Mixed-reality PDF editor

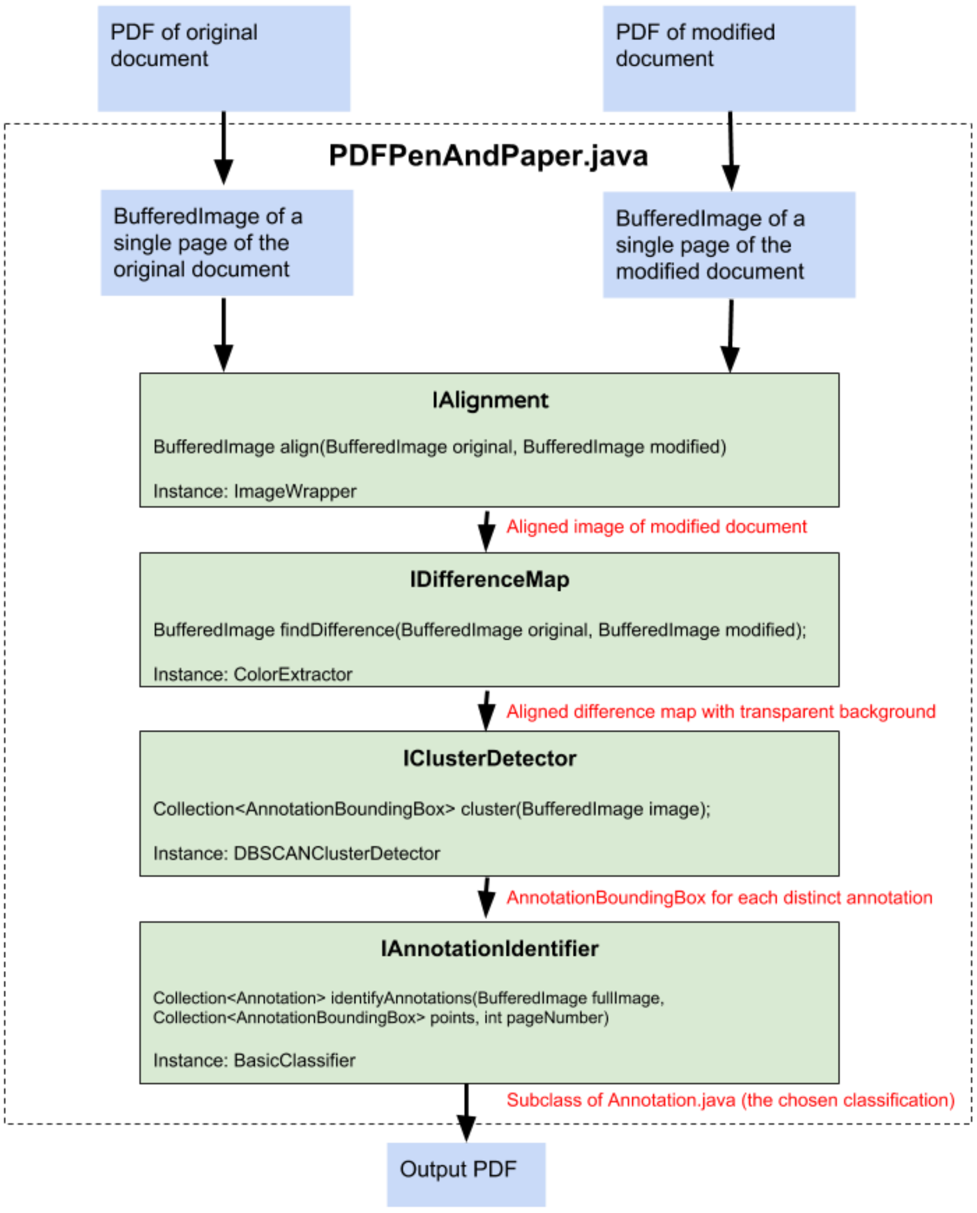
Developer API SDK Documentation

Andrius Grabauskas, Aga Koc, Joe Malt, Gideon Mills & Diptarko Roy

Introduction

This document describes the main public methods and relevant public types at each stage of the pipeline used to process the input images. For each class, minimal code examples for some common tasks are provided and we describe how a third-party developer may extract the raw information at each stage of the pipeline in order to adapt it to their particular application.

Overview of Pipeline Stages


PDF of original document
PDF of modified document
PDFPenAndPaper.java
BufferedImage of a single page of the original document
BufferedImage of a single page of the modified document
IAlignment
BufferedImage align(BufferedImage original, BufferedImage modified)
Instance: ImageWrapper
Aligned image of modified document
IDifferenceMap
BufferedImage findDifference(BufferedImage original, BufferedImage modified);
Instance: ColorExtractor
Aligned difference map with transparent background
IClusterDetector
Collection<AnnotationBoundingBox> cluster(BufferedImage image);
Instance: DBSCANClusterDetector
AnnotationBoundingBox for each distinct annotation
IAnnotationIdentifier
Collection<Annotation> identifyAnnotations(BufferedImage fullImage, Collection<AnnotationBoundingBox> points, int pageNumber)
Instance: BasicClassifier
Subclass of Annotation.java (the chosen classification)
Output PDF

The PDFPenAndPaper class converts each PDF page into a BufferedImages for the other modules, and then reapplies the annotations to produce a PDF as output, while acting as a controller class which integrates all the modules together.

ImageWrapper.java

Public Methods

- **public** BufferedImage align(BufferedImage original, BufferedImage modifiedBeforeARCorrect)

Related Public Types

- TextBoundingBox
- LineSegment
- Coordinate

Code Examples

Aligning two images

Assuming original and modified are two BufferedImage instances (of possibly different dimensions and aspect ratios), images of the original PDF page and the scan respectively, to produce a new BufferedImage instance with the pixels aligned with the original (by aligning the bounding boxes of the text in each image) and with the output being of the same dimensions as original:

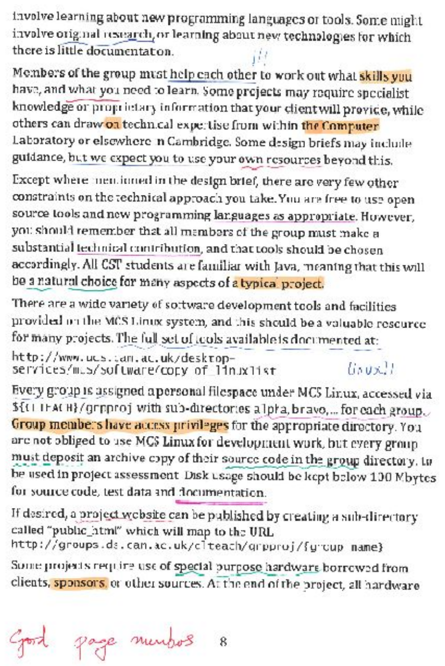
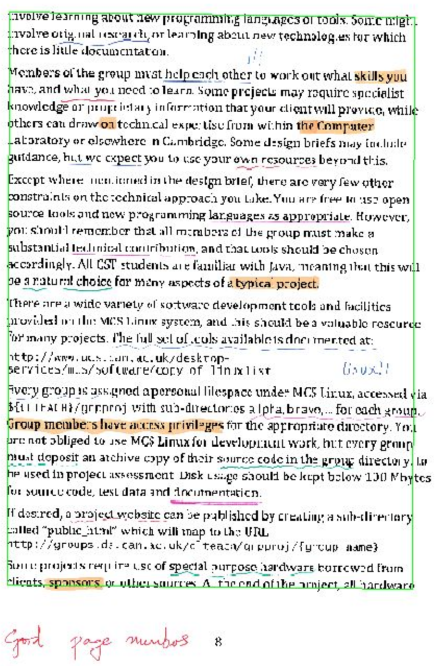
```
ImageWrapper imageWrapper = new ImageWrapper();
BufferedImage aligned = imageWrapper.align(original, modified);
```

Get coordinates of the text bounding box of the image

```
BufferedImage bufferedImage = ...;
TextBoundingBox textBoundingBox = (new ImageWrapper(bufferedImage)).boundingBox();
```

Get a text bounding box overlaid on an input image

```
BufferedImage bufferedImage = ...;
BufferedImage originalPlusTextBoundingBoxOverlay = (new
ImageWrapper(bufferedImage)).getWithBoundingBox();
```

(BufferedImage) bufferedImage	(BufferedImage) originalPlusTextBoundingBoxOverlay
	

involve learning about new programming languages or tools. Some might involve original research, or learning about new technologies for which there is little documentation.

!!

Members of the group must help each other to work out what skills you have, and what you need to learn. Some projects may require specialist knowledge or proprietary information that your client will provide, while others can draw on technical expertise from within the Computer Laboratory or elsewhere in Cambridge. Some design briefs may include guidance, but we expect you to use your own resources beyond this.

Except where mentioned in the design brief, there are very few other constraints on the technical approach you take. You are free to use open source tools and new programming languages as appropriate. However, you should remember that all members of the group must make a substantial technical contribution, and that tools should be chosen accordingly. All CST students are familiar with Java, meaning that this will be a natural choice for many aspects of a typical project.

There are a wide variety of software development tools and facilities provided on the MCS Linux system, and this should be a valuable resource for many projects. The full set of tools available is documented at:

http://www.ucs.cam.ac.uk/desktop-services/mcs/software/copy_of_linuxlist

Linux!!

Every group is assigned a personal filespace under MCS Linux, accessed via ${CLTEACH}/grpproj with sub-directories alpha, bravo, ... for each group. Group members have access privileges for the appropriate directory. You are not obliged to use MCS Linux for development work, but every group must deposit an archive copy of their source code in the group directory, to be used in project assessment. Disk usage should be kept below 100 Mbytes for source code, test data and documentation.

If desired, a project website can be published by creating a sub-directory called "public_html" which will map to the URL http://groups.ds.cam.ac.uk/clteach/grpproj/{group_name}

Some projects require use of special purpose hardware borrowed from clients, sponsors, or other sources. At the end of the project, all hardware

Good page numbers 8

involve learning about new programming languages or tools. Some might involve original research, or learning about new technologies for which there is little documentation.

!!

Members of the group must help each other to work out what skills you have, and what you need to learn. Some projects may require specialist knowledge or proprietary information that your client will provide, while others can draw on technical expertise from within the Computer Laboratory or elsewhere in Cambridge. Some design briefs may include guidance, but we expect you to use your own resources beyond this.

Except where mentioned in the design brief, there are very few other constraints on the technical approach you take. You are free to use open source tools and new programming languages as appropriate. However, you should remember that all members of the group must make a substantial technical contribution, and that tools should be chosen accordingly. All CST students are familiar with Java, meaning that this will be a natural choice for many aspects of a typical project.

There are a wide variety of software development tools and facilities provided on the MCS Linux system, and this should be a valuable resource for many projects. The full set of tools available is documented at:

http://www.ucs.cam.ac.uk/desktop-services/mcs/software/copy_of_linuxlist

Linux!!

Every group is assigned a personal filespace under MCS Linux, accessed via ${CLTEACH}/grpproj with sub-directories alpha, bravo, ... for each group. Group members have access privileges for the appropriate directory. You are not obliged to use MCS Linux for development work, but every group must deposit an archive copy of their source code in the group directory, to be used in project assessment. Disk usage should be kept below 100 Mbytes for source code, test data and documentation.

If desired, a project website can be published by creating a sub-directory called "public_html" which will map to the URL http://groups.ds.cam.ac.uk/clteach/grpproj/{group_name}

Some projects require use of special purpose hardware borrowed from clients, sponsors, or other sources. At the end of the project, all hardware

Good page numbers 8

ColorExtractor.java

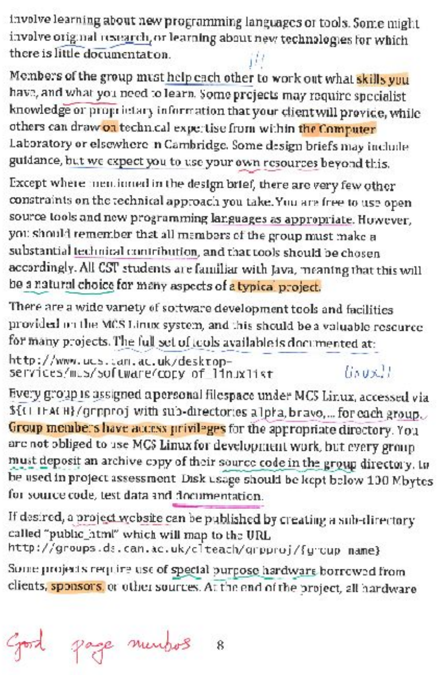
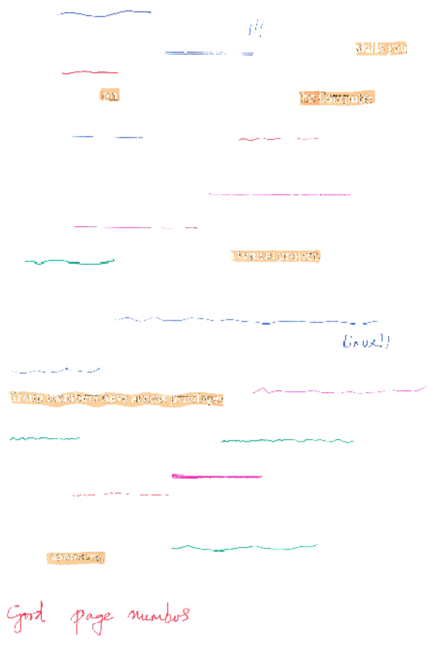
Public Methods

- **public static** BufferedImage extractColorComponent(BufferedImage original)

Code Examples

Extract an image of annotations (with transparent background) from an image of a modified document

```
BufferedImage input = ….;
BufferedImage differenceMap = ColorExtractor.extractColorComponent(input);
```

(BufferedImage) input	(BufferedImage) differenceMap
	

involve learning about new programming languages or tools. Some might involve original research, or learning about new technologies for which there is little documentation. !!!

Members of the group must help each other to work out what skills you have, and what you need to learn. Some projects may require specialist knowledge or proprietary information that your client will provide, while others can draw on technical expertise from within the Computer Laboratory or elsewhere in Cambridge. Some design briefs may include guidance, but we expect you to use your own resources beyond this.

Except where mentioned in the design brief, there are very few other constraints on the technical approach you take. You are free to use open source tools and new programming languages as appropriate. However, you should remember that all members of the group must make a substantial technical contribution, and that tools should be chosen accordingly. All CST students are familiar with Java, meaning that this will be a natural choice for many aspects of a typical project.

There are a wide variety of software development tools and facilities provided on the MCS Linux system, and this should be a valuable resource for many projects. The full set of tools available is documented at:
`http://www.ucs.cam.ac.uk/desktop-services/mcs/software/copy_of_linuxlist` Linux!!

Every group is assigned a personal filespace under MCS Linux, accessed via `${CLTEACH}/grpproj` with sub-directories alpha, bravo, ... for each group. Group members have access privileges for the appropriate directory. You are not obliged to use MCS Linux for development work, but every group must deposit an archive copy of their source code in the group directory, to be used in project assessment. Disk usage should be kept below 100 Mbytes for source code, test data and documentation.

If desired, a project website can be published by creating a sub-directory called "public_html" which will map to the URL `http://groups.cl.cam.ac.uk/clteach/grpproj/{group_name}`

Some projects require use of special purpose hardware borrowed from clients, sponsors, or other sources. At the end of the project, all hardware

Good page numbers 8

DBSCANClusterDetector.java

Public Methods

- **public** Collection<AnnotationBoundingBox> cluster(BufferedImage im)

Related Public Types

- AnnotationBoundingBox
- ClusteringPoint

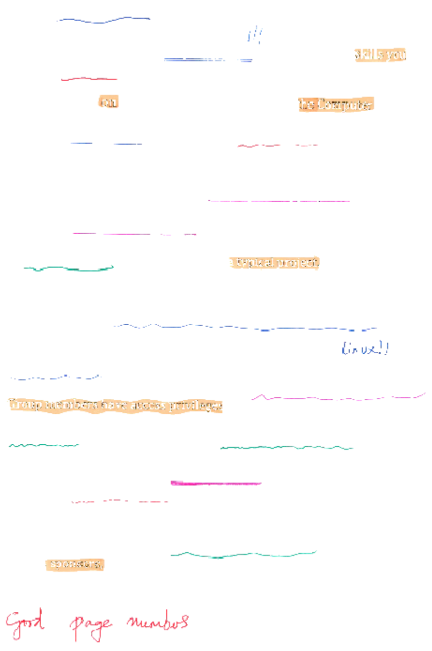
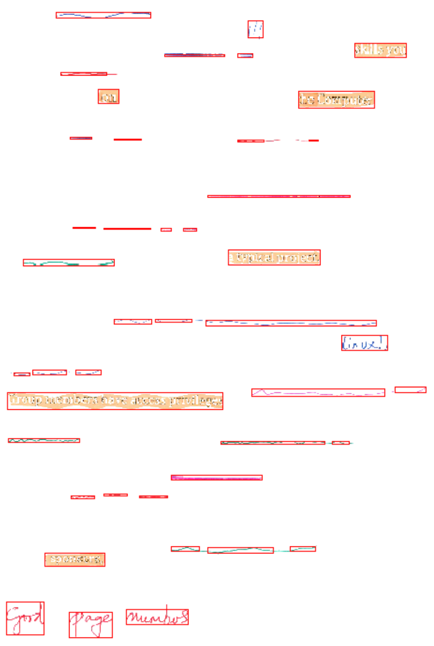
Code Examples

Get a list of AnnotationBoundingBox instances for the annotations detected in the image

```
BufferedImage im = ...;
Collection<AnnotationBoundingBox> clusters = cluster(im);
```

Produce an image with the clusters found by DBSCAN overlaid

```
String diffMapPath = ...;
String outPath = ...;
DBSCANClusterDetectorTest.overlayClusters(diffMapPath, outPath);
```

Image read from diffMapPath	Image saved to outPath
	

BasicClassifier.java

Public Methods

- `public Collection<Annotation> identifyAnnotations(BufferedImage fullImage, Collection<AnnotationBoundingBox> points, int pageNumber)`

Related Public Types

- Annotation (subclasses Underline, Highlight, Text)

Code Examples

Modifying the parameters of the BasicClassifier

The method identifySingleAnnotation

```
 private static Annotation identifySingleAnnotation(BufferedImage fullImage,
                                                    BufferedImage annotationSubImage,
                                                    AnnotationBoundingBox annotationBoundingBox,
                                                    int pageNumber,
                                                    boolean defaultToText,
                                                    double coverageThreshold,
                                                    double aspectRatioThreshold)
```

requires a full page image of the difference map (fullImage) so that the coordinates of the Annotation are correct with respect to the coordinate system used by PDFs (with the origin at the bottom left corner), and a single BufferedImage which the subimage corresponding to the annotation, in addition to the coordinates of the annotation bounding box, the page number on which this annotation is present, and the following parameters which control the behaviour of the basic classifier:

- defaultToText : if set to true, then the classifier will only attempt to classify the annotations into either highlights or text (in which case the raw image of the annotation is overlaid on the PDF by the next stage).
- coverageThreshold, aspectRatioThreshold are parameters controlling the decision boundaries. A larger value of coverageThreshold means that a larger percentage of the pixels must be non-transparent before we discard the possibility that the annotation is `text'. A larger aspectRatioThreshold means that the ratio of the width to the height (of the subimage of non-transparent pixels within the annotation subimage) must be larger before we choose `underline' rather than `highlight'.

Get a list of Annotation (including classification) from a BufferedImage of a modified document

```
IAnnotationIdentifier annId = new BasicClassifier();
Collection<AnnotationBoundingBox> clusterPoints = clusterDetector.cluster(scan);
Collection<Annotation> annotations = annId.identifyAnnotations(scan, clusterPoints, page);
```

Note: The value of `page' may be anything in this example.

PDFPenAndPaper.java

This class controls the whole pipeline and integrates all of the components together, as well as providing an interface between PDF documents (which are the top-level inputs and outputs) and BufferedImages corresponding to individual pages of the input PDF (which is the format used by the image processing components in the pipeline). To run the whole pipeline simply initialize an instance of the PDFPenAndPaper class.

```
new PDFPenAndPaper(originalFile, scannedFile, Path_to_output);
```

Where,

- `originalFile` **- is a java.io.File that is a PDF**
- `scannedFile` **- is a java.io.File that is a PDF**
- `Path_to_output` **- is a String of the path where the output PDF with annotations should be stored.**

Annotation.java

Annotation coordinates must be provided in PDF coordinates, i.e. with the origin at the bottom. The Annotation class is an abstract class that is extended by **Underline.java, HighLight.java, Text.java.**

- **Underline.java**
    - `UnderLine ul = new UnderLine(x, y, width, pageNumber);`
    - (float) x, y -- are the PDF coordinates of the top left corner of the annotation's bounding box.
    - (float) width -- is the width of the annotation's bounding box.
    - (int) pageNumber -- is the page of the PDF on which the annotation is to be applied.

- **Text.java**
    - `Text text = new Text(x, y, annotationImage, pageNumber);`
    - (float) x, y -- are the PDF coordinates of the top left corner of the annotation's bounding box.
    - (BufferedImage) annotationImage -- is the raw image of the annotation taken from the difference map produced by the extraction stage, with a transparent background.
    - (int) pageNumber -- is the page of the PDF on which the annotation is to be applied.

- **Highlight.java**

- `hl = new Highlight(x, y, width, height, pageNumber);`

- (float) x, y -- are the PDF coordinates of the top left corner of the annotation’s bounding box.
- (float) width, height -- are the width and height of the highlight respectively.
- (int) pageNumber -- is the page of the PDF on which the annotation is to be applied.