SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Data Engineering & Analytics

Optimization of Video-based Facial Data Extraction for Time-series Correlation Analysis using Distributed-Data Processing Framework

Norma Puspitasari

SCHOOL OF COMPUTATION, INFORMATION AND TECHNOLOGY — INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Data Engineering & Analytics

Optimization of Video-based Facial Data Extraction for Time-series Correlation Analysis using Distributed-Data Processing Framework

Optimierung Videobasierter Gesichtsdatenextraktion zur Zeitreihenkorrelationsanalyse unter Verwendung eines Verteilten

Datenverarbeitungssystems

Author:	Norma Puspitasari
Supervisor:	Prof. Gudrun Klinker, PhD
Advisor:	Chloe Eghtebas
Submission Date:	15.12.2023

I confirm that this master's thesis is my own work and I have documented all sources and material used.

Munich, 15.12.2023 Norma Puspitasari

Acknowledgments

I would like to express my deepest gratitude to my supervisor Prof. Dr. Gudrun Klinker for providing me with the opportunity to work on such an interesting topic. I also want to thank my advisor Chloe Eghtebas for her thoughtful advice and support. I am really grateful for her willingness to meet with her students regularly, her time to help me when it feels like a dead end and her insightful feedback. I am also grateful to my family and friends for their continuous support and love throughout my studies.

Abstract

Facial extraction data plays a crucial role in synchrony measurement. There are open source tools available now for extracting Facial Action Units (FAU) data, such as OpenFace and PyFeat. However, it is still difficult to run a user study and have a real-time analysis of Facial Action Units (FAU) data at the same time because the tools are built either as a standalone application or a library for offline usage. Currently, we are building a video-conferencing platform that initiates the implementation of analysis filter for research purposes which still has a lot of rooms for improvement, such as low running fps, mixed frames and data lost. This thesis explores the optimization of facial extraction tool integration to the video-conferencing platform by utilizing OpenFace as a case study. The methodology involves implementing RabbitMQ as a robust queue system to manage and process facial data in real-time. This thesis also introduces a feature that enabled user to have the facial extraction data after running an experiment in the platform called post-processing. Finally, the real-time data was compared with the post-processing one and it showed that they have similar values. The network and memory measurement was also conducted and confirmed that it improved the real-time analysis filter performance.

Kurzfassung

Gesichtsextraktionsdaten spielen eine entscheidende Rolle bei der Synchronitätsmessung. Es gibt inzwischen Open-Source-Tools für die Extraktion von Facial Action Units (FAU)-Daten, wie z. B. OpenFace und PyFeat. Es ist jedoch nach wie vor schwierig, eine Nutzerstudie durchzuführen und gleichzeitig eine Echtzeitanalyse von FAU-Daten (Facial Action Units) vorzunehmen gleichzeitig durchzuführen, da die Tools entweder als eigenständige Anwendung oder als Bibliothek für die Offline-Nutzung. Derzeit entwickeln wir eine Videokonferenzplattform, die die die Implementierung von Analysefiltern für Forschungszwecke, die noch viel Raum für Verbesserungen bietet, wie z.B. niedrige Bildwiederholrate, gemischte Bilder und Datenverluste. Diese Arbeit untersucht die Optimierung der Integration des Gesichtsextraktionstools in die Videokonferenzplattform anhand von OpenFace als Fallstudie. Die Methodik beinhaltet die Implementierung von RabbitMQ als robustes Warteschlangensystem zur Verwaltung und Verarbeitung von Gesichtsdaten in Echtzeit. Diese Arbeit führt auch eine Funktion ein, die eine Funktion vor, die es dem Benutzer ermöglicht, die Gesichtsextraktionsdaten nach der Durchführung eines Experiments in der Plattform zu erhalten, die so genannte Nachbearbeitung. Schließlich wurden die Echtzeitdaten mit den nachbearbeiteten Daten verglichen und es zeigte sich, dass sie ähnliche Werte aufweisen. Die Netzwerk- und Speichermessungen wurden ebenfalls durchgeführt und bestätigten, dass die Leistung des Echtzeit-Analysefilters verbessert wurde.

Contents

1 Introduction

Video-conferencing platforms have become indispensable tools, serving as conduits for professional collaborations, social engagements, and educational endeavors. As these virtual spaces continue to shape the landscape of contemporary communication, the need to enhance the quality of interactions becomes paramount. This thesis embarks on an exploration into the integration of a facial extraction tool, exemplified by OpenFace, utilizing a queue system powered by RabbitMQ, to advance the measurement of synchrony within video-conferencing platforms.

The integration of the facial extraction tool is motivated by a desire to decode the subtle nuances of non-verbal communication, specifically focusing on facial expressions, gestures, and emotional cues. By delving into these aspects, we aim to uncover the intricate elements that contribute to effective virtual engagement. As a representative case study, OpenFace serves as a powerful tool in this endeavor, offering advanced algorithms capable of real-time extraction of facial features.

Moreover, this research addresses the broader implications of incorporating the facial extraction tool into virtual communication environments. Privacy concerns and ethical considerations are integral aspects of this study, emphasizing the responsible implementation of technology to respect user privacy and ensure data security. The implementation of RabbitMQ as a queue system further enriches this exploration, providing a robust and scalable framework for the seamless processing of facial data.

Through this endeavor, we anticipate refining video-conferencing platforms by integrating the facial extraction tool, thereby fostering a more comprehensive understanding of synchrony during virtual interactions. The case study with OpenFace, coupled with the queue system facilitated by RabbitMQ, not only contributes to the technical aspects of integration but also underscores the importance of responsible and ethical practices in the utilization of facial extraction tools in virtual environments.

Citation test [Lam94].

Acronyms must be added in `main.tex` and are referenced using macros. The first occurrence is automatically replaced with the long version of the acronym, while all subsequent usages use the abbreviation.

E.g. `\ac{TUM}, \ac{TUM}` $\Rightarrow$ Technical University of Munich (TUM), TUM

For more details, see the documentation of the `acronym` package[1].

[1]`https://ctan.org/pkg/acronym`

1.0.1 Subsection

See Table 7.1, Figure 7.1, Figure 7.2, Figure 7.3.

Table 1.1: An example for a simple table.

A	B	C	D
1	2	1	2
2	3	2	3

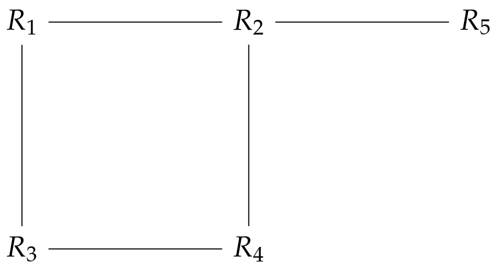


Figure 1.1: An example for a simple drawing.

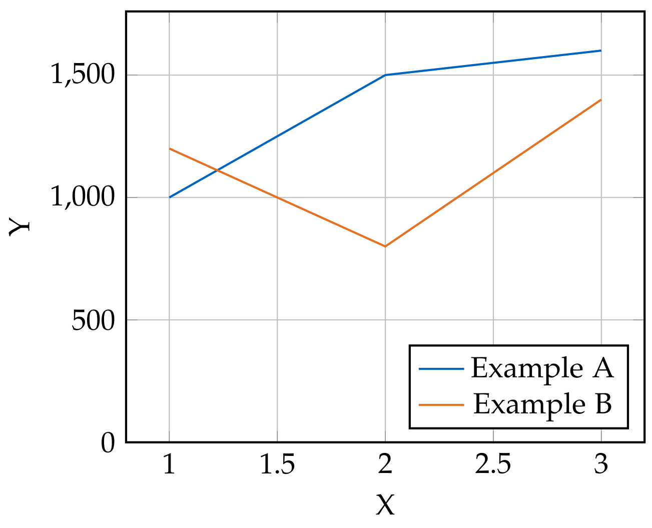


Figure 1.2: An example for a simple plot.

```
SELECT * FROM tbl WHERE tbl.str = "str"
```

Figure 1.3: An example for a source code listing.

2 Related Work

2.1 Section

Citation test [Lam94].

Acronyms must be added in `main.tex` and are referenced using macros. The first occurrence is automatically replaced with the long version of the acronym, while all subsequent usages use the abbreviation.

E.g. `\ac{TUM}, \ac{TUM}` $\Rightarrow$ TUM, TUM

For more details, see the documentation of the `acronym` package[1].

2.1.1 Subsection

See Table 7.1, Figure 7.1, Figure 7.2, Figure 7.3.

Table 2.1: An example for a simple table.

A	B	C	D
1	2	1	2
2	3	2	3

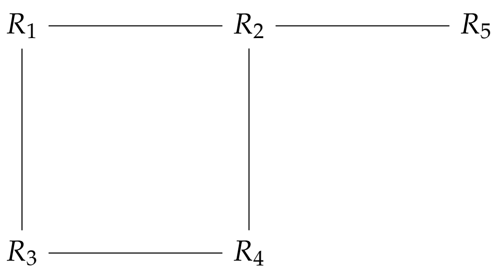


Figure 2.1: An example for a simple drawing.

[1]`https://ctan.org/pkg/acronym`

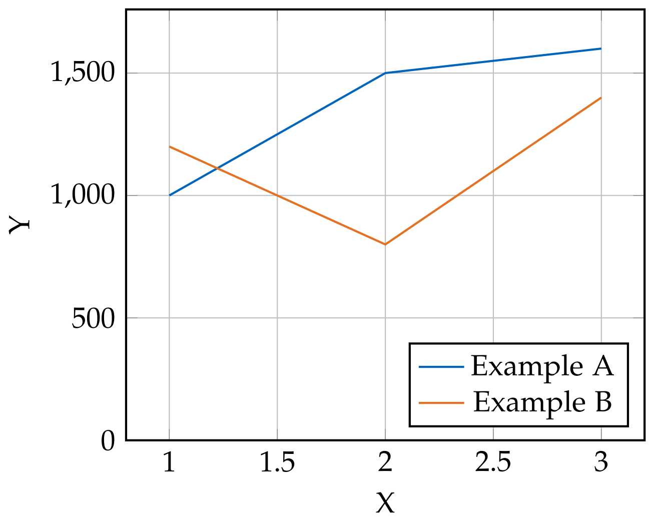


Figure 2.2: An example for a simple plot.

```
SELECT * FROM tbl WHERE tbl.str = "str"
```

Figure 2.3: An example for a source code listing.

3 Requirements Evaluation

3.1 Section

Citation test [Lam94].

Acronyms must be added in `main.tex` and are referenced using macros. The first occurrence is automatically replaced with the long version of the acronym, while all subsequent usages use the abbreviation.

E.g. `\ac{TUM}, \ac{TUM}` ⇒ TUM, TUM

For more details, see the documentation of the `acronym` package[1].

3.1.1 Subsection

See Table 7.1, Figure 7.1, Figure 7.2, Figure 7.3.

Table 3.1: An example for a simple table.

A	B	C	D
1	2	1	2
2	3	2	3

$$\begin{tikzcd}
R_1 \arrow[r, dash] \arrow[d, dash] & R_2 \arrow[r, dash] \arrow[d, dash] & R_5 \\
R_3 \arrow[r, dash] & R_4 &
\end{tikzcd}$$

Figure 3.1: An example for a simple drawing.

[1]`https://ctan.org/pkg/acronym`

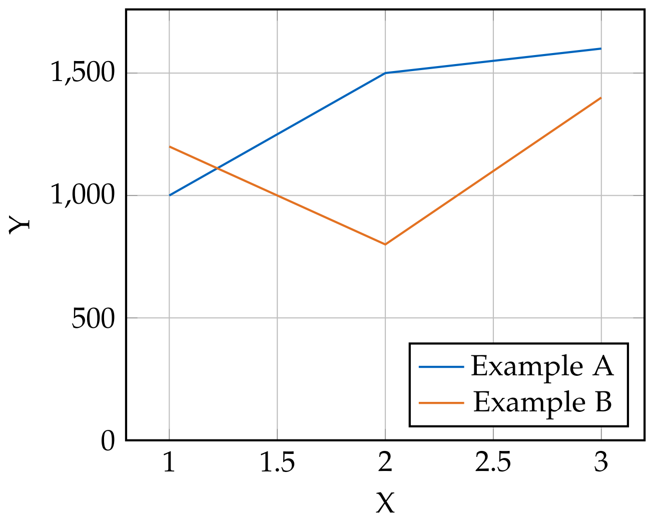


Figure 3.2: An example for a simple plot.

```
SELECT * FROM tbl WHERE tbl.str = "str"
```

Figure 3.3: An example for a source code listing.

4 Implementation

4.1 Section

Citation test [Lam94].

Acronyms must be added in `main.tex` and are referenced using macros. The first occurrence is automatically replaced with the long version of the acronym, while all subsequent usages use the abbreviation.

E.g. `\ac{TUM}, \ac{TUM}` $\Rightarrow$ TUM, TUM

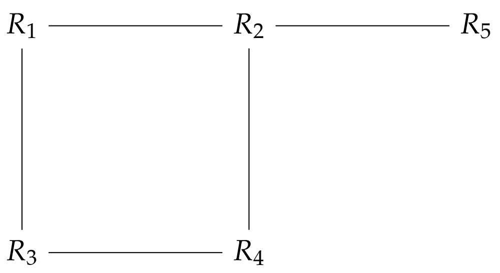
For more details, see the documentation of the `acronym` package[1].

4.1.1 Subsection

See Table 7.1, Figure 7.1, Figure 7.2, Figure 7.3.

Table 4.1: An example for a simple table.

A	B	C	D
1	2	1	2
2	3	2	3



$$\begin{tikzcd} R_1 \arrow[r, dash] \arrow[d, dash] & R_2 \arrow[r, dash] \arrow[d, dash] & R_5 \\ R_3 \arrow[r, dash] & R_4 & \end{tikzcd}$$

Figure 4.1: An example for a simple drawing.

[1]`https://ctan.org/pkg/acronym`

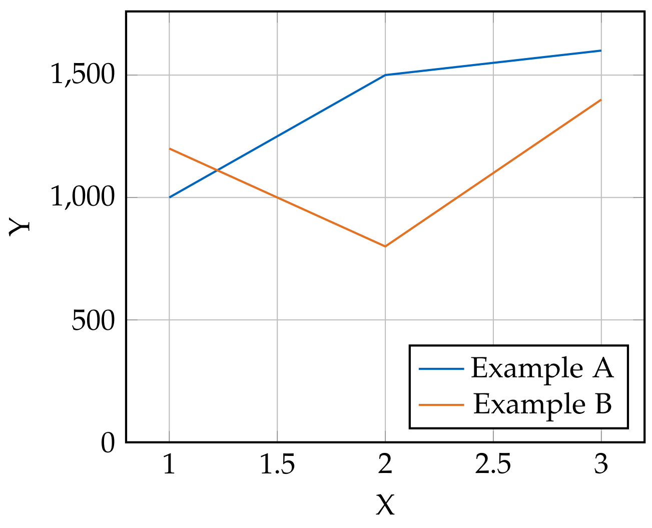


Figure 4.2: An example for a simple plot.

```
SELECT * FROM tbl WHERE tbl.str = "str"
```

Figure 4.3: An example for a source code listing.

5 Method

5.1 Section

Citation test [Lam94].

Acronyms must be added in `main.tex` and are referenced using macros. The first occurrence is automatically replaced with the long version of the acronym, while all subsequent usages use the abbreviation.

E.g. `\ac{TUM}, \ac{TUM}` $\Rightarrow$ TUM, TUM

For more details, see the documentation of the `acronym` package[1].

5.1.1 Subsection

See Table 7.1, Figure 7.1, Figure 7.2, Figure 7.3.

Table 5.1: An example for a simple table.

A	B	C	D
1	2	1	2
2	3	2	3

$$\begin{tikzcd}
R_1 \arrow[r, dash] \arrow[d, dash] & R_2 \arrow[r, dash] \arrow[d, dash] & R_5 \\
R_3 \arrow[r, dash] & R_4 &
\end{tikzcd}$$

Figure 5.1: An example for a simple drawing.

[1]`https://ctan.org/pkg/acronym`

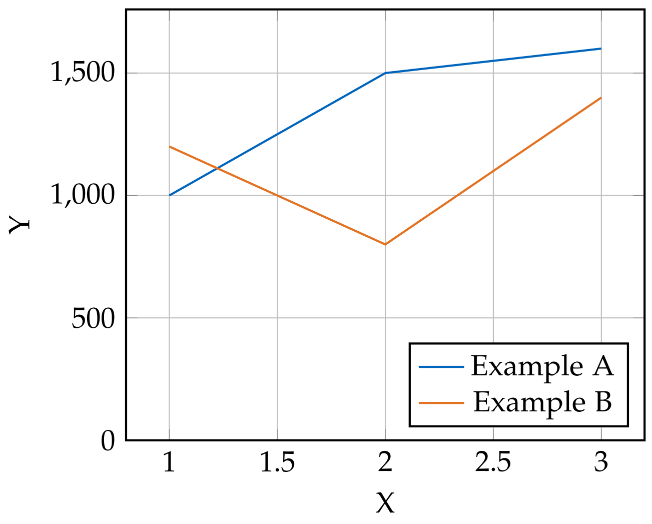


Figure 5.2: An example for a simple plot.

```
SELECT * FROM tbl WHERE tbl.str = "str"
```

Figure 5.3: An example for a source code listing.

6 Results

6.1 Section

Citation test [Lam94].

Acronyms must be added in `main.tex` and are referenced using macros. The first occurrence is automatically replaced with the long version of the acronym, while all subsequent usages use the abbreviation.

E.g. `\ac{TUM}, \ac{TUM}` ⇒ TUM, TUM

For more details, see the documentation of the `acronym` package[1].

6.1.1 Subsection

See Table 7.1, Figure 7.1, Figure 7.2, Figure 7.3.

Table 6.1: An example for a simple table.

A	B	C	D
1	2	1	2
2	3	2	3

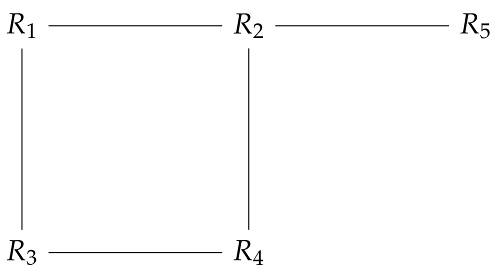


Figure 6.1: An example for a simple drawing.

[1]`https://ctan.org/pkg/acronym`

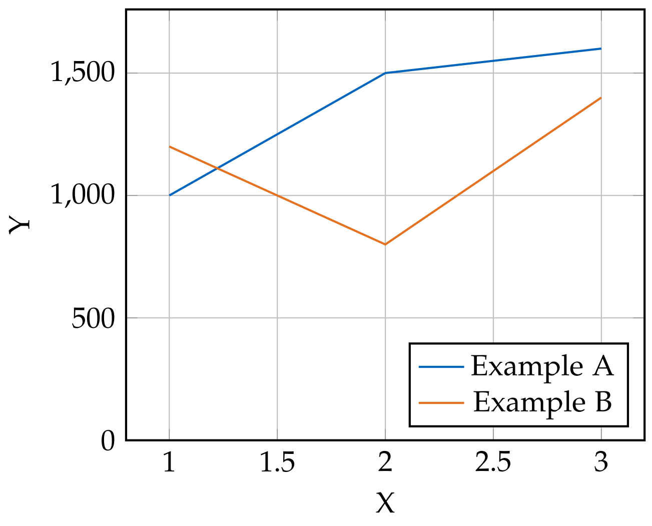


Figure 6.2: An example for a simple plot.

```
SELECT * FROM tbl WHERE tbl.str = "str"
```

Figure 6.3: An example for a source code listing.

7 Discussion

7.1 Section

Citation test [Lam94].

Acronyms must be added in `main.tex` and are referenced using macros. The first occurrence is automatically replaced with the long version of the acronym, while all subsequent usages use the abbreviation.

E.g. `\ac{TUM}, \ac{TUM}` $\Rightarrow$ TUM, TUM

For more details, see the documentation of the `acronym` package[1].

7.1.1 Subsection

See Table 7.1, Figure 7.1, Figure 7.2, Figure 7.3.

Table 7.1: An example for a simple table.

A	B	C	D
1	2	1	2
2	3	2	3

$$\begin{tikzcd}
R_1 \arrow[r, no head] \arrow[d, no head] & R_2 \arrow[r, no head] \arrow[d, no head] & R_5 \\
R_3 \arrow[r, no head] & R_4 &
\end{tikzcd}$$

Figure 7.1: An example for a simple drawing.

[1] `https://ctan.org/pkg/acronym`

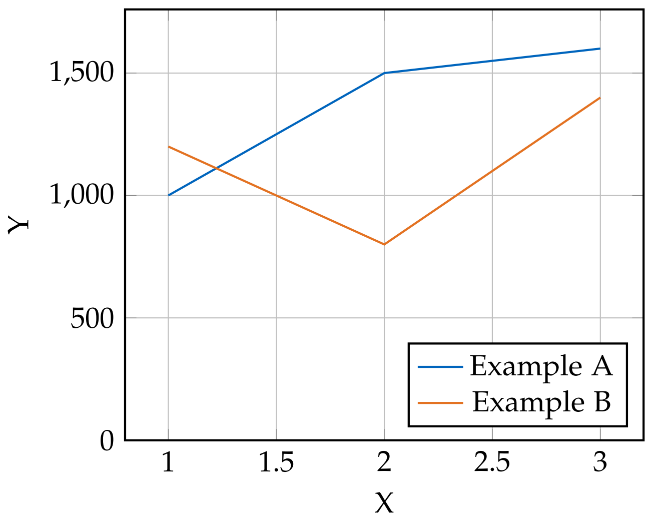


Figure 7.2: An example for a simple plot.

```
SELECT * FROM tbl WHERE tbl.str = "str"
```

Figure 7.3: An example for a source code listing.

Abbreviations

TUM	Technical University of Munich

List of Figures

List of Tables

Bibliography

[Lam94] L. Lamport. *LaTeX : A Documentation Preparation System User's Guide and Reference Manual*. Addison-Wesley Professional, 1994.