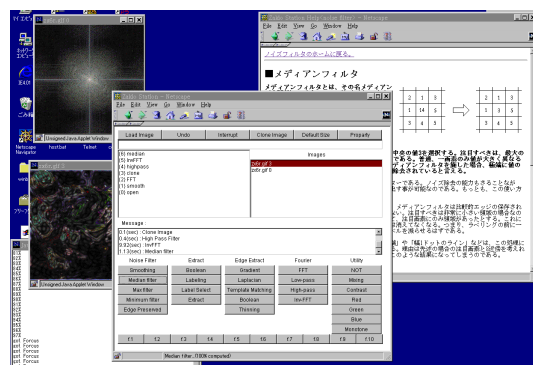


図 1 本システムを実行中の様子

Fig. 1 View of practical operations in proposed system.

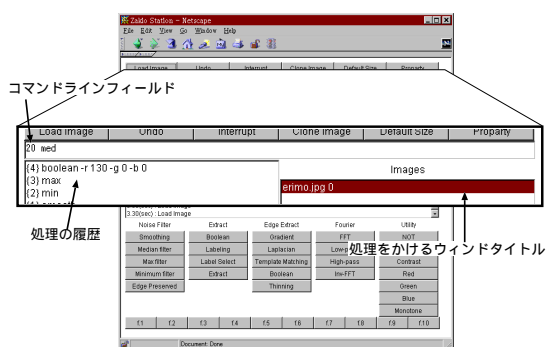


図 2 CUI 部の様子

Fig. 2 Character User Interface.

2.3 画像処理クラスライブラリ

2.2 節で示した処理は、他のクラスおよびメソッドと依存関係をなくし、static 宣言しているの、java.lang.Math と同様に自作のプログラムからも利用可能である。また、このクラス群は 2.4 節で述べるリモート・メソッドにも利用することができる。

これによって、単に本システムによって学習するだけでなく、プログラミングに応用してゆくことで、画像処理に対する興味ならびに知識の増加がさらに期待できる。

2.4 リモート・メソッド

画像処理の種類は非常に多く、すべてを実装するというのは、現実的ではない。たとえ考えうる限りの処理を用意したとしても、利用者が望むすべての処理を実装できる保証もない。

そこで、実装されていない処理のプログラミングは学習者に委譲してしまい、学習者が実装したプログラムをサーバ・システム側から実行できるような仕掛けを考案した。これによって、以下のような利点を得ることができる。

- 利用者が実装すれば、どのような処理でも本シス

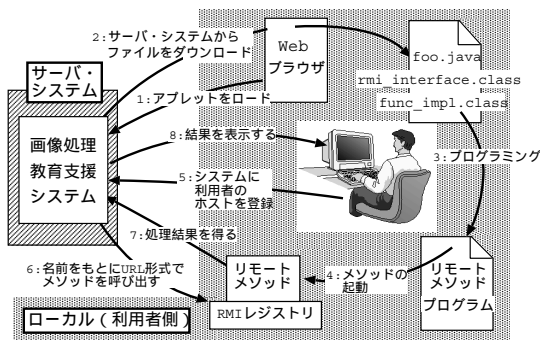


図3 リモート・メソッドの主な流れ

Fig. 3 Procedures in practical operations in remote method.

テム上で動作させることが可能。

- 既存のフィルタ以外にも新たに考案した処理をフィルタとして利用できる。
- システムにすでに実装されている処理も読み込んで実行することが可能であり、自動化のためのフィルタの評価等も行える。
- 演習課題として、学生実験の際等、受講者にプログラムさせることが可能。処理のメソッド部分のプログラミングだけで画像に処理を施すことができるため、比較的手軽に取り組むことができる。
- 本システムを採用した指導者側から見て、たとえば教えたい処理が未実装であっても、その部分を自作するだけで利用することが可能である。

2.4.1 リモート・メソッドの流れ

本システムは、メソッドの実装に手間がかかったり、複雑なセットアップが必要になったりしないような構成を考えた。実際には必要なスケルトン・ファイルをダウンロードし、処理部分のプログラミングを行えば、動作するように設計した。リモート・メソッドは Java RMI⁵⁾ を利用しており、図3のような流れで実行される。以下にその手順を示す。

(1) Webブラウザを利用して、本システムのアプレットをロードし起動する。(2) サーバ・システムから必要な3個のファイル(`foo.java`, `rmi_interface.class`, `func_impl.class`) をダウンロードする。(3) ダウンロードした `foo.java` というスケルトン・ファイルを編集してプログラミングを行う。残りのファイルは、`foo.java` をコンパイルする際に必要となる。(4) `rmiregistry` というネーミング・サービスを行うプログラム(JDK1.1以降に標準添付) を利用者側のホストで別途起動したうえで作成したプログラムを動作させる。この段階でネーミング・サービスにバインド・ネームが登録される。(5) システムのアプレット上か

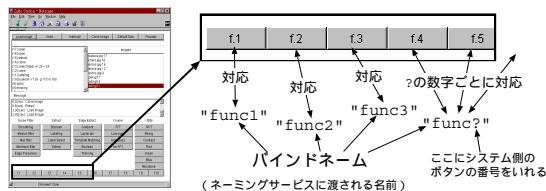


図4 ボタンとバインド・ネームの関係

Fig. 4 Relationship between buttons and bind-name.

ら利用者側のホスト名を登録する。この段階でサーバ・システム側は、ネーミング・サービスに問い合わせ、メソッドがあるかチェックする。(6) 図4に示すリモート・メソッド用のボタンをクリックすると、利用者側ホストで起動しているメソッドが呼び出される。

2.4.2 リモート・メソッドのプログラミング

リモート・メソッドを利用するためにプログラミングを行う際には、サーバ・システム側からダウンロードするスケルトン・ファイル `foo.java` の以下に示す部分を編集すればよい。ただし、プログラム中の行番号は、`foo.java` の実際の行番号である。

```

10 : public int[] function(int w,int h,int pic[])
11 :     throws RemoteException {
12 :     for(int y=0;y<h;y++){
13 :         for(int x=0;x<w;x++){
14 :             //ここに処理を書く
15 :         }
16 :     }
17 :     return pic;
18 : }

25 :     foo obj = new foo();
26 :     Naming.rebind("func1", (func_impl)obj);

```

*ただし、int w (画像の幅), int h (画像の高さ),
int pic[] (画像のピクセルマップ)

プログラミングは、実現したい処理を14行目の位置に挿入すればよい。さらに、26行目の"func1"がバインド・ネームであり、図4に示すようにボタンと対応しており、複数のプログラムを登録したい際には、この番号が衝突ないように登録する。

また、クラスに処理名を付けたい場合や同一ディレクトリでいくつものプログラムを動作させたい場合には、`foo.java` のファイル名ならびにそのソース内の `foo` という文字列を任意の文字列に変更することができる。

2.5 オンラインヘルプ

本システムは、初学者・非専門家向けの教育支援システムであるので、処理の参考となるヘルプはきわめて重要である。初学者の場合、得られた結果から、どの部分に注目すれば行った処理の特徴をつかむことができ

表 1 処理時間の計測に用いたマシン

Table 1 Machine specifications and OSs of used PCs.

マシン No.	CPU (Clock[MHz])	OS	RAM
1	Pentium (150)	WinNT	64 MB
2	Pentium (150)	WinNT	64 MB
3	Pentium III (500)	Win98	128 MB
4	Pentium III (500)	Linux	128 MB

1はAppletViwer(JDK1.1.8), それ以外はNetscape 4.06にて実行

表 2 各処理に要する時間(単位: sec)

Table 2 Required processing time [sec].

マシン No.	1	2	3	4
メディア	0.6	1.1	0.3	4.5
Edge Preserved	9.9	61.6	6.6	26.3
Gradient	1.1	1.1	0.2	3.4
Template Matching	3.4	20.2	2.2	6.7
FFT	7.7	8.9	1.2	21.2
逆FFT	4.5	6.6	1.1	20.2
モノトーン	0.2	0.4	0.0	0.3

るかどうか分からない場合が多い。また、ダイアログ等も、どのように操作すればよいのか分からない場合も多く、そのような際、手助けとなるように、HTML形式でリンクブルに書かれたヘルプを準備した。

3. 処理 負 荷

Java 言語でプログラムを開発した場合、実行時間を問われることが多い。実際、システムを運用する際に処理時間が長いと、利用者は嫌気をさして使わなくなる。そこで、表 1 に示す従来開発を行ってきた PC と最新の PC を用いて計測した処理時間を表 2 に示した。ただし、結果は OS の負荷状態によっても若干の影響が出ることを付け加えておく。被処理画像は 256×256 [pixels] サイズの JPEG ファイルで行った。なお、Applet は総計サイズが 176 KB 程度であるので、Web サーバへのネットワーク負荷もそれほど大きくはならない。

今回は比較的重い処理を行ったが、表 2 から、最新鋭のマシンでなくとも十分に使用に耐える結果が得られているのが分かる。近年のコンピュータ性能の急速な向上を考え合わせれば、今後、運用上、処理時間は問題ないといつてよいほどである。

4. 学生実験とアンケート

本システムを平成 10, 11 年度の本学電気電子工学科 2 年時学生実験の中の一テーマに援用した。今回は、表 3 に示す仕様の PC を用いて、1 回の実験で 2 班(1 班 3~4 人)同時に進行する形式で行った。

図 5 は、実験時のネットワーク構成である。Web

表 3 実験システムのスペック

Table 3 Specifications of machines in a student laboratory.

	A 班	B 班
CPU	Pentium 166	Cyrix 133 (166)
RAM	80 MB	同左
OS	Windows 98	Windows 95
解像度	1024 × 786	同左
ブラウザ	Netscape Navi. 4.06	Netscape Com. 4.6
プリンタ	EPSON MJ-930C	同左
モニタ	15 inch (NEC)	同左

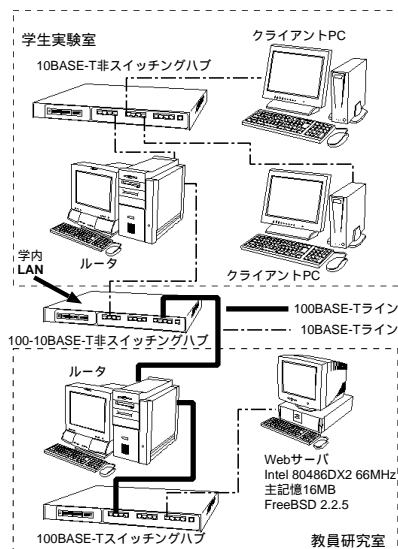


図 5 ネットワークの構成

Fig. 5 Network configuration.

サーバは、他所に置かれ、100-10BASE-T 自動切替え非スイッチングハブを介して実験室内のサブネットワークから 2 台のクライアント PC でアクセスする形態をとった。

平成 11 年度後期の実験においては、開発されたシステムに対する実験者からの評価を得るように、実験終了後にアンケートを配り、翌週までに回答させ、実験レポートとともに提出させた。また、今回のアンケートでは、システムに対する評価だけではなく、バグの報告や、希望する機能等の意見も求めた。

4.1 アンケート結果

アンケートは、選択形式 3 問と筆記形式 3 問で行った。以下にアンケート結果を述べるが、まず、表 4 に示した筆記形式の結果をまとめたものを参照し、これを考慮に入れながら選択形式の問いを考察する。

●「問 1: この実験を行ったことによって画像処理に興味を持ちましたか?」

図 6 から、結果として、画像処理に非常に興味を持つ

表 4 アンケートから得られた意見をまとめた表
Table 4 List of results of questionnaires.

システムの良い点	システムの悪い点
<ul style="list-style-type: none"> ● 操作の仕方が簡単で使いやすい。(14) ● 実際に結果が目に見える。(10) ● パソコンを使うことができ、楽しかった。(6) ● 処理の種類が豊富(5) ● 面白かった。(5) ● 画像処理に対する興味が深まった。(5) ● 処理が速い。(3) ● 実験が自分の手を動かして行われる。(3) ● 画像についての知識を得ることができる。 ● HTML のヘルプが便利。 ● 初心者でも使いやすい。 ● またやりたいと思った。 ● このような実験をもっとふやしてほしい。 	<ul style="list-style-type: none"> ● パソコンが 1 班に 1 台というのがやりにくかった。(11) ● もっと違う種類の画像を用意してほしい(女の子の画像等)。(4) ● 違いがもっとはっきり分かればよかった。(4) ● 処理が遅いのもあった。(3) ● ウィジェットが英語で分かりにくい。(3) ● 目が疲れる。(2) ● ダイアログの操作が面倒(2) ● 眠くなった。(2) ● 結果を出さないとよく分からない処理もあった。(2) ● プリント出力も Java でやった方がよかった。 ● もっといろんな応用があればよい。 ● 用語がよく分からない。

項目の末尾の括弧書きの数字はその項目の件数を表す。

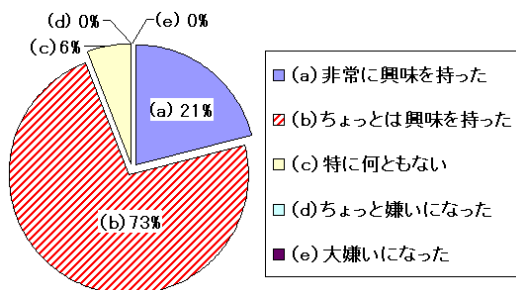


図 6 画像処理に対する興味の喚起(問1)

Fig. 6 Results of Question 1.

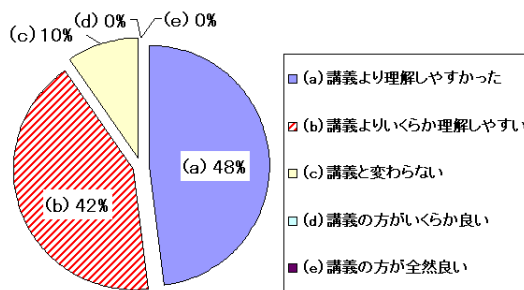


図 7 既存の講義との比較(問2)

Fig. 7 Results of Question 2.

たと評価した学生が 2 割おり、この回答をした学生は、今後は画像処理に触れる際に興味をもって取り組めるのではと思われる。表 4 の良い点の中にもあるように、『画像処理に対する興味が深まった。(5)』という意見からも、実験を行った数割の学生には画像処理に対する興味を喚起できたと考える。

本システムは、画像処理を第一線で行う研究者向けの CAI システムというよりは、これから画像処理に取り組もうといった人たちにに向けた『チュートリアル』的な教育支援システムである。したがって、今回のような、興味が持てたという意見は、期待どおりの結果といえる。

●「問 2: 信号(画像)処理を講義で受けるのと比べて、今回の実験では、画像処理の振舞いについてどちらが理解しやすいと思いませんか?」
このアンケート項目は、今まで学んできた環境に比べ、本システムのような、コンピュータ上で実験を行い、計算(処理)結果を画面に表示し、技術を体験学習できるようなシステムについて、学習する立場からみてどのように思うのかの評価を行う。

図 7 に示すように、9 割が理解しやすいと答えている。

この問いの回答の場合も、純粋に講義と比べての結果というのではなく、今回の実験に対して『新鮮み』を感じていたり、表 4 から分かるように「パソコンを使うことができ、楽しかった。(6)」といった理由があるからと考えられる。また、「実際に結果が目に見える。(10)」「実験が自分の手を動かして行われる。(3)」等、本システムの特徴をあげて「良かった」という意見も多く、これは本システムの成功例と評価できよう。

●「問 3: 今回実験で用いたシステムは、画像処理を学習するシステムとしては使いやすいか?」
この問いは、本システムの使いやすさについての質問である。図 8 に示すように、7 割弱の学生が使いやすいとしている。ところが、「パソコンが班に 1 台というのがやりにくかった。(11)」等というような悪い点も指摘されている。しかし、システムの悪い点にあげられた項目のうちのほとんどがシステムの根本に対する問題ではなく、計算機環境の改善ならびにシステムの向上を望む声が多い。さらに、実験環境や実験の内容に関する意見が多いが、いくつかの問題点をかかえつつも、本システムの有益性を示すことができたと考ええる。

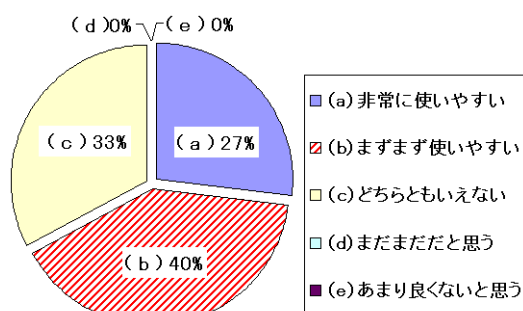


図 8 教育支援システムとしての評価 (問 3)

Fig. 8 Results of Question 3.

5. む す び

アンケート結果から、本システムが学生実験への援用システムとして十分に実用に供するレベルであることが判明した。

今後は、HTML のヘルプを強化し、そのヘルプを実験の実験の補助にして作業を行わせ、独学の支援に足るものであるかの意見、講義とあわせた実習を行った場合の意見等も調査する必要がある。

また、今回のように 2 つの端末で行うというものではなく、高速スイッチングハブを用いて、1 クラス程度の人数で同時に実験を行ったり、課題のレポートとして Java RMI を用いて実際にプログラミングの体験をしたりすることも実施する予定である。

最後に、本システムは、WWW を介して公開する用意があることを付記する。

参 考 文 献

- 1) 佐藤, 岡崎, 河合, 山本, 田村: 画像処理ワークステーション VIEW-Station のソフトウェアアーキテクチャ, 情報処理学会論文誌, Vol.31, No.7, pp.1015-1026 (1990).
- 2) Efford, N.: *Digital Image Processing, A Practical Introduction using Java*, Addison-Wesley (An imprint of Pearson Education) (2000).
- 3) Koryllos, K. and Fisher, R.B.: Interactive Textbooks; Embedding Image Processing Operator Demonstrations in Text.
<http://marathon.csee.usf.edu/hiprjava/hiprjava.html> (July 1997).
- 4) 宮崎, 古川, 相知: Java を用いたネットワーク指向画像処理教育支援ソフトウェアの開発, 日本シミュレーション学会, 第 17 回シミュレーションテクノロジーカンファレンス, pp.139-142 (June 1998).
- 5) Farley, J.: *Java Distributed Computing*, O'Reilly & Associates (1997). 小俣裕一 (監

訳), 豊福剛 (訳) Java 分散コンピューティング, O'Reilly Japan (1998).

(平成 12 年 4 月 28 日受付)

(平成 12 年 10 月 6 日採録)



古川 達也 (正会員)

1956 年生, 1984 年 3 月九州大学大学院工学研究科博士後期課程修了. 同年 4 月長崎大学工学部助手. 1986 年佐賀大学理工学部講師. 1987 年同助教授. 現在に至る. 数値解析, 計算機応用に関する研究に従事. 工学博士. 電気学会会員.



松尾 匡章

1969 年生, 1993 年 3 月佐賀大学大学院理工学研究科修士課程修了. 1993 年 4 月 (有) フライトプラン入社. 1997 年 12 月 (株) ディーディーエス取締役. MPU 組み込みシステムの設計・開発に従事. 2000 年 4 月佐賀大学工学系研究科博士後期課程入学. 電子情報通信学会会員.



宮崎 純生

1975 年生, 2000 年 3 月佐賀大学大学院工学系研究科博士前期課程修了. 同年 4 月日本電信電話 (株) 入社.



堂園 浩 (正会員)

1961 年生, 1989 年 3 月京都大学大学院工学研究科博士後期課程修了. 1989 年 4 月徳島大学工学部助手. 1991 年 4 月熊本大学工学部助手. 1992 年 7 月同講師. 1994 年 4 月佐賀大学理工学部助教授. 現在に至る. フローサイトメトリーへの計算機の応用に関する研究に従事. 工学博士. 計測自動制御学会会員.



相知 政司

1964 年生, 1989 年 3 月長崎大学大学院工学研究科修士課程修了. 同年 4 月 NTT に入社. 1991 年 4 月佐賀大学理工学部助手. 2000 年 4 月同講師. 現在に至る. 博士 (工学). 数値電界解析に関する研究に従事. 電気学会会員.