

# 差異画素抽出に基づく画面照合機能の特徴とする GUI 試験システム

鶴崎真理子<sup>1</sup>      阿倍博信<sup>1</sup>

**概要：**本論文では，GUI アプリケーションの回帰テストにおいて，改修前の試験操作を記録しておき，改修後に操作の自動再生を行う GUI 操作記録再生機能と，改修後の画面を改修前のものと比較し，表示が正常であるかどうか自動判定することにより試験結果の確認作業を自動化する画面照合機能で構成される GUI 試験システムについて述べる．さらに，GUI 試験システムの画面照合機能において，画面差異の傾向から特徴量を抽出し，教師あり学習により分類モデルを生成することで表示異常のパターン毎に分類することによって，試験結果の確認作業を効率化する方式を検討している．Windows アプリケーション画面について本方式の評価を行い，画面差異により作成した表示異常パターンの分類モデルの有効性を確認した．

## A Test System of GUI Applications Using the Screen Matching Function Based on Extraction of Different Pixels

MARIKO TSURUSAKI<sup>1</sup>      HIRONOBU ABE<sup>†1</sup>

### 1. はじめに

ソフトウェアの回帰試験や環境のリプレース時の GUI 試験において，人手の試験操作や画面表示の目視確認による労力が大きいという課題がある．例えば，ソフトウェアの改修時やプラットフォームの更新時に，変更がない箇所についても不具合やバグが発生しないか確認することで品質を担保する必要があるため，複数回にわたる試験操作や膨大な数の画面の目視確認が実施されている．また，試験対象の端末やプラットフォームも PC，タブレット，組み込み機器，Windows，Android，Linux といった OS やネイティブアプリケーション，ブラウザ等多岐に渡っており，これらの複数の環境毎に試験を行うのに労力がかかっている．

これらのソフトウェア試験時の労力を削減することを目的とした，操作自動再生ツールや画像比較ツール等は多数存在する．しかし，それらの多くは試験環境のプラットフォームやアプリケーションが制限される，画面スクリーンショットの一致／不一致判定しかできないため，不一致の画面に対して目視で確認するのに労力がかかる，等の課題があり，適用が難しいケースも存在する．

本研究では，リモートデスクトップの一種である VNC(Virtual Network Computing)ベースの GUI 操作記録再生技術及び，不一致画面から表示異常の原因を推定する画面照合技術による GUI 試験支援方式を検討している．具体的には，初回の試験操作及び画面のスクリーンショットを

記録しておき，2 回目以降の試験で自動再生を行う．初回試験時の画面のスクリーンショットをマスタとして 2 回目以降の画面のスクリーンショットと比較し，画面が正常に表示されているか自動で判定する．VNC はマルチプラットフォームに対応している RFB (Remote Frame Buffer) プロトコル[1]を使用しているため，本システムもマルチプラットフォームに対応している．また，監視制御システム等で適用されているデスクトップ仮想化技術である VDI (Virtual Desktop Infrastructure) [2]では，VNC により画面の伝送を行っているシステムもあり[3][4]，これらのシステムとの親和性も高い．

画面照合フェーズでの実際の運用では，画面が正常に表示されていると判定された画面については除外し，残りの画面について目視で確認すればよい．しかし，前述のように，それらの画面についてどこに異常が出ているか，なぜ表示に不具合がでているか等，目視で確認・調査し，試験報告書等のドキュメントにまとめる作業が必要であり，開発現場の労力がかかっている．本研究では，これらの作業にかかる労力を軽減することを目的として，画面内の差異の傾向から差異のパターンに分類し，自動的に表示不具合の原因を判定する方式を検討している．この方式の評価として，画面差異の傾向から差異のパターン毎に分類できるか調査した．その結果，評価データについて約 80%の精度で差異のパターン毎に分類できることを確認した．

本稿では，2 章にて関連研究，3 章にて GUI 試験の省力化方式，4 章にて評価，考察について述べ，6 章にてまとめを行う．

<sup>1</sup> 三菱電機株式会社 情報技術総合研究所

## 2. 関連研究

ソフトウェア開発現場では、ソフトウェアの小規模な改修や環境のアップデートの度に試験操作や画面の表示確認を実施する必要があり、労力がかかっているという問題がある。この問題への対応策として、試験操作の記録及び自動再生を行う GUI 操作記録再生技術、表示結果の自動確認を行う画像比較技術が知られている。

### 2.1 GUI 操作記録再生技術

表 1 で示しているように、ソフトウェアの回帰試験や環境のリブレース時のソフトウェア試験を効率化するツールは世の中に多数存在する[5]。回帰試験とは、ソフトウェア改修時に変更がない箇所についても試験を行うことにより品質を確保する試験である。ソフトウェア改修時以外でも、OS 等のプラットフォーム更新時に行う試験であり、頻繁に実施されるようなシステムについて効率化を期待できる。

回帰試験効率化を目的とした操作の記録・自動再生機能を持つツールとして、HP Unified Functional Testing[6]、UWSC[7]、Silk Test[8]、Selenium[9]等が知られている。HP Unified Functional Testing、UWSC、Silk Test は Windows や Mac 等のテスト環境上で動作し、マウス操作やキー操作のイベントを記録し、画面のスクリーンショット取得機能、簡易な画面比較機能がある。Selenium はブラウザ試験用のツールであり、Web コンテンツのボタンやテキスト等のオブジェクトを認識して、それらに対する操作の記録機能や、スクリーンショット取得機能、HTML 照合機能がある。また、モバイル向けのツールとして、Silk Mobile[10]、モバイル、Windows 等に対応している Ranorex[11]等もある。

これらのツールの課題として、動作環境が特定の OS、アプリケーションに限定される、操作を記録している操作スクリプトの編集等のメンテナンスに労力がかかる等がある。例えば、Selenium を使用する場合、ソフトの改修時や環境のアップデート時に、操作スクリプトの対応する箇所を修正しなければならない。

表 1 操作記録再生ツール例

Table 1 Examples of Capture and Replay Tools.

ツール	テスト環境	概要
HP Unified Functional Testing	Internet Explorer, Firefox, Chrome, Safari, Windows アプリケーション他	ブラウザ、ネイティブアプリにおいて操作の記録再生を行う。画像、PDF、Word について照合機能がある
UWSC	Windows XP / Vista / 7 / 8 / 8.1	Windows 上のブラウザ、ネイティブアプリにおいてマウス・キーボード操作を記録して再生を行う。

Silk Test	Windows XP / Vista / 7 / 8 / 8.1/10 他	ブラウザ、ネイティブアプリにおいて操作の記録再生を行う。
Selenium	Firefox, Safari, Opera, Linux, Windows, OS X, Solaris (ブラウザのみ)	クロスブラウザ、クロスプラットフォームの UI テストツールとして有名である。ブラウザに表示されるオブジェクトに対する操作を記録する。
Silk Mobile	Android, iOS, BlackBerry, Windows Mobile, Windows Phone, Symbian	USB ケーブルでスマートフォンやタブレット端末等のデバイスと接続し、操作を記録・再生する
Ranorex	Android, iOS, Windows8 他	ブラウザ、ネイティブアプリにおいて操作の記録再生を行う。PC やタブレット等の多数の環境に対応している

### 2.2 画像比較技術

画像比較技術の関連技術として、類似画像検索がある。類似画像検索は、Google Similar Images[12]や Yahoo! ラボの Visual Seeker[13]などのアプリケーションに使用されている。これらは、画像を指定すると、類似した画像をインターネット上から探し、表示するサービスである。類似画像検索では、一般的に以下の手法が用いられている。

#### (1) 相関ベースの手法

図 1 で表しているように、正規化二乗誤差等、照合する二枚の画像各画素の輝度値の相関値を評価に用いる。代表的な方式として、SSD (Sum of Squared Difference) や SAD (Sum of Absolute Difference) が知られている。

#### (2) 特徴ベースの手法

図 2 で表しているように、SIFT(Scale-Invariant Feature Transform)[14]、SURF(Speeded Up Robust Features)[15]、ORB(Oriented-BRIEF)[16]等の局所特徴量を二枚の画像からそれぞれ抽出し、その対応関係を評価に用いる。例えば、操作記録・再生画面の対応点を取り、特徴点の個数に占める割合を類似度とする。一般に、自然背景等の複雑な画像の照合を行う際に有効な手法として知られている。

文献[17]では、これらの方式を基に類似度を定義し、閾値による判定を行う方式を提案している。この方式により操作記録・再生時の画面について類似度を算出し、画面が正常に表示されているか判定するためには、閾値を決める必要がある。例えば、類似度を 0～1 の範囲になるように定義し、閾値を 0.99 とした場合、タイトルバーや画面内の枠のわずかなずれによるわずかな画素の違いの出るような

ケースの表示が正常であると判定できるかもしれない。しかし、画面内の小さなテキストフォーム内の文字が一文字違うなど、類似度の値の差は小さいが表示が異常であるケースについて誤判定してしまう可能性がある。また、局所特徴量ベースで GUI 画面の比較をする場合、自然画像と違い、あまり特徴量が抽出されない GUI 画面で適切な判定ができないという問題がある。

GUI 試験の現場で使用する場合、画面照合で正常と判定されたものについては人手で画面の確認を行わず、異常と判定されたものについてのみ人手で画面の確認を行うという使い方をする。文献[17]のような方式では、表示が異常であるものを正常であると誤判定してしまう可能性があるため、現場での使用には適していない。

実際に試験の現場で使用されている、操作の記録・自動再生機能、ビットマップチェックによる画面比較機能をそなえているツールとして、前述の HP Unified Functional Testing が有名である。画面比較機能では、操作記録・再生時に取得した画面スクリーンショットについて、範囲を指定しビットマップチェックにより画面が正常に表示されているか判定する。この機能は、ビット単位で比較しているので、画像のうち 1 ビットでも異なると不一致と判定されてしまう。例えば、目視でも判別が困難な、タイトルバーや画面内の枠のわずかなずれによる画素の違いで不一致と判定されてしまうことがある。

画面比較機能を使用する場合、比較の結果完全一致の画面を除外し、不一致と判定されたものについて目視で確認するという運用をすることが想定される。このとき、不一致と判定される画面が多いと、目視で確認しなければならない画面数が多くなり、あまり労力の削減を期待できない。

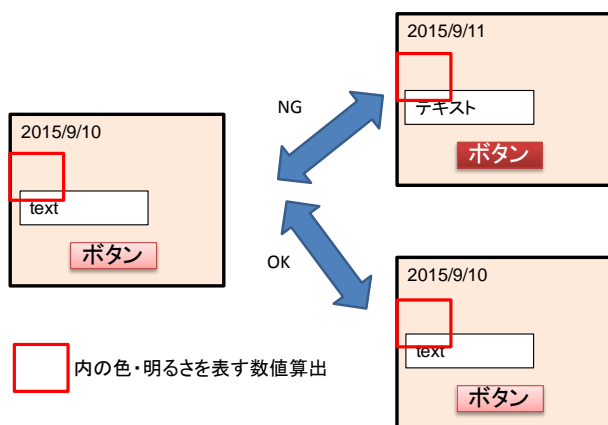


図 1 相関ベースの手法  
Fig 1 Matching Based Correlation.

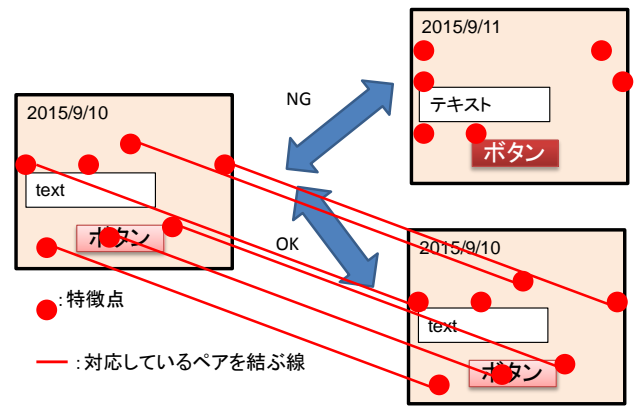


図 2 特徴ベースの手法  
Fig 2 Matching Based Image Features.

### 3. GUI 試験支援方式

前章で述べた GUI 試験における課題の解決策として、VNC ベースの GUI 操作記録再生技術[18]と差異画素抽出による GUI 画面照合技術で構成される、GUI 試験システムを提案する。GUI 試験システムのシステム構成例及び処理フローを図 3、図 4 に表している。

GUI 操作記録再生技術では、初回の操作及び画面スクリーンショットを記録し、2 回目以降の操作時に、操作の自動再生及び画面スクリーンショットの記録を行う。

画面照合処理では、操作記録時に取得した画面スクリーンショットをマスタとして、操作再生時の画面スクリーンショットと比較する。このとき、対応する画素の RGB 値や輝度値等と比較し、完全一致であった画面については照合対象画面から除外し、それ以外について目視で確認するという運用を想定している。不一致であった画面について目視で確認する際に、画面のどこに差異が出ているかを強調表示し、差異画素の分布の傾向等から画面の表示異常の原因毎に自動で分類することにより、画面の表示確認作業を効率化する。

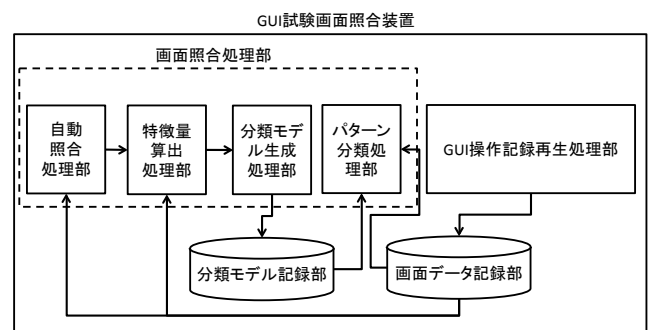


図 3 GUI 試験省力化システム構成例  
Fig 3 Example of System Configuration of Test System of GUI Applications.



図 4 GUI 試験省力化方式処理フロー  
Fig 4 Flow of Test of GUI Applications.

### 3.1 GUI 操作記録再生方式

本方式は、リモートデスクトップの一種である VNC に GUI 操作の記録機能及び再生機能を追加することにより、GUI 試験を支援するものである。具体的には、図 5 で表しているシステム構成図の操作端末 (VNC クライアント) から試験端末 (VNC サーバ) の遠隔操作を行う際に送信される操作コマンドを操作端末側で記録しておく。再生時には、操作端末で記録した操作コマンドを読み込み、試験端末に操作コマンドを再送信することによって自動再生を行う。VNC は RFB プロトコルにより操作コマンドの送受信をする。このプロトコルは Windows, Linux, Android 等のマルチプラットフォームに対応しているため、本方式もこれらのプラットフォームで適用することができる。

RFB プロトコルによる操作コマンドは、表 2~4 で表しているように、マウス座標・クリック情報、キー種別・押下情報、試験端末に対する画面更新情報の要求であり、これらの操作コマンドを時刻情報と共に操作シナリオとして記録する。図 6 はその一例である。操作再生時には、この時刻情報に対応する間隔で操作コマンドを送出する。この送出間隔を調整することにより、倍速再生を行うことも可能である。操作記録・再生時に、一定間隔又は特定の操作コマンドが送出されるタイミング等で試験端末の画面スクリーンショットをキャプチャしておき、操作記録時の画面スクリーンショットをマスタとして操作再生時の画面スクリーンショットと照合することによって、画面表示確認を効率化することもできる。

また、リモートデスクトップの構成を利用しているため、図 7 のように、プラットフォームが異なる複数台の試験端末や仮想環境について同時に再生することによって、複数台端末の同時試験を行うことも考えられる。また、複数台端末が連携して動作するシステムやコンテンツについても、操作端末で起動するそれぞれの試験端末について VNC クライアントの同期機能を追加する等の方式により操作再生を行うことが可能である。

表 2 マウス座標・クリック情報

Table 2 Commands of Mouse Moving and Clicking.

バイト数	値	内容
1	5	コマンド型
1		クリック情報
2		X 座標
3		Y 座標

表 3 キー種別・押下情報

Table 3 Commands of Key Code.

バイト数	値	内容
1	4	コマンド型
1		押下フラグ
2		パディング
4		キー種別

表 4 試験端末に対する画面更新情報要求コマンド

Table 4 Commands of Frame Buffer Update Request.

バイト数	値	内容
1	3	コマンド型
1		増加分
2		X 座標
2		Y 座標
2		幅
2		高さ

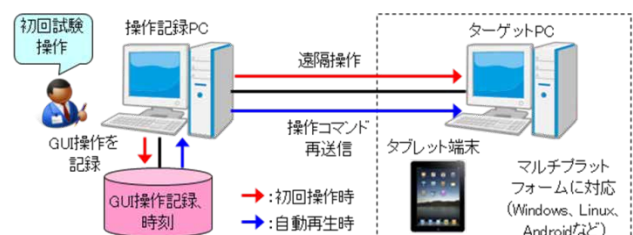


図 5 GUI 操作記録再生技術システム構成

Fig 5 System Configurations of Operations Capture and Replay.

```

0,k,0,0,65506,,
0,p,0,686,274,,
62,p,1,686,274,,
16,p,0,686,274,,
140,p,0,685,275,,
0,p,0,685,277,,
16,p,0,682,279,,
0,p,0,679,282,,
16,p,0,674,287,,
0,p,0,666,292,,
15,p,0,657,299,,
0,f,1,0,0,1366,768
63,p,0,550,376,,
0,p,0,531,390,,
0,p,0,508,404,,
15,p,0,488,420,,
0,p,0,469,433,,
16,p,0,451,446,,
0,p,0,440,454,,
31,p,0,419,474,,
16,p,0,403,490,,
...

```

図 6 シナリオファイル例

Fig 6 Example of Scenario File.

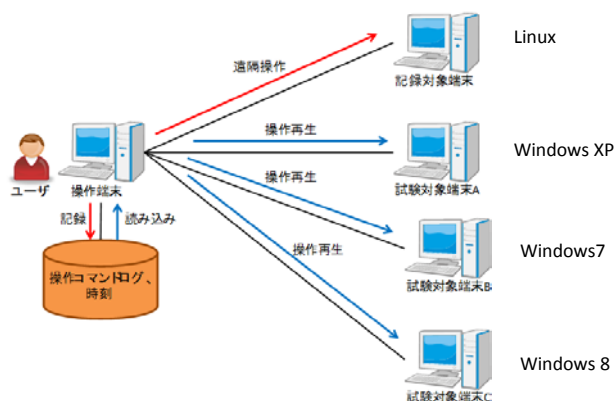


図 7 複数台同時再生時のシステム構成

Fig 7 System Configurations of Multiple Device Replaying.

### 3.2 GUI 画面照合方式

GUI 試験における画面表示確認作業を効率化することを目的として、操作記録・再生時の画面スクリーンショット画像を照合し試験結果を自動判定する。具体的には、操作記録時の画面スクリーンショットをマスタとして、操作再生時の画面スクリーンショットとの画素単位の比較を行う。画素比較の結果、画面スクリーンショット内の全ての画素が完全に一致している画面は目視で確認する必要がないため確認対象から除外し、それ以外の画面の組について目視で表示を確認する。

目視で表示を確認する際、画面内のどこに差異が出ているのか、どのような原因で不具合が出ているのか、等を目

視で確認し、試験報告書としてまとめる作業がある。ソフトウェア開発の現場では、2.2 で述べたツールや機能を使用して画面を比較し、表示作業を効率化する取り組みを行っているが、このような完全に一致していない画面について目視で確認する作業に工数がかかっている。

この課題を解決するために、図 8 で表しているような、完全一致でない画面の組について、画面内の差異の傾向から差異のパターンに分類し、ユーザに提示する機能を検討する。

このとき、差異の傾向やパターンは、コンテンツ、あるいはシステム毎に異なるので、コンテンツやシステム毎に差異の特徴量やパターンを定義する必要がある。例えば図 9 の場合、画面の構成要素として日付、テキストボックス、ボタンが存在するが、それらの構成要素上に出現する差異画素の輝度値の差分や分布領域等を差分特徴量として抽出してモデルを生成する。日付やテキストの場合、文字が異なる差異の場合が多いので、小さい領域に差異が集中している、ボタンの差異の場合、フォーカスによる色の変化による差異が多いので、輝度差が小さい等の特徴があることが考えられる。

具体的には以下の手順でコンテンツ毎に分類モデルを生成し、モデルにより差異のパターンに分類する。

#### (1) 操作記録／再生画面比較

操作記録時に取得した画面スクリーンショットをマスタとして、操作再生時に取得した画面スクリーンショットと比較する。まず、画面スクリーンショット内の画素が全て完全に一致しているか、画素値のマッチングにより判定する。画面内の全ての画素値が完全に一致している画面の組については目視確認をする必要がないため照合対象から除外し、それ以外の画面について差異パターン分類を行う。

#### (2) NG 画面の差異特徴量算出

NG と判定された画面の組について、差異の特徴から差異パターンの分類を行う。差異の特徴の例として差異領域の座標や大きさ、輝度差の最大／平均／分散、差異領域内の差異画素の個数等がある。

#### (3) NG パターン分類モデル作成

照合するデータを学習データとテストデータに分割する。学習データについて、目視により判定した差異のパターンをラベル、特徴量をデータとして SVM (Support Vector Machine) [19]等により、NG パターン分類モデルを生成する。

#### (4) NG パターン分類

NG パターン分類モデルを基に、テストデータについて NG パターン分類を行う。具体的には、不一致と判定された画面の組の差異画素を分類モデルに入力し、NG パターンを出力する。



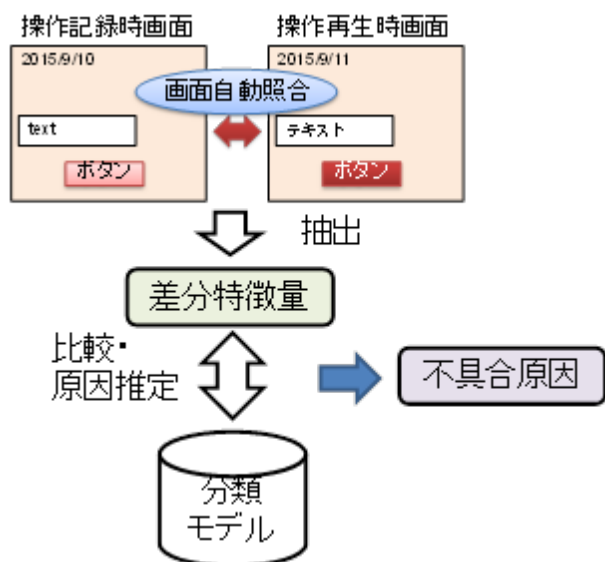


図 8 GUI 画面照合の流れ  
Fig 8 Flow of the Screen Matching.

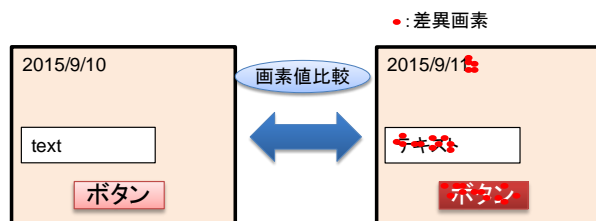


図 9 差異画素抽出の例  
Fig 9 Example of Extraction of Different Pixels.

## 4. 差異特徴分類方式

3 章で提案した GUI 画面照合方式について、評価を下記の手順で実施した。

### 4.1 差異特徴分類分類処理

画面自動照合で不一致と判定された画面の組の差異の特徴量から、教師あり学習により、差異の内容ごとに定義した差異クラスに正しく分類できるか評価した。

下記の手順にしたがって、差異クラスに分類できるか評価した。

#### (1) 差異領域抽出

不一致の画面の組から、手動で差異領域を抽出する。

図 10 はその一例である。図 10 の場合、画面の構成要素である日付表示部分、テキストフォーム、ボタンに差異が生じているので、これらの構成要素の表示領域を差異領域として抽出する。このとき、画面の位置ずれや画面全体が異なる差異等の、全域に差異画素がまだらに分布するような画面において、個々の領域を抽出するのではなく、画面の構成要素全体を包含するよ

うに領域を抽出する。

#### (2) 各差異領域の特徴量算出

各差異領域について、表 7 で表しているような差異領域のサイズや、差異領域内の輝度差分の最大／最小／平均値等の特徴量を算出する。

#### (3) 各差異領域についてクラス分類

各差異領域について、差異のクラス毎に分類する。表 5 はクラス定義の一例を表している。クラス定義については表示異常例、画面内の要素などコンテンツによって適切なものが異なるので、コンテンツ毎に設定することが望ましい。

#### (4) SVM によるクラス分類

差異領域の特徴量及びクラスにより SVM でモデルを生成し、クラスの予測を行う。今回はデータ数が少ないため、GUI アプリケーションについて、クロスバリデーションによる差異パターン分類モデルの評価を行う。今回は、SVM を使用してクラス分類を実施したが、決定木やニューラルネットワーク等の他の教師あり学習によるアルゴリズムを使用してもよい。

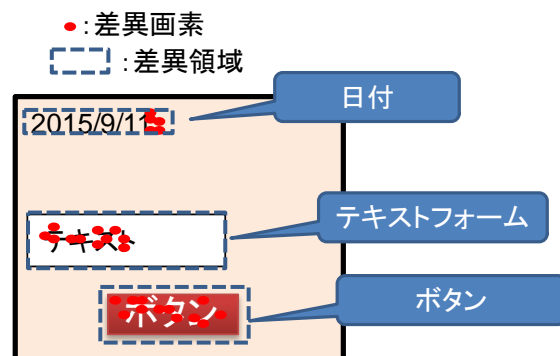


図 10 差異領域抽出の例  
Fig 10 Example of Extraction of Different Regions.

表 5 差異クラス

Table 5 Class of Different Types.

差異原因	内容
可変状態表示	日付・時刻、ファイル名やプロジェクト名等、表示が変わってもよいもの
タイトルバー	タイトルバーの色の差異などの文字以外の差異
位置ずれ	画面内の線や要素がずれている
グラフ・表	グラフ、表の中身が異なる
スクロールバー	スクロールバーの大きさ、位置が異なる
ボタン	ボタンフォーカス、ボタン有無
テキストフォーム	テキストフォームの中身の差異

固定状態表示	表示がかわってはいけないもの
チェックボックス	チェックボックスのチェック有無，フォーカス
画面全体差異	画面自体が異なる，画面サイズが異なる

#### 4.2 評価実験

表 6 に表しているように，Java により開発された Windows アプリケーションの画面スクリーンショット画像をグレースケール変換し，0～255 の画素値を比較したときに 1 画素以上異なった画面の組について差異クラス毎に分類できるか評価を行った。

表 6 評価条件

Table 6 Condition of the Evaluation Experiment.

項目	判定精度
評価画像	Java により開発された Windows ネイティブアプリケーションの画面スクリーンショット.
画像枚数 (枚)	73
差異領域数	192
自動照合方式	画素値(グレースケール, 0～255)比較

#### 4.3 結果

差異クラスについて，クロスバリデーションによる評価を実施した．クロスバリデーションの分割数は 5，カーネルはデフォルトの RBF カーネル (ガウス放射基底関数カーネル) を使用した．各差異特徴量による判定精度，判定精度が高い差異特徴量を組み合わせた場合の判定精度を表 7 に表している．判定精度＝正しく判定できた差異数／全体の差異数×100 (%) である．

表 7 を見ると，差異領域サイズ，差異領域座標，輝度差分平均値，輝度差分最大値を組み合わせた場合に最も判定精度が高くなり，約 80% の精度で正しく判定できることがわかる．

表 7 評価結果：各差異特徴量の判定精度

Table 7 Evaluation Results: Accuracy of Classification

使用特徴量	判定精度 (%)
差異領域サイズ	65
差異領域座標	65
類似度	4
輝度差分平均	50
輝度差分分散	4
輝度差分最大	53
差異領域サイズ+差異領域座標	73

差異領域サイズ+輝度差分平均	70
差異領域サイズ+輝度差分最大	70
差異領域サイズ+差異領域座標+輝度差分平均+輝度差分最大	81

#### 4.4 考察

表 7 からわかるように，差異領域サイズ，差異領域座標，輝度差分平均値，輝度差分最大値を組み合わせて使用した場合の判定精度が最も高かった．このときの誤判定のパターンを表 8 に表している．

表 8 差異クラスの評価における誤判定件数

Table 8 Number of Classification Error

誤判定ケース		件数
真のクラス	推定クラス	
ボタン	可変状態表示	3
可変状態表示	テキストフォーム	6
チェックボックス	可変状態表示	3
タイトルバー	位置ずれ	2
テキストフォーム	可変状態表示	6
位置ずれ	画面全体差異	5
グラフ・表	可変状態表示	1
ボタン	グラフ・表	1
スクロールバー	可変状態表示	2
可変状態表示	ボタン	1
画面全体差異	位置ずれ	5
チェックボックス	ボタン	1
グラフ・表	ボタン	1

表 8 をみると，可変状態表示⇔テキストフォームの誤判定が最も多いことがわかる．この原因として考えられるのは，両クラスの差異領域サイズと輝度値の差分の分布の傾向が類似していることである．このパターンの具体的な誤判定例として，テキストフォーム内の背景色が白色以外である場合があった．

また，画面全体差異⇔位置ずれについても誤判定が多かった．これらは，差異が画面全体に分布するため，差異領域サイズや差異領域座標が類似した値になることが原因であることが考えられる．

判定精度を向上させるために下記の①～③の方法が考えられる．

##### ① 差異のクラス分け方法

実際の不具合例に基づいたクラス分け，試験報告書へ

の記入形式に合ったクラス分けを検討することが必要である。

## ② 使用する特徴量

テキストフォームと可変状態表示、位置ずれと画面全体差異を分離できるような特徴量を検討すべきである。例えば、パターンマッチングとの組み合わせ、局所特徴量の使用等が考えられる。

## ③ クラス分類の方式

クラス分類時や SVM のパラメータのチューニングやアルゴリズムの設定等を検討するべきである。

## 5. おわりに

本稿では、GUI アプリケーション試験において、VNC ベースの GUI 操作記録再生技術と、GUI 操作記録再生時にキャプチャした画面スクリーンショットの画像照合技術による GUI 試験支援方式を提案した。

VNC ベースの GUI 操作記録再生方式により、限定されたプラットフォーム、アプリケーションだけでなく、マルチプラットフォームでの適用を実現できる。さらに、記録した操作コマンドについて、試験シナリオ編集を行うことによってテキスト入力値を変更する、複数のシナリオを一括再生する、シナリオの一部を削除する等の応用も可能である。

また、画面照合機能において、操作記録・再生時に取得した画面スクリーンショット画像を照合し、画像内の差異画素の傾向から差異の原因パターンに分類する方式の検討、評価を実施した。具体的には、画面自動照合で不一致と判定された画面の組の差異の特徴量から、教師あり学習によりモデルを生成し、差異のクラスに分類できるか評価した。評価結果から、約 80% の精度で正しいクラスに分類できることを確認した。この結果から、各差異のクラス毎に差異に特徴があることが分かった。

今後は、差異クラスの定義や特徴量、使用するパラメータ、アルゴリズムを検討することによって、精度の向上や分類機能の実現を目指す。また、差異領域について手動で検出して評価を実施したが、実際のアプリケーションの機能として適用するためには差異領域の自動検出が必要である。さらに、差異領域の自動検出方式や差異クラス及び特徴量の選定方式等を検討する。

## 参考文献

- [1] Richardson, T. and Levine J. : The Remote Framebuffer Protocol, RFC6143 (2011).
- [2] 日経コンピュータ, 日経 Linux, 日経 NETWORK(編):すべてわかる仮想化大全, 日経 BP (2014).
- [3] 渡部洋司, 三田村謙一, 工藤謹正: 電力系統監視制御システムの技術動向, 東芝レビュー, Vol.63, No.4, pp.2-5(2008).
- [4] 菅井尚人, 塚原研, 鶴薫, 郡光則: 次世代電力系統制御システム向けプラットフォーム技術, 三菱電機技報, Vol.89, No.11,

pp.600{604 (2015).

- [5] ASTER テストツール WG: テストツールまるわかりガイド (入門編) Version 1.0.0, ソフトウェアテスト技術振興協会(2012).
- [6] Unified Functional Testing:  
<http://www8.hp.com/jp/ja/software-solutions/unified-functional-automated-testing/>.
- [7] UWSC:  
<http://www.uwsc.info/>
- [8] Silk Test:  
<http://www.microfocus.co.jp/products/silk/silktest/>.
- [9] Slenium:  
<http://www.seleniumhq.org/>
- [10] Silk Mobile:  
<https://www.microfocus.co.jp/products/silk/silkmobile/>
- [11] Ranorex:  
<https://www.techmatrix.co.jp/quality/ranorex/>
- [12]<http://googleblog.blogspot.jp/2009/10/similar-images-graduates-from-google.html>
- [13]<http://visseeker2.yahoo-labs.jp/vs/>
- [14]D. G. Lowe, "Object recognition from local scale-invariant features", Proc. of IEEE International Conference on Computer Vision (ICCV), pp.1150-1157, 1999.
- [15]Herbert Bay, Tinne Tuytelaars, Luc VanGool "SURF: Speeded Up Robust Features", computer vision-ECCV Lecture Notes in Computer Science, 2006.
- [16]E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: an efficient alternative to SIFT or SURF. In Proc. of IEEE International Conference on Computer Vision, 2011.
- [17] 鶴崎真理子, 阿倍博信, "マルチプラットフォーム対応 GUI 試験システムにおける画面照合方式," 第 14 回情報科学技術フォーラム(FIT2015), 2015
- [18] 鶴崎真理子, 阿倍博信, "マルチプラットフォーム対応 GUI 試験省力化方式," 電子情報通信学会総合大会, 2015
- [19]津田宏治, "サポートベクターマシンとは何か", 電子情報通信学会誌, Vol. 83, No. 6, pp. 460-466, 2000.