

## 講義スライドを二次利用したプログラミング動画生成手法

大久保 孝明 † 井上 亮文 † 星 徹 †

† 東京工科大学コンピュータサイエンス学部

### 1 はじめに

プログラミングのスキルを向上させるためには、プログラムの動作や流れを体感することが重要である[1]。しかし、プログラミングの授業で使用される電子スライドはソースコードや文法の説明が中心になり、実際の動作や流れをイメージしにくい。そのため、電子スライドを利用した復習は、プログラミングのスキル向上には効果が小さく、適切な復習方法ではない。

プログラムの動作を体感する最善の方法は、学生が実際にソースコードを打ち、実行して確認するという方法である。しかし、PCを使用することから学習環境が制限され、どこでも気軽に復習することができない。講師がプログラムの流れを追った動画教材を作成する方法もあるが、撮影機材が必要であることや、編集の手間から講師に大きな負担を強いることになる。

本研究では、学生のプログラミングに対する学習効果の向上と講師の教材作成の負担の軽減を目的とし、電子スライドを二次利用した動画生成手法を提案する。

### 2 電子スライド

#### 2.1 特徴

プログラミングの授業で使用される電子スライドは、文法や構文を説明するページとソースコードや実行画面を説明するページに分けられる。その中でもソースコードや実行画面の説明は、プログラミングならではのソースコードの書式やコメントの方法など、共通の表記方法が使用されている。本研究では、この特徴をプログラミング特有表記と呼ぶ。

#### 2.2 現状と要求

配信された電子スライドを見るには、PC や PDA など特定の端末が必要になる。学生の多くが持っているような小型メディアプレーヤーで視聴するには、電子スライドを画像に変換する必要がある。しかし、元の電子スライドが小型メディアプレーヤーのディスプレイのサイズを考慮したデザインでないため、画像の一枚絵で見ると、字が読みづらい、内容が分かりづらいと

Creating method of a programming animation by secondary user of a lecture slide

† Takahiro OKUBO(tokubo@star.cs.teu.ac.jp)

† Akifumi INOUE(akifumi@cs.teu.ac.jp)

† Tohru HOSHI(hoshi@cs.teu.ac.jp)

School of Computer Science, Tokyo University of Technorogy

(†)

1404-1 Katakura, Hachioji, Tokyo 192-0982, Japan

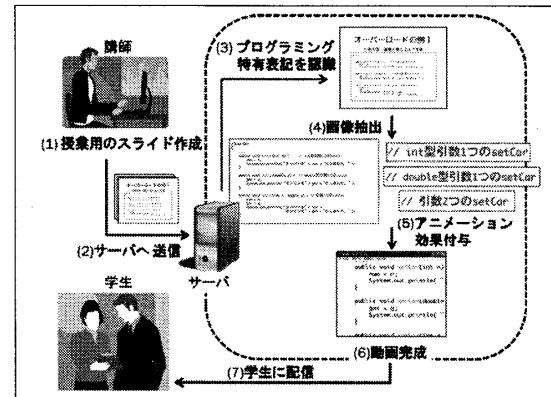


図 1: システム概要

いった問題がある。

以上より本研究では、プログラミングの復習教材に対して、

1. 作成済みの電子スライドをそのまま活用する、
2. 学生が利用しやすい小型メディアプレーヤでの利用を可能とする、
3. 動画内容は学習効果の高いソースコードや実行画面を中心とした内容にする、

の 3 点を考慮して、動画形式の教材を自動的に作成することを目標とする。

#### 3 提案

本研究で提案するシステム概要を図 1 に示す。まず、講師は授業で使用した電子スライドをサーバへ送信する。次にサーバでは受信した電子スライドに画像処理を施し、2.1 節で述べたプログラミング特有表記を認識する。認識したプログラミング特有表記から、動画の材料となる画像を抽出し、切り取る。最後に切り取った画像にズームやスクロールといったアニメーション効果を付与し、動画を生成する。

講師は授業用の電子スライドを作成する作業に、出来上がった電子スライドをサーバに送信するという手間を加えるだけで、別途動画教材を学生に提供することができる。学生は動画形式で配布された教材を小型メディアプレーヤで利用することができる。動画の内容は、授業で使用した電子スライドからソースコードや実行画面に絞ったものである。抽象的で分かりにく

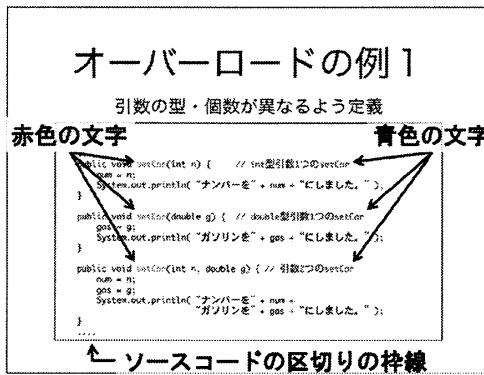


図 2: 電子スライドのページ例

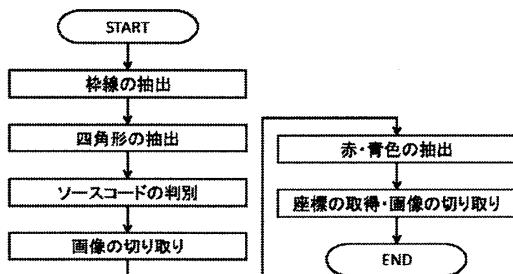


図 3: 画像処理の流れ

い箇所を動的に繰り返し確認できるので、学習効果の高い復習方法が可能となる。

#### 4 実装

認識するプログラミング特有表記は、ソースコードを囲む四角の枠線と赤・青色の文字にした。これらは実際に授業で使用している電子スライドを比較し、表記方法の使用頻度や色の効果などを考慮して決定した。図 2 にシステムへ送信する電子スライド例を示す。図 2 の電子スライドは本大学で実際に使用されたものである。

画像処理の流れを図 3 に示す。サーバでは、まず電子スライドを JPEG 形式の画像に変換する。ソースコードと実行画面を特定するため、変換した画像に対して 2 値化処理を行い、(1) 画像にある枠線をすべて抽出する。(2) 抽出した枠線から頂点が 4 つあるものを四角形と判断し、さらに(3) 抽出した画像と電子スライドの元画像の面積を比較する。条件にあったものをソースコードと判断し、(4) 画像を切り取る。(5) 切り取った画像から赤と青の文字画像を特定する。色ごとに再度 2 値化処理とモルフォロジー変換を行い、文字列をひとつの塊として抽出する。抽出した画像にラベリング処理を行い、(6) 画像の切り取りと座標の取得を行う。

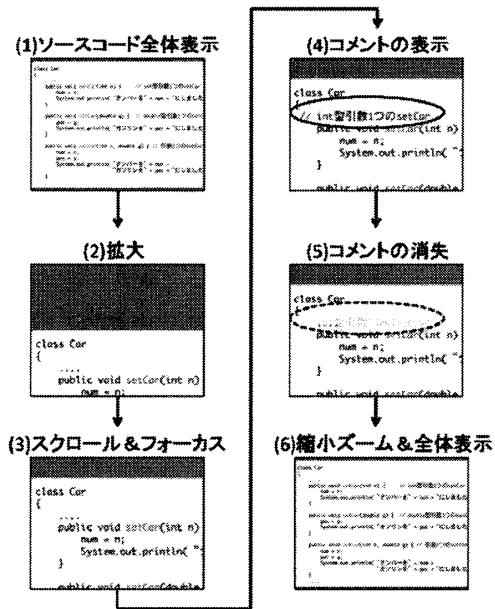


図 4: 動画の構成例

取得した画像や座標から動画のストーリーやカメラワークを自動で決定し、アニメーション効果を付与する。図 2 の電子スライドから生成される動画の構成内容を図 4 に示す。まず、学生がどのプログラムを見ているか把握できるように(1) ソースコードの全体図を表示する。次に、(2) 全体図からソースコードの先頭が画面の中央に表示するように拡大ズームする。(3) ソースコードの赤い文字で書かれた部分が動画画面の中央に表示されるようにスクロールし、一時停止する。(4)(5) 青い文字で書かれたコメントが出現して消える。最後に(6) 再度全体を確認できるよう縮小ズームし、ソースコードをスクロールしてフェードアウトする。

#### 5まとめ

本研究では授業で使用された電子スライドを二次利用し、復習用の動画を自動生成する手法の提案と実装を行った。プログラミング特有表記を利用し、小型メディアプレーヤーのような限られた画面サイズでも、ソースコードや実行画面の流れを動的に追える教材を自動生成することができた。

今後はシステムが認識するプログラミング特有表記の種類を増やすとともに、学習におけるより効果的な動画の表示方法の検討を行っていきたい。

謝辞 本研究の一部は科学研究費補助金若手研究(B)（課題番号 19700648）の助成による。

#### 参考文献

- [1] 史 他:「プログラミング教育のための可視化ツールの開発」、信学技報. ET Vol.107, No.205 pp.11-16