

Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

©Copyright

by

Vineet Kashyap

2011

to my

Mother, Father, Brother and Sister

with love



AutoTC:Automatic Time-code Recognition
for the purpose of synchronisation of subtitles
in the Broadcasting of Motion Pictures using the SMPTE standard

A thesis presented in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE in
COMPUTER SCIENCE

School of Engineering and Advanced Technology (S.E.A.T)

at Massey University, [Albany],

New Zealand.

by

VINEET KASHYAP, B.E., P.G. Dip. Computer Science

October 2011

Acknowledgements

I would like to thank my supervisor Dr. Tom Moir for helping me in my research who not only mentored me but guided and motivated me throughout the tenure of this challenging piece of research. Special thanks to New Zealand Maori Television for their time and co-operation which helped me in defining a clear set of requirements, limiting the scope for developing a software application. Last but not the least, sincere thanks to Yash Raj Films India Pvt. Ltd. for providing Hindi movie scripts. Thanks to Air New Zealand and Event cinemas for their input. My sincere respect and thanks to my parents for their unconditional love and support. Also, thanking Massey University and School of Engineering and Advanced Technology (SEAT) IT support staff for providing excellent support during my research.

A paper based on this research has also been submitted at Australasian Language Technology Workshop (ALTA) 2011.

Abstract

Time-coding requires manual marking of the in/out points of every dialogue spoken in the film. This process is very manual, cumbersome and time-consuming which takes about 8-10 hours for an average film duration of 2 and a half hours. AutoTC, a multi-threaded client-server application has been built for the automatic recognition of time-codes for the purpose of automatic synchronisation of subtitles in the broadcasting of Motion Pictures.

It involves generating time-codes programmatically based on the input video's frame rate to be subtitled and using the audio to recognise in/out points automatically using the principles of Voice Activity Detection. Results show that the time taken to recognise time-codes automatically is approximately 1/6th compared to the the time taken by a professional time-coder using 'Poliscript'[18], a commercial tool used in the production of subtitles. 'IN-SYNC', a new performance metric, has been proposed to evaluate the accuracy of the developed system which will foster further research and development in the field of automatic subtitling in an attempt to make it the de-facto standard. The application has been tested on the NOIZEUS[30] corpus giving an IN-SYNC accuracy of 65% on clean data with 6 mis-detections and an average of 51.56% on noisy data with 13 mis-detections which is very encouraging.

The application can also send data to the MOSES[32] server developed for producing draft translations from Hindi to English which will make the subtitling process much faster, efficient and quality-centric.

Contents

	Page
Acknowledgements	iv
Abstract	v
Table of Contents	vi
List of Tables	ix
List of Figures	x
List of Algorithms	xii
Chapter	
1 Introduction	1
1.1 Research Outline	6
1.2 Literature Review	10
1.2.1 Automatic Speech Recognition (ASR)	11
1.2.2 Poliscript	11
Reading speeds	12
Subtitle Duration and Interval	13
1.2.3 YouTube	13
1.2.4 Same Language Subtitling (SLS)	15
1.3 Applications	16
1.4 Initial Requirements and Objectives	18
1.5 Thesis Scope	19
2 Time-coding	22
2.1 Time-Code Formats	24
2.2 Time-Code Types	25

2.3	Limitations	28
2.3.1	Missing Frames	28
2.3.2	Time-code discontinuities	29
2.4	Subtitle Transmission	29
3	Voice Activity Detection (VAD)	31
3.1	Introduction	31
3.2	Current State-of-the-Art	33
3.2.1	Desirable aspects of VAD algorithms	33
3.2.2	Frame Size and Windowing	34
3.2.3	Initial Value of Threshold	35
3.2.4	Feature Extraction	35
3.2.5	Adaptive Thresholding[6]	36
3.3	Basic Architecture	38
3.4	Algorithm Design	38
3.4.1	Time-code Generation	39
3.4.2	Time-code Recognition	40
3.4.3	Calculate Algorithm Accuracy	41
4	Experiments, Results and Discussion	45
4.1	Out-of-domain evaluation	47
4.2	Optimisations	48
4.2.1	Stereo-to-Mono Conversion	48
4.2.2	Down-sampling	53
4.3	Implementation Issues	54
5	Conclusion and Future Work	56
5.1	Conclusion	56
5.2	Future Work	57

References	58
Appendix	
A Time-coding using Poliscript	63
A.1 Selecting the programme to import	63
A.2 Re-stripping the Time Code	64
B User Manual	68
B.1 Setup Guide for Running AutoTC on a 64-bit Windows 7	68
B.2 Setup Guide for Running AutoTC on a 32-bit Windows XP	69
B.3 User Interface	70
B.3.1 Time-codes & Setup Wizard	71
B.3.2 Generate SMPTE time-code	72
B.3.3 Recognise SMPTE time-code	73
B.3.4 Translate	73
B.3.5 Final preview	74
C Time-coding using AutoTC	77
D Software Language and Compiler	78
D.1 Software Language and Compiler	78
D.2 Third-Party Libraries	78
D.2.1 Aforge.NET Framework [19]	78
D.2.2 Tesseract [20]	79
D.2.3 XMLRPC.NET [21]	79
D.2.4 C# WAV Class[22]	80

List of Tables

1.1	Media File Formats for Subtitle Preparation	6
1.2	Subtitle Duration & Interval	14
2.1	Frame Rates	26
3.1	List of sentences used in NOIZEUS[30]	43
4.1	Time taken for Recognising time-codes manually using Poliscript	46
4.2	Time taken for Recognising time-codes automatically using AutoTC	46
4.3	Time-Code detection factors with their sub-optimal values	47
4.4	Performance of OCR based on Font Type	48
4.5	Performance of TCR for speech with different noise sources	49
4.6	Summary of the WAV file	50

List of Figures

1.1	Corpus size vs. BLEU scores	8
1.2	The process of subtitling	9
1.3	Adding and Editing Captions/subtitles using YouTube	15
1.4	Sub-Rip subtitle file format	20
2.1	Burnt-in type time-code (BITC)	22
2.2	Film reel	25
3.1	Block diagram of a VAD	32
3.2	2-second audio waveform	33
3.3	Frame overlap	35
3.4	Feature values extracted and plotted using MATLAB for sp02.wav from the NOIZEUS corpus	36
3.5	Impulse Response of $H(z)$ for $p = 0.2$	37
3.6	Fall-time for different values of p	38
3.7	Basic Architecture	39
3.8	Flowchart for the proposed TCR for finding the IN point	44
4.1	Stereo track with two distinct channels	51
4.2	Stereo splitting	52
4.3	Deleting one channel from the stereo track	53
4.4	Stereo combining	54
B.1	Enable codec using ffdshow	69
B.2	Application Main Window	70

LIST OF FIGURES

B.3 Application File Menu	71
B.4 Application Help Menu	72
B.5 Setup Time-Code	72
B.6 Application Run Wizard	73
B.7 For producing draft translations	74
B.8 Translation using MOSES Server	75
B.9 Translation using Client AutoTC	75
B.10 Preview Video	76

List of Algorithms

1	Pseudo-code for the proposed Time-code Generation algorithm	40
2	Pseudo-code for computing the accuracy	42