Content-based Video Retrieval for Pattern Matching Video Clips

Adam Jaamour

Bachelor of Science in Computer Science with Honours
The University of Bath
May 2019

Content-based Video Retrieval for Pattern Matching Video Clips

Submitted by: Adam Jaamour

COPYRIGHT

Declaration

This dissertation is submitted to the University of Bath in accordance with the requirements of the degree of Bachelor of Science in the Department of Computer Science. No portion of the work in this dissertation has been submitted in support of an application for any other degree or qualification of this or any other university or institution of learning. Except where specifically acknowledged, it is the work of the author.

Signed:

Abstract

Your abstract should appear here. An abstract is a short paragraph describing the aims of the project, what was achieved and what contributions it has made.

Contents

List of Figures

List of Tables

Acknowledgements

Add any acknowledgements here.

Chapter 1

Introduction

This is the introductory chapter.

1.1 Example Section

Like all chapters, it will have a number of sections

1.1.1 Example Subsection

. . . and sub-sections

Example sub-subsection

. . . and sub-subsections.

Table 1.1: An example table

Items	Values
Item 1	Value 1
Item 2	Value 2

1.2 Another section

Another section, just for good measure. You can reference a table, figure or equation using `\ref`, just like this reference to table 1.1.

1.3 Example lists

1.3.1 Enumerated

1. Example enumerated list
    - a nested enumerated list item
2. Second item in the list

1.3.2 Itemized

- Example itemized list
    - a nested itemized list item
- Second item in the list

1.3.3 Description

Item 1 Example description list

Item 2 Second item in the list

Chapter 2

Literature Survey

This is the chapter for your Literature Survey.

You will wish to cite authors like (**?**) or (**?**). Alternate commands are used to cite **?** as a noun, or cite **?** work possessively, or add text to the citation, (e.g. **?**).

If these citations do not compile correctly, ensure you have the Harvard package installed. You can pick up the Harvard package in the zip file of the dissertation template files you downloaded.

Chapter 3

Requirements

If you are doing a primarily software development project, this is the chapter in which you review the requirements decisions and critique the requirements process.

Chapter 4

Design

This is the chapter in which you review your design decisions at various levels and critique the design process.

Chapter 5

Implementation and Testing

This is the chapter in which you review the implementation and testing decisions and issues, and critique these processes.

Code can be output inline using `\lstinline|some code|`. For example, this code is inline: `public static int example = 0;` (I have used the character `|` as a delimiter, but any non-reserved character not in the code text can be used.)

Code snippets can be output using the `\begin{lstlisting} ... \end{lstlisting}` environment with the code given in the environment. For example, consider listing 5.1, below.

Listing 5.1: Example code

```
public static void main() {

  System.out.println("Hello World");

}
```

Code listings are produced using the package "Listings". This has many useful options, so have a look at the package documentation for further ideas.

Chapter 6

Results

This is the chapter in which you review the outcomes, and critique the outcomes process. You may include user evaluation here too.

Chapter 7

Conclusions

This is the chapter in which you review the major achievements in the light of your original objectives, critique the process, critique your own learning and identify possible future work.

Appendix A

Design Diagrams

Appendix B

User Documentation

Appendix C

Raw results output

Appendix D

Code

D.1 File: main.py

```
import cv2


video_capture =
    cv2.VideoCapture('../recordings/recording_circle_red_right.mov')
# video_capture =
    cv2.VideoCapture('../animations/output/circle_blue_right.avi')

if not video_capture.isOpened():
    print("Error opening video file")

# read video until completion or user exit
while video_capture.isOpened():
    # read capture frame by frame
    ret, frame = video_capture.read()

    if ret:
        # display current frame
        cv2.imshow('Frame', frame)
        # user exit on "q" key press
        if cv2.waitKey(25) & 0xFF == ord('q'):
            break
    else:
        break

# tidying upq
video_capture.release()
cv2.destroyAllWindows()
```