Dipartimento di Informatica, Bioingegneria,
Robotica ed Ingegneria dei Sistemi

Evaluation of human movement qualities: A methodology based on transferable-utility games on graphs

by

Ksenia Kolykhalova

Theses Series **DIBRIS-TH-2018-XX**

DIBRIS, Università di Genova

Via Opera Pia, 13 16145 Genova, Italy http://www.dibris.unige.it/

Università degli Studi di Genova

Dipartimento di Informatica, Bioingegneria,

Robotica ed Ingegneria dei Sistemi

Dottorato di Ricerca in Informatica

Ph.D. Thesis in Computer Science

Evaluation of human movement qualities: A methodology based on transferable-utility games on graphs

by

Ksenia Kolykhalova

February, 2018

Dottorato di Ricerca in Informatica
Dipartimento di Informatica, Bioingegneria, Robotica ed Ingegneria dei Sistemi
Università degli Studi di Genova

DIBRIS, Univ. di Genova
Via Opera Pia, 13
I-16145 Genova, Italy
http://www.dibris.unige.it/

Ph.D. Thesis in Computer Science (S.S.D. INF/01)

Submitted by Ksenia Kolykhalova
DIBRIS, Univ. di Genova
ksenia.kolykhalova@dibris.unige.it

Date of submission: February 2018

Title: Evaluation of human movement qualities:
A methodology based on transferable-utility games on graphs

Advisor:
Marcello Sanguineti
Dipartimento di Informatica, Bioingegneria, Robotica ed Ingegneria dei Sistemi
Università di Genova
marcello.sanguineti@unige.it

Gualtiero Volpe
Dipartimento di Informatica, Bioingegneria, Robotica ed Ingegneria dei Sistemi
Università di Genova
gualtiero.volpe@unige.it

Giorgio Gnecco
IMT School for Advanced Studies Lucca
giorgio.gnecco@imtlucca.it

Antonio Camurri
Dipartimento di Informatica, Bioingegneria, Robotica ed Ingegneria dei Sistemi
Università di Genova
antonio.camurri@unige.it

Ext. Reviewers:

Rossana Damiano
Dipartimento di Informatica
Università degli Studi di Torino
rossana@di.unito.it

Vito Fragnelli
Dipartimento di Scienze e Innovazione Tecnologica
Università del Piemonte Orientale
vito.fragnelli@uniupo.it

Abstract

A novel computational method for the analysis of expressive full-body movement qualities is introduced, which exploits concepts and tools from graph theory and game theory. The human skeletal structure is modeled as an undirected graph, where the joints are the vertices and the edge set contains both physical and nonphysical links.

Physical links correspond to connections between adjacent physical body joints (e.g., the forearm, which connects the elbow to the wrist). Nonphysical links act as "bridges" between parts of the body not directly connected by the skeletal structure, but sharing very similar feature values. The edge weights depend on features obtained by using Motion Capture data. Then, a mathematical game is constructed over the graph structure, where the vertices represent the players and the edges represent communication channels between them. Hence, the body movement is modeled in terms of a game built on the graph structure. Since the vertices and the edges contribute to the overall quality of the movement, the adopted game-theoretical model is of cooperative nature.

A game-theoretical concept, called Shapley value, is exploited as a centrality index to estimate the contribution of each vertex to a shared goal (e.g., to the way a particular movement quality is transferred among the vertices). The proposed method is applied to a data set of Motion Capture data of subjects performing expressive movements, recorded in the framework of the H2020-ICT-2015 EU Project WhoLoDance, Project no. 688865. Results are presented: development of novel method, contribution to the scientific community with a new data corpus, application the discussed method to 100 movement recordings and creation of database archive of stimuli for further use in research studies in the framework of the WhoLoDance Project.

Dedicated to the loving memory of my mother, Nelli Kolykhalova, who was very proud of me and supportive in every my life choice, my guiding light, my shining star, always in my heart forever.

Ksenia

Acknowledgements

This dissertation concludes three years of study and research. I am deeply grateful to my advisors Antonio Camurri, Gualtiero Volpe, Marcello Sanguineti and Giorgio Gnecco for their support and supervision throughout this time.

I thank my colleagues of Casa Paganini Infomus Research Center for making our laboratory such an enjoyable workplace. I thank my colleagues, who became my good friends who supported and motivated me throughout my academic journey, namely Stefano Piana, Paolo Alborno, Simone Ghisio, Paolo Coletta, Maurizio Mancini, Radoslaw Niewiadomski and Erica Volta.

I would like to thank my brother Artom and my father Valery for their unconditional love and belief in me.

Table of Contents

List of Figures

Part I

Background

Chapter 1

Introduction

Contents

1.1 Overview

The main purpose of this Ph.D thesis is the development of a novel approach and the associated computational method for the analysis of expressive full-body movement qualities, combining tools and methods from graph theory and cooperative game theory.

The proposed method first models the human skeletal structure as an undirected graph, where the joints are the vertices and the edge set contains both physical and nonphysical links. The edge weights depend on features obtained by using Motion Capture data.

Physical links correspond to connections between adjacent physical body joints (e.g., the forearm, which connects the elbow to the wrist).

Nonphysical links act as "bridges" between parts of the body not directly connected by the skeletal structure, but sharing very similar feature values. Then, we construct a mathematical game [63] over this skeletal graph, in which the vertices represent the players of the game, and the edges represent communication channels between the players. Body movement is modeled in terms of a cooperative game [63] built on the graph structure. Since the vertices and the edges contribute to the overall quality of the movement, the adopted game-theoretical model is a cooperative one. The so-called Shapley value [63], which provides a criterion to rank the players according to their importance, is then computed for such a game and used as a centrality index to estimate the contribution of each vertex to a shared goal (e.g., to the way a particular movement quality is transferred among the vertices).

The methodology is applied on a data set of Motion Capture data of subjects performing expressive movements, recorded in the framework of the H2020-ICT-2015 EU Project WhoLoDance. Results are presented: development of a novel method, contribution to the scientific community with a new data corpus, application of the discussed method to 100 movement recordings and creation of a database archive of stimuli for further use in research studies in the framework of the WhoLoDance Project. Using this approach for the analysis of movement features related to expressive gestures and emotional communication can enable the design of multimodal interfaces involving full-body human interaction, which communicate nonverbal expressive and emotional content.

1.2 Research Objective

The thesis focuses on the development of novel computational methods for the analysis of expressive full-body movement qualities, which exploits concepts and tools from graph theory and game theory. Previously game and graph theory techniques haven't been jointly used in the analysis of movement performed only by one person, in intra-personal way.

The developed approach can be summarized as follows, starting from the study of movement analysis in general and finishing with the application of the developed methods.

The work on the development of approach, presented in this dissertation is divided into four mayor blocks:

1. Movement features and expressive qualities
2. Motion capture data and sensory data acquisition
3. Computational models and feature extraction
4. Analysis of movement

Figure 1.1 presents the general work plan divided into these 4 parts.

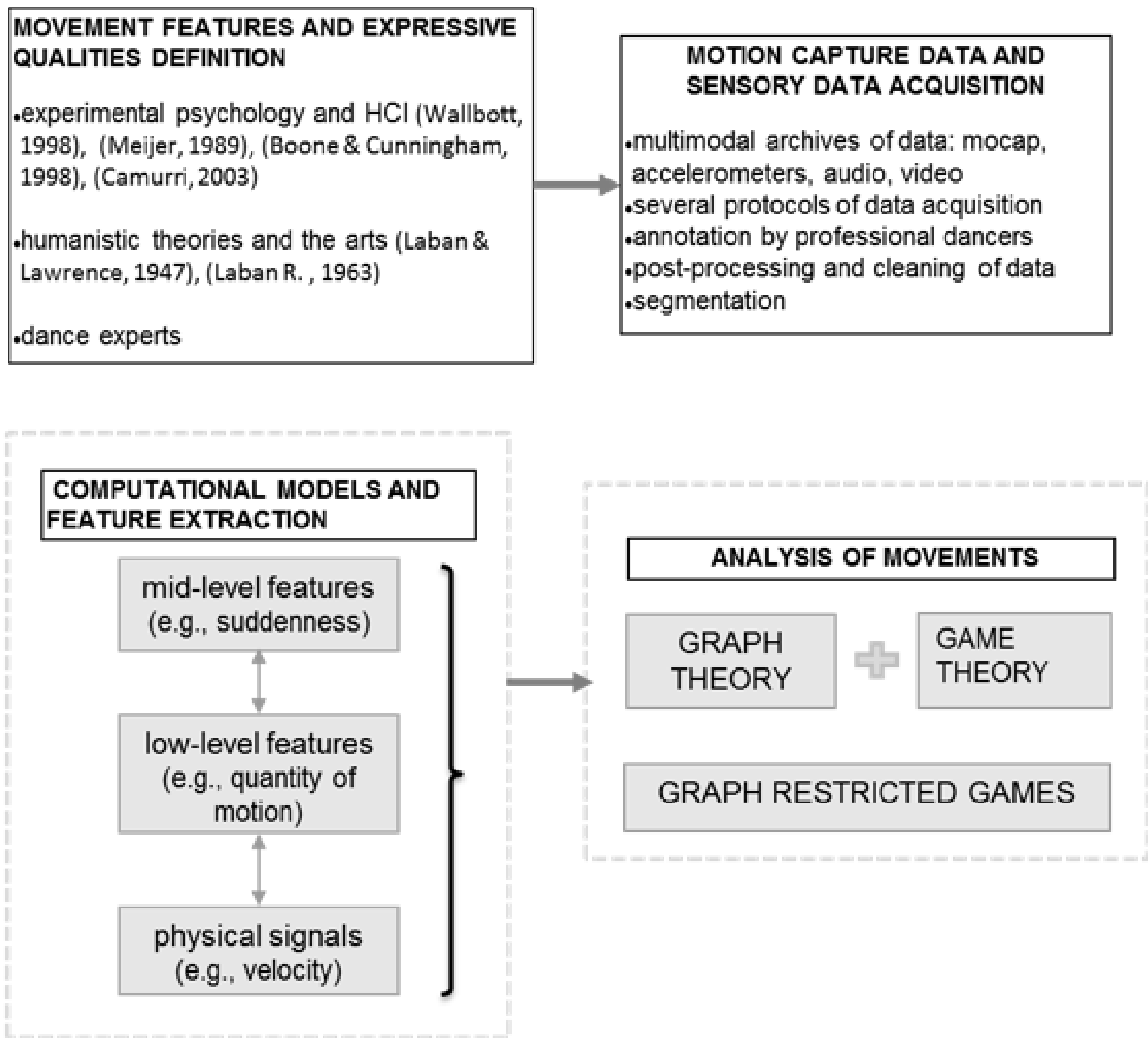

Figure 1.1: General Plan of the Work

1.3 Research Contribution

1.3.1 Major Contributions

The major contributions are the following:

- **Contribution 1**: Investigation of techniques for extracting and analyzing movement features related to expressive gestures and emotional communication.

- **Contribution 2**: Development of a novel approach and the associated computational method for the analysis of expressive full-body movement qualities, combining tools and methods from graph theory and cooperative game theory.
- **Contribution 3**: Creation of a new database of movement segments for use by the scientific community. Application of the proposed method on the created database of movement segments.

Despite the fact that the combination of graph and game theory techniques is already used in several research fields, to the best of the authors' knowledge, approaches aimed at the analysis of expressive gestures through a combination of these methods have not yet been developed.

First research contributions of the proposed method in the framework of this dissertation has been presented and published in:

Ksenia Kolykhalova, Giorgio Gnecco, Marcello Sanguineti, Antonio Camurri, and Gualtiero Volpe. 2017. Graph-restricted game approach for investigating human movement qualities. In Proceedings of the 4th International Conference on Movement Computing (MOCO '17), Kiona Niehaus (Ed.). ACM, New York, NY, USA, Article 30, 4 pages.
DOI: https://doi.org/10.1145/3077981.3078030

The work is partially supported by H2020-ICT-2015 EU Project WhoLoDance, Project no. 688865

1.3.2 Additional Contributions

Before the application of mathematical tools and the development of new methods for the analysis of expressive features of human movements, one has to understand the nature and mechanisms of the expressive movement features.

During my Ph.D program, I have contributed to several European Projects that were focused on different aspects of the analysis of human movements. Those contributions became the basis of this Ph.D dissertation and a fundamental starting point for the development a further method based on transferable-utility games on graphs applied for analysis of expressive features of human-body movements.

Since during the Ph.D i worked on several topics in movement analysis, below the projects on movement analysis to which i contributed are listed together with my publications.

These publications can be considered a useful contribution to the fundamental basis of the dissertation, however the present work is mostly focused on the graph-restricted game approach.

Therefore, we do not discuss publications in detail and leave them to the interested of the reader.

European International Projects

- EU ICT H2020 DANCE "Dancing in the dark" Project n.645553
  1) Development of serious game platform for validating sonification of human full-body movement qualities.
  2) Contribution to the creation of several multimodal datasets of movement qualities.

- EU ICT H2020 TELMI "Technology Enhanced Learning of Musical Instrument Performance" Project n. 688269
  Contribution to the creation of a multimodal corpus for technology-enhanced learning of violin playing, concerning the movement features for a musical instrument playing.

- EU ICT H2020 WHOLODANCE "Whole-body interaction learning for dance Education" Project n. 688865
  1) Contribution to the creation of multimodal datasets of movement qualities.
  2) Design of novel approach based on graph and game theory techniques.
  3) Creation a dataset of one hundred movement segments with application of the graph restricted game approach, as it can be used for the investigation of the propagation of the movement and the concept of origin of the movement.
  4) Design of evaluation platform.

Publications

- Kolykhalova, Ksenia, Antonio Camurri, Gualtiero Volpe, Marcello Sanguineti, Enrico Puppo and Radoslaw Niewiadomski. "A multimodal dataset for the analysis of movement qualities in karate martial art." 2015 7th International Conference on Intelligent Technologies for Interactive Entertainment (INTETAIN) (2015): 74-78.

- Ksenia Kolykhalova, Paolo Alborno, Antonio Camurri, and Gualtiero Volpe. 2016. A serious games platform for validating sonification of human full-body movement qualities. In Proceedings of the 3rd International Symposium on Movement and Computing (MOCO '16). ACM, New York, NY, USA, , Article 39 , 5 pages.
  DOI:https://doi.org/10.1145/2948910.2948962

- Gualtiero Volpe, Ksenia Kolykhalova, Erica Volta, Simone Ghisio, George Waddell, Paolo Alborno, Stefano Piana, Corrado Canepa, and Rafael Ramirez-Melendez. 2017. A multi-modal corpus for technology-enhanced learning of violin playing. In Proceedings of the 12th Biannual Conference on Italian SIGCHI Chapter (CHItaly '17). ACM, New York, NY, USA, Article 25, 5 pages. DOI: https://doi.org/10.1145/3125571.3125588

- Niewiadomski, R., Kolykhalova, K., Piana, S., Alborno, P., Volpe, G., Camurri, A., Analysis of Movement Quality in Full-Body Physical Activities, accepted to: ACM Transaction On Interactive Intelligent Systems. doi: 10.1145/3132369

1.4 Dissertation structure

This thesis is divided into four parts: 1) Background, 2) Approach, 3) Experimental setup and Software, 4) Future work and Conclusion, containing nine chapters in total.

Chapter 2: This part of the document presents the theoretical and mathematical preliminaries of Graph and Game Theory separately, starting from their history and impact on scientific world, finishing with a useful mathematical notation that is used throughout the dissertation.

Chapter 3: Here we present the scientific literature review and state of the art concerning the main topics we discuss in this work. More in detail, first we introduce the state of the art in the analysis of movement qualities, then we present a review of applications of Graph Theory separated for different disciplines in science and movement analysis respectively as a subsections. After that we review the application areas of Game Theory, as well in science and movement and behavioral analysis separately. We conclude this chapter with the investigation on how Game and Graph Theory were applied together as a method in several studies.

Chapter 4: For a better understanding of the method, we proceed with an introduction of the developed methodology and a formulation of the Problem and a Plan of implementation.

Chapter 5: In this chapter, we introduce some theoretical properties of the cooperative game constructed on a graph.

Chapter 6: In this chapter we present the methodology which was followed in order to design the Motion Capture Sessions, in a way that maximized the impact of the technologies used, to optimize the time management, select the movement sequences that we needed to capture and store, and build a repository of movement which will cover the needs of the testing purposes of the method.

Sections 1 and 2 outline the technical process and provide details on the equipment and techniques and about how the recording sessions were realized whereas section 3 presents a summary of the outcomes, both in terms of content produced and infrastructure. In particular, in section

1 (Motion Capture System), we describe the steps which we followed to organize the work-flow with the Motion Capture System, we describe the design of the Motion Capture sessions, including a short description of the technologies used, the locations, the context of capturing and quality control. In the third part we summarize the outcome of the Motion Capture sessions and we describe in details one of the dataset that became a dataset for experimental purposes for the proposed method of this work.

Chapter 7: In this chapter we introduce the technical development of software modules using Eyesweb and Matlab.

Here we describe the EyesWeb applications (Patches) developed to: (i) read and convert motion captured data from the full Marker set used during the recording to a simplified set of joints (i.e., matching the Kinect sdk joint set) and (ii) read Shapley analysis results and motion capture data from file and render them on a point-light display highlighting, frame by frame, the joint corresponding to the maximum Shapley value in order to generate stimuli for algorithm evaluation.

The software model in Matlab, responsible for graph creation, clustering and computation of Shapley values are presented in a separate section in this chapter.

Chapter 8: As a part of the Future Work contribution we present in this chapter the proposal for an evaluation platform, in order to perform a statistical study based on the opinions of the professional dancers. The study will take place in the final stage of the H2020-ICT-2015 EU Project WhoLoDance, Project no. 688865.

Chapter 9: Finally in the last chapter we conclude the dissertation.

Chapter 2

Preliminaries on Graph and Game Theory

Contents

This chapter provides the theoretical background and related work to this thesis. It begins by reviewing the historical impact, definitions and theorems related to game theory that could be referred to throughout the thesis. The concept of Shapley value is introduced. The following section consists of the review of the useful notations related to graph theory which are used in the thesis. The Chapter is concluded by introducing the proposed approach of graph restricted games, in which graph and game theory are combined together.

2.1 What is Game Theory?

Game theory is a formal discipline, which was first established in 1944 by the economist Oskar Morgestern and the mathematician von Neumann after publication of ”Theory of Games and Economic Behavior”. However, first examples of using game theory concepts were already present in studies by the mathematicians Emile Borel and Antoine Cournot.

Game theory is a mathematical field which studies cooperation and conflict behaviours of agents, when the outcome of the actions of the agents are interdependent. Therefore, a *game* represents a conflict and cooperative interaction of agents, where the *agents* can be players, groups or individuals, firms, markets, stocks, consumers etc.

To study, understand and analyze the strategic interactions of the agents, game theory provides specific concepts in order to model and design such decision-making processes.

The methodology of game theory allows one to analyze problems with strategic choice and improve strategic decision making. In order to model a problem as game, one have to enumerate the players and the possible behaviours (strategies) they can follow.

2.1.1 Definitions of Games

In this section we define some concept related to game theory according to [66].

Game theory is the study of competitive or/and cooperative interaction behavior. The formal objects in game theory are: 1) a *game* - model of strategical situation and 2) the *players* - agents or groups of agents, which interact with each other. The interaction involves different strategic actions, based on the preferences of the players or availability to the player. These actions are influence the outcome of the whole game.

"A *coalitional (or cooperative)* game is a high-level description, specifying only what payoffs each potential group, or coalition, can obtain by the cooperation of its members. However, it is not explicit by which process the coalition forms. Cooperative game theory investigates such coalitional games with respect to the relative amounts of power held by various players, or how a successful coalition should divide its proceeds" [66].

Coalitional (or cooperative) games are often used in the analysis of players behavior, in both politics and international relationship.

As an example, we can report the concept of the Nash bargaining solution, an important concept in economics, where players bargain over an outcome. Players have bargaining options that give some level of utility and also "fall back" options, in case if bargaining does not result in agreement. The solution depends exactly on the relative power of the two agents that bargain. Nash predicted the outcome of bargaining will be the outcome that maximizes the product of the gains. Nash modeled the solution within the cooperative interaction, which does not focuses on specific offers and counteroffers, but rather focuses mostly on the outcome of the bargaining process.

Noncooperative game theory, in contrast with cooperative games, is concerned about the competition of individual players or groups of players (formed coalitions) in which self-enforcing alliances are possible, because of the absence of enforcement by external means. Noncooperative games deal with the prediction of strategies and payoffs of the players in order to find a solution - Nash equilibria [52]. This concept of games does not analyze the strategic interaction that occurs within each coalition and the distribution of outcomes between players of a united coalition, it does not try to predict the joint action that a group of players may take and the resulting collective payoffs.

Agents of noncooperative games make a decision based only on their own interest of maximum outcome, that is why this game concept is named ”noncooperative”. However, the cooperation between players can occur anyway, if an agent consider this cooperation on his own best interests.

Apart from the number of players, Nash’s noncooperative model of bargaining process has the following specific features: the available actions that each player can perform and constrains that may occur, the information available to make an offer at a given time and the objective function that each player is trying to maximize/minimize.

Players in a game can be different according to specific assumptions. A *rational* player (major assumption in many variety of games) is one who always chooses an action which gives an outcome he most prefers, given what he expects his opponents to do. The main objective of a game with rational players is to make a prediction about how those rational players will play this game, in another words the main goal is to provide an insight on how to play the game against other rational players with the best outcome. More details about rationality assumption can be found in the book [32], which focuses on noncooperative game theory with rational players. This work in experimental economics uses a ”descriptive” approach of game theory, providing an important baseline case in economic theory, with the aim to provide good advice to the decision-maker.

2.1.2 Cooperative Games

In this section we present in details the notation used for cooperative games. The concept of cooperative games is used in developing the proposed approach in the framework of the present Ph.D thesis.

The definitions and notations in this section are stated according to [19].

We will start from the most basic mathematical model that describes a cooperative game. The game is a model of players interaction (competition, cooperation, bargaining etc.).

Agents - the players of the game: non-empty set $N = \{1, ..., n\}$.
A coalition is simply a subset of the players N: C, C', C^1, C_1... are symbols used to denote coalitions.
The grand coalition is the set N of all players.

A game G is given by a pair (N, v), where $N = \{1, ..., n\}$ is a finite, non-empty set of agents and $v : 2^N \rightarrow \mathbb{R}$ is a characteristic function, which maps each coalition $C \subseteq N$ to a real number $v(C)$. The number $v(C)$ is referred to as the value of the coalition C. The characteristic function is a "generalized" solution, that can tell what each player can expect to get he plays in a optimal way. This concept was first introduced by mathematician John von Neumann in 1928 [69].

The main goal in cooperative game theory is how to divide the coalitional value among the agents. However, the characteristic function of a game model does not show how coalitional value $v(C)$ should be divided among the agents. One of the solution concept of cooperative game theory is the *Shapley value* that is introduced in the following section.

Transferable utility games (TU games) are games characterized by the implicit assumption that the coalitional value in characteristic function games is that the coalitional value $v(C)$ can be divided among the members of C in any way that the members of C choose.

The empty coalition is $\emptyset$ with a coalition value equal to 0;
The value of each coalition can be non-negative or non-positive: in the first case, players form coalitions in order to make a profit; in a second case, agents form coalitions in order to share a cost.

One of the main aims of the cooperative game theory is to understand what will be the outcome of the game. In this concept an outcome consists in the following:

1. a partition of players into coalitions, called a coalition structure; and
2. a payoff vector, which distributes the value of each coalition among its members.

Given a characteristic function game $G = (N, v)$, a coalition structure over N is a collection of non-empty subsets $CS = \{C^1, ..., C^k\}$ such that

- $\bigcup_{j=1}^{k} C^j = N$, and
- $C^j \cap C^j = \emptyset$ for any $i, j \in 1, ..., k$ such that $i \neq j$.

The payoff vector is a vector $x = (x_1, ..., x_n) \in \Re^n$ for a coalition structure $CS = \{C_1, ..., C_k\}$ over $N = \{1, ..., n\}$ if

- $x_i \geq 0$ for all $i \in N$, and
- $\prod_{i \in C^j} x_i \leq v(C^j)$ for any $j \in 1, ..., k$.

An outcome of G is a pair (CS, x), where CS is a coalition structure over G and x is a payoff vector for CS. The total payoff of a coalition $C \subseteq N$ is defined as $x(C)$ = $\sum_{i \in C} x_i$. .

2.1.3 Solution Concept: The Shapley Value

The Shapley value is the most prominent way of dividing up some value among the members of the transferable utility game. The basic idea of the Shapley value concept is that members should receive payments or shares proportional to their average margin contribution.
The outcome provided by the characteristic function can differ: for example, one can get a payoff vector that allocates the entire payoff to only one player, when all the agents contributed equally to the value of coalition, can be considered less preferable. Instead, a payoff vector should allocate the payoff among of all players in a "fair" way.

One approach is identify axioms that express properties of a "fair" payoff division. The following sets of criteria are introduced:

1. fairness - how well each agent's payoff reflects his contribution
2. stability - what are the incentives for the agents to stay in the coalition structure

.

The Shapley value solution can be applied to a wide range of problems, starting from bargaining over division of a cost, finishing with power problems.

The solution concept we consider in the thesis aims to capture the notion of fairness in transferable utility games. It is known as the Shapley value after Lloyd S. Shapley [55].

The Shapley value is a solution concept that is usually formulated with respect to the grand coalition: it defines a way of distributing the value $v(N)$ that is obtained by the grand coalition.

The Shapley value is based on the intuition that the payment that each agent receives should be proportional to his contribution. A naive implementation of this idea would be to pay each agent according to how much he increases the value of the coalition of all other players when he joins it, i.e., setting the payoff of the player i to $v(N) - v(N_i)$. However, under this payoff scheme the total payoff assigned to the agents may differ from the value of the grand coalition.

To formally define the Shapley value, we need some additional notation. Fix a transferable utility game $G = (N, v)$. Let Π_N denote the set of all permutations of N, i.e., one-to-one mappings from N to itself. Given a permutation $\pi \in \pi_N$, we denote by $S_\pi(i)$ the set of all predecessors of i in π, i.e., we set $S_\pi(i) = j \in N|\pi(j) < \pi(i)$. For example, if $N = \{1, 2, 3\}$ then

$$\Pi_N = \{(1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1), (3, 1, 2), (3, 2, 1)\}$$

Moreover, if

$$\pi = (3, 1, 2)$$

then

$$S_\pi(3) = \emptyset$$
$$S_\pi(1)) = 3$$
$$S_\pi(2) = 1, 3.$$

The marginal contribution of an agent i with respect to a permutation π in a game $G = (N, v)$ is denoted by $\Delta_\pi^G(i)$ and is given by

$$\Delta_\pi^G(i) = v(S_\pi(i) \cup i) - v(S_\pi(i))$$

This quantity measures how much i increases the value of the coalition consisting of its predecessors in π when he joins them.

Given a transferable utility game $G = (N, v)$ with $|N| = n$, the **Shapley value** of a player $i \in N$ is denoted by $\phi_i(G)$ and is given by

$$\phi_i(G) = \frac{1}{n!} \sum_{\pi \in \Pi_N} \Delta_\pi^G(i)$$

.

Therefore, we can understand the Shapley value of a player i: as the average marginal contribution of that player, where the average is taken over all permutations of N.

The Shapley value has some properties, which are reported in the following propositions considered valuable and introduced bellow with propositions.

First, it is *efficient* . This means that it distributes the whole value of the grand coalition among all agents.

Proposition 1. For any characteristic function game $G = (N, v)$ we have $\sum_{i=1}^{n} \phi_i(G) = v(N)$.

If the player does not contribute to any coalition, then the solution concept Shapley value does not give any payoff to this player.
With a transferable utility game $G = (N, v)$, a player $i \in N$ is said to be a *dummy* if $v(C) = v(C \cup i)$ for any $C \subseteq N$. The Shapley value of a dummy player is 0.

Proposition 2. Consider a transferable utility game $G = (N, v)$. If a player $i \in N$ is a dummy in G, then $\phi_i(G) = 0$.

Another useful property of the Shapley value is that symmetric players have equal Shapley values. Symmetric players are those players i and j for which $v(C \cup i) = v(C \cup j)$ for any coalition $C \subseteq N \setminus \{i, j\}$.

Proposition 3. Consider a transferable utility game $G = (N, v)$. If players i and j are symmetric in G, then $\phi_i(G) = \phi_j(G)$.

Finally the additive property of the Shapley value can be introduced, when the same group of players N is involved: if group of players N that is involved in two coalitional games G^1 and G^2, i.e., $G^1 = (N, v^1), G^2 = (N, v^2)$. The *sum* of G^1 and G^2 is the coalitional game $G^+ = G^1 + G^2$ given by $G^+ = (N, v^+)$, where for any coalition $C \subseteq N$ we have $v^+(C) = v^1(C) + v^2(C)$. Then the Shapley value of a player i in G^+ is the sum of his Shapley values in G^1 and G^2.

Proposition 4. Consider two transferable utility games $G^1 = (N, v^1)$ and $G^2 = (N, v^2)$ over the same set of players N. Then for any player $i \in N$ we have $\phi_i(G^1 + G^2) = \phi_i(G^1) + \phi_i(G^2)$.

To summarize, the Shapley value possesses four desirable properties:

1. Efficiency: $i \in N\phi_i(G) = v(N)$;
2. Dummy player: if i is a dummy, then $\phi(G) = 0$;
3. Symmetry: if i and j are symmetric in G, then $\phi_i(G) = \phi_j(G)$;
4. Additivity: $\phi_i(G^1 + G^2) = \phi_i(G^1) + \phi_i(G^2)$ for all $i \in N$.

The first advantage of these properties is that they can be used in order to simplify the mathematical computation of the Shapley value. Secondly, these properties can be considered as axioms. Indeed, the Shapley value concept is a solution, payoff distribution that satisfies those axioms, and vice versa if we find the payoff that satisfy axioms introduced above, then we find Shapley value.

2.2 What is Graph Theory?

Graphs provide a natural way to model connections between different objects. They are very useful in depicting communication networks, social networks and many other kinds of networks.

Graph theory is a branch of mathematics concerned about how networks can be encoded and their properties measured. It has been enriched in the last decades by growing influences from studies on social and complex networks.

Graph theory can be defined as the study of graphs: graphs are mathematical structures used to model pair-wise relations between objects from a certain collection. Graph can be defined as set V of vertices and a set of edges. Here, V is collection of $|V| = n$ abstract data types. Vertices can be any abstract data types and can be presented by points in the plane. These abstract data types are also called nodes. A line (line segment) connecting these nodes is called an edge. Again,

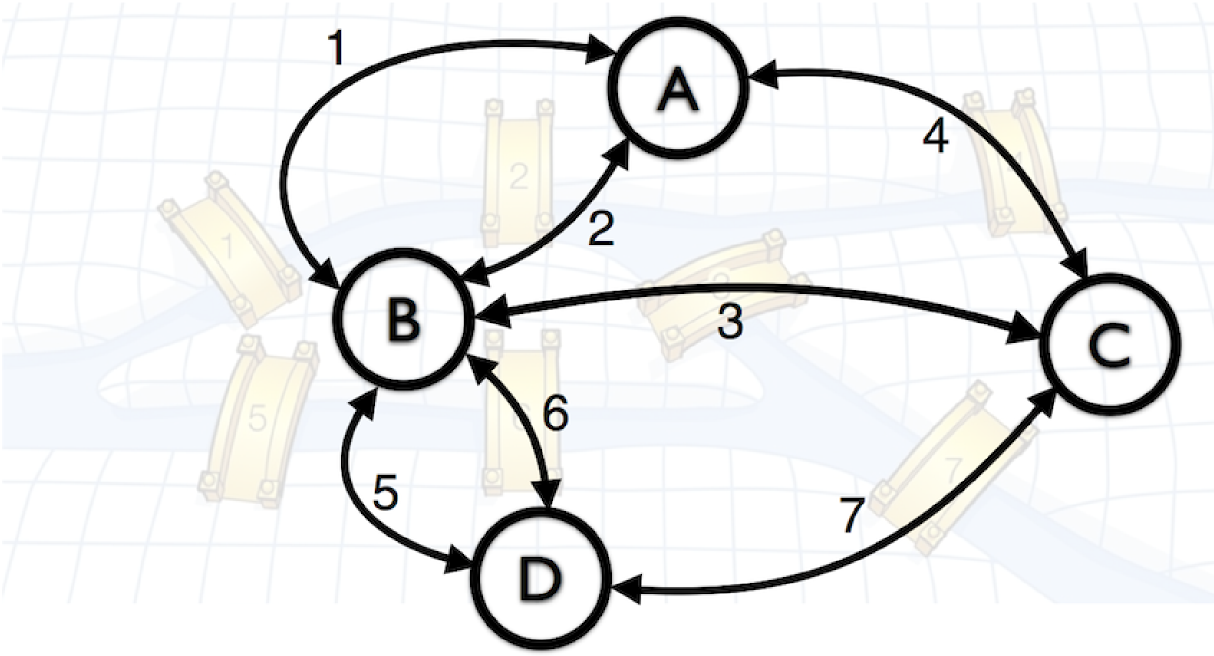

Figure 2.1: Königsberg bridge problem

more abstractly saying, an edge can be an abstract data type that shows a relation between the nodes (which again can be abstract data types).

In this section we introduce how the graph theory was developed and began as a mathematical branch that is able to solve many theoretical and real-world problems.

Leonhard Paul Euler (1707- 1783) was a pioneering Swiss mathematician. Euler was a founder of graph theory, as he solved for the first time a problem in graph theory, which is described in the following.

Problem

The 'Königsberg bridge' problem originated in the town of Königsberg, located on the Pregel river. Seven bridges connected the main territory of the town with 2 islands. The problem was formulated in this way: does it is possible to walk over all the bridges once and only once? The position of the seven bridges is reported in the picture 2.1 [1]

Euler solved that seven bridges of this town can not be crossed only once. He solve this problem using concepts of graph theory. He graphically illustrate a picture where lines (bridges) connected pieces of land, that where pictured as dots. In this way, the problem became better

[1] http://semanticommunity.info/Data_Science/Graph_Databases/Tutorial#Slide_6_Konigsberg_(Prussia)_173.

understandable and simplified by eliminating information that was not useful in particular case. In result the final picture could look similar to nodes and edges shown on 2.1.

Euler not only proved that that it is not possible to cross these bridges exactly once, but also introduced a new concept, one of fundamental importance of graph theory: the degree of the nodes. Degree of nodes is a number of edges connected to a given vertex.

To explain the solution of this problem, Euler proved the following theorem.
Theorem
Any given graph can be travelled with each edge travelled only once if and only if the graph has zero nodes with odd degree or exactly two nodes with odd degree.

In the case of the Königsberg bridge problem, the graph has 4 nodes, connected by two edges between each node, therefore each node has an odd degree. In this way Euler explained that it is not possible to travel the town crossing each bridge only once.

2.2.1 Definitions of Graphs

We are going to use standard graph-theoretical notation. For more details, one can see book of Berge [7] or the more recent book by Diestel [24]. The following definitions are taken from [7]:

A *graph* $G := (V, E)$ consists of a set V of *vertices* (or *nodes*) and a set $E \subseteq V \times V$ of *edges*. For any vertex $v \in V$, the vertices that are connected by an edge to v are called its *neighbours*. Let $V' \subseteq V$ be a subset of vertices and $E(V') \subseteq E$ denote the set of all the edges between them in G. The *subgraph induced by* V' is the pair $(V', E(V'))$. A graph is *connected* iff its edges form a path between any two vertices.

The set of all connected induced subgraphs of G is denoted by $S(G)$. In a connected graph (V, E), the vertex $v \in V$ is a *cut vertex* iff its removal splits the graph, i.e., if $(V \setminus v, E(V \setminus v))$ is not connected. If the subgraph induced by V' is not connected, then it consists of several *connected components* K_i, $i = 1, \ldots, m$. The set of such connected components is denoted as $K(V') := \{K_1, K_2, \ldots, K_m\}$.

For any vertex $v \in V$, we denote by $N(v)$ the set of *neighbours* of v in the graph, i.e., the vertices that are connected by an edge to v.

A graph with the vertex set V is said to be on graph *on* V. The vertex set of a graph G is referred to as $V(G)$, its edge set as $E(G)$. These conventions are independent of any actual names of these two sets: the vertex set W of a graph $H = (W, F)$ is still referred to as $V(H)$, not as $W(H)$. We shall not always distinguish strictly between a graph and its vertex or edge set. For example we may speak of a vertex $v \in G$ (rather than $v \in V(G)$, an edge $e \in G$, and so on.

The number of vertices of a graph G is its *order*, written as $|G|$; its number of edges is denoted by $||G||$. Graphs can be *finite, infinite, countable* and so on according to their order.

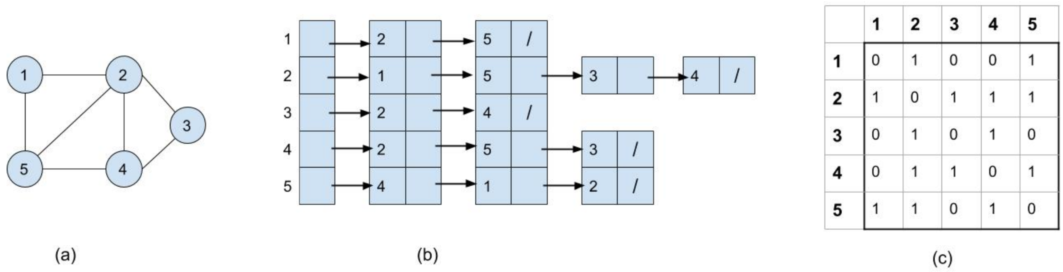

Figure 2.2: Two representations of an undirected graph [20]. (a) An undirected graph G having five vertices and seven edges. (b) An adjacency-list representation of G. (c) The adjacency-matrix representation of G.

A vertex v is *incident* with an edge e if $v \in e$, i.e. if e is an edge at v. The two vertices incident with an edge are its *endvertices* or *ends*, and that edge *joints* its ends. An edge $\{x, y\}$ is usually written as xy or (xy).

If $x \in X$ and $y \in Y$, then xy is an $X - Y$ *edge*. The set of all $X - Y$ *edges* in a set E is denoted by $E(X, Y)$. The set of all the edges in E at a vertex v is donated by $E(v)$.

2.2.2 Representations of Graphs

The following definitions and notations are taken from [20]:

There are two standard ways to represent a graph $G = (V, E)$: as a collection of adjacency lists or as an adjacency matrix. Either way is applicable to both directed and undirected graphs. The adjacency-list representation is usually preferred, because it provides a compact way to represent sparse graphs—those for which $|E|$ is much less than $|V|^2$.

An **adjacency-list representation** of a graph $G = (V, E)$ consists of an array Adj of $|V|$ lists, one for each vertex in V. For each $u \in V$, the adjacency list $Adj[u]$ contains all the vertices v such that there is an edge $(u, v) \in E$. That is, $Adj[u]$ consists of all the vertices adjacent to u in G. (Alternatively, it may contain pointers to these vertices.) The vertices in each adjacency list are typically stored in an arbitrary order.

An example is give on the figures: figure 2.2 is an adjacency-list representation of the undirected graph in Figure 2.2(a). Similarly, Figure 2.3(b) is an adjacency-list representation of the directed graph in Figure 2.3(a).

If G is a directed graph, the sum of the lengths of all the adjacency lists is $|E|$, since an edge of the form (u, v) is represented by having v appear in $Adj[u]$. If G is an undirected graph, the sum of the lengths of all the adjacency lists is $2|E|$, since if (u, v) is an undirected edge, then u

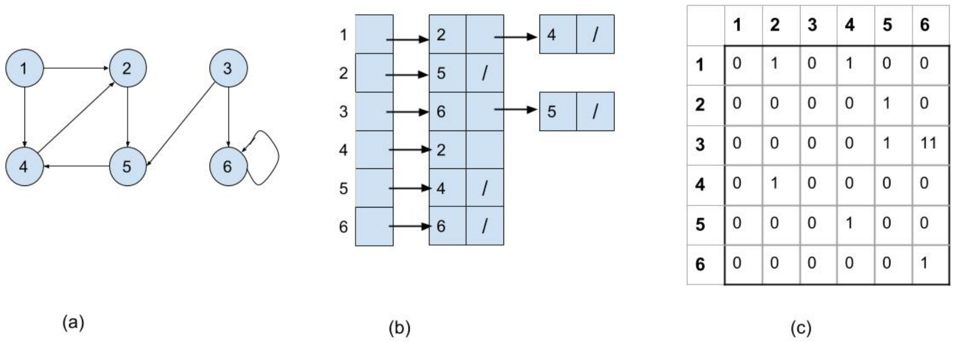

Figure 2.3: Two representations of a directed graph [20]. (a) A directed graph G having six vertices and eight edges. (b) An adjacency-list representation of G. (c) The adjacency-matrix representation of G.

appears in v's adjacency list and vice versa u' adjacency list.

For both directed and undirected graphs, the adjacency-list representation has the desirable property that the amount of memory it requires is $\Theta(V + E)$.

Adjacency lists can readily be adapted to represent weighted graphs, that is, graphs for which each edge has an associated weight, given by a weight function $w : E \to \mathbb{R}$. For example, let $G = (V, E)$ be a weighted graph with weight function w. The weight $w(u, v)$ of the edge $(u, v) \in E$ is simply stored together with vertex v in u's adjacency list. The adjacency-list representation is quite robust in that it can be modified to support many other graph variants. A potential disadvantage of the adjacency-list representation is that there is no quicker way to determine if a given edge (u, v) is present in the graph than to search for v in the adjacency list $Adj[u]$.

This disadvantage can be remedied by an adjacency-matrix representation of the graph, at the cost of using asymptotically more memory.

For the adjacency-matrix representation of a graph $G = (V, E)$, we assume that the vertices are numbered $1, 2, ..., |V|$ in some arbitrary manner. Then the adjacency-matrix representation of a graph G consists of a $|V| * |V|$ matrix $A = (a_{i,j})$ such that

$$a_{i,j} = \begin{cases} 1\, if(i, j) \in E \\ 0\, otherwise \end{cases}$$

Figures 2.2(c) and 2.3(c) represents the adjacency matrices of the undirected and directed graphs in Figures 2.2(a) and 2.3(a), respectively. The adjacency matrix of a graph requires $\Theta(V^2)$ memory, independent of the number of edges in the graph.

Observing the symmetry along the main diagonal of the adjacency matrix in Figure 2.2(c), we define the **transpose** of a matrix $A = a_{i,j}$ to be the matrix

$A^T = (a^T_{i,j})$ given by $a^T_{i,j} = a_{i,j}$.

Since in an undirected graph, (u, v) and (v, u) represent the same edge, the adjacency matrix A of an undirected graph is its own transpose: $A = A^T$. In some applications, it pays to store only the entries on and above the main diagonal of the adjacency matrix, thereby cutting the memory needed to store the graph almost in one half.

Like the adjacency-list representation of a graph, the adjacency-matrix representation can be used for weighted graphs. For example, if $G = (V, E)$ is a weighted graph with an edge-weight function w, the weight $w(u, v)$ of the edge $(u, v) \in E$ is simply stored as the entry in row u and column v of the adjacency matrix. If an edge does not exist, a NIL value can be stored as its corresponding matrix entry, though for many problems it is convenient to use a value such as 0 or ∞.

Chapter 3

State of the Art

Contents

As one of the most active research areas in computer vision, human movement analysis has received a big attention in recent years [3]. Novel generations of human-computer interfaces integrating computational models of social affective communication are receiving a growing interest as well [46].

3.1 Movement Qualities

Movement qualities are among the fundamental properties of human movement. They result from the movement dynamics, i.e., its temporal characteristics, and can be defined independently of specific spatial trajectories, forms, or shapes. These properties of movement, widely used in dance practice, have been explored significantly in Human Computer Interaction (HCI). At the beginning, movement qualities like body expression were widely studied in humanistic theories and arts.

As an example, the dance theorist Rudolf Laban considered movement qualities as a central notion to investigate human motion [37]. Since then, Laban Movement Analysis (LMA) [36] has been applied in various other contexts such as early childhood development, sport, and rehabilitation. It was experimentally shown that movement qualities can communicate social relations and intentions, such as emotional states [17], affiliation [38], cultural background [51], agreement [12], group cohesion [30], dominance and empathy [67].

The connection between emotions and movement qualities was investigated in [70], defining the following categories that show significant differences for characterizing emotions: "movement activity" (overall "quantity of motion", i.e., a category related to energy), "expansiveness/spatial extension", and "movement dynamics/energy/power". The terminology employed therein refers to physical systems with their dynamics, energy, and power.

Several approaches dedicated to automatically detecting emotional states from body movement were recently developed. In [8], the authors proposed to detect emotional states from low-level features of hand movements, such as a maximum distance of the hands from the body, and their average speed and acceleration. In [18], expressive qualities of the movement such as the amount of motion, contraction and directness indexes, as well as velocity and acceleration were used to classify four emotions (anger, sadness, joy, and pleasure) with dynamic time-warping classifiers.

3.2 Applications of Graph Theory

3.2.1 Graph Theory in Science

Graph theory has made a big impact into different areas of science. The biggest use of graph theory refers to computer science, but nevertheless applications in physics, chemistry, biology, mathematics are increasing significantly recently. In this section we give a review of applications of graph theory in the literature.

Physics
Graph can represent atomic structures, topology of atoms, local connections between interacting parts of a system, model systems in quantum chaos, in the study of waveguides, in photonic crystals.

In [31] a graph was used to represent dynamics of a physical process.
In [28] quantum graphs were used as a model to study quantum chaos, the quantum mechanics of systems that are classically chaotic.

Chemistry
A graph can represent a molecule (where the nodes are atoms and the edges are bonds). Graph

theory can be used to model topology such as the mathematical study of isomerism.

In [14] reaction graphs were used to represent intramolecular rearrangements of octahedral complexes and degenerate rearrangement of carbonium ions.
In [4] the systematic enumeration of “diamond hydrocarbons” was introduced as a graph theory problem and solved by applying dual-graph approach.

Linguistics
Graph can be used to model semantic networks – ways of storing interconnected knowledge (where the nodes are concepts and the edges are connections between concepts).
A hierarchical graph can represent the syntax and semantics.

[62] considered rhyme graphs corresponding to rhyming corpora, where the nodes are words and the edges are observed rhymes.
In [44] graph theory was used to improve the translation from English to hindi.
[26] introduced the task of computing vector space representations for the meaning of word occurrences, which can vary widely according to the context.

Sociology
A graph can represent a real network of human beings or a social network (where the nodes are people and the edges are interactions between people). Edges can be relevant to model acquaintanceship, influence or friendship.

[54] investigated the “cohesion” in social networks, assuming that in a cohesive network the removal of a person (node) has little effect.
[22] argued the achievable security and privacy degree of an online social network strongly depends on the graph-theoretical properties of the social graph representing the real friendship relations between the users.

Biology
A graph can be used to model the population of bacteria and animals (where the nodes are regions and edges are migration paths of the animal species).
A graph can represent molecular structures and DNA.

[29] proposed an alignment-free technology for DNA sequence similarity analysis based on graph theory concepts and genetic codes.
In [39] the authors applied a network approach in order to model heterogeneous biological interactions. They developed a system called ”megNet” for visualizing heterogeneous biological data, and showed its utility for biological networks.

Mathematics
Graph theory is used for modelling several practical problems: data structures and algorithms for the exploration of graphs. Usually pairwise connection between nodes has a numerical value. For example in a travel network, a weighted edge can represent the length of a rout, or the load of the vehicle.

Mathematical applications of graph theory are common for several real-world problems. In [5] a classification is proposed for academic timetabling problems, mathematical models, solution methods and data representation based on graph theory.
In [1] an overview can be found about applications of graph theory to model techniques for frequency assignment problems in wireless communication.

Computer Science
Graph theory can be used in data mining, image segmentation, clustering, image capturing, networking etc.
Graphs can represent data organization, databases, computational connections between devices, connections between links of a website.

[73] investigated the problem of generating and selecting graph models that reflect the properties of real inter-networks.
In [40], the relationship between different entities was modelled by a graph embedded into a common Euclidean space. The entities introduced were spatio-temporal descriptors and spin-images, connected according their relationship.

3.2.2 Graph Theory in Movement Analysis

One of the most recent studies of graphs in movement analysis was presented in the article [33], therein, a graph-based approach was used for motion capture data representation and analysis.

In that work the authors [33] aimed at ”providing better representation methods for motion capture data in order to improve performance in terms of classification, recognition, synthesis and dimensionality reduction”. In their paper, a novel representation method was proposed inspired by algebraic and spectral graph theoretic concepts. They proposed a method to represent motion data in a space constructed with bases for skeleton-like graphs.

Authors tried to address the problem of dimensionality reduction for human Mocap data using a new linear approach, moving from a high-dimensional data space to a lower-dimensional feature space. They propose a new representation of motion capture data using spectral graph theory.

Human motion was taken into consideration, in order to construct human-like graphs.

To state the problem mathematically, the authors assumed a given motion sequence represented as a sequence of frames. Each frame was represented by specifying the three-dimensional coordinates for all the selected joints in the body. Each frame $M_i(i = 1, 2, ..., n)$ was represented as a 15-by-3 matrix because the coordinates of 15 joints were measured and collected. $M_{i,j}$, the j^{th} row of M_i, is the three-dimensional vector of coordinates of the j^{th} joint at the particular time i.

In that paper, the authors proposed a novel representation for motion capture data. They applied graph theory and treated the motion data as graph signals resided on the graph they construct. Authors claimed that ”reprocessing for the motion data can lead to a better input for the following processing steps such as dimensionality reduction and classification”.

Moreover many studies which are formulated as a graph theory problems are related to computer vision. In the field of human action recognition one can find examples such as the following. In [10], machine learning techniques were applied to the graph associated with each frame, based on interest points. Distances between a scene frame-graph and a set of prototypes were used together with Hidden Markov Models (HMMs) for classification. In [49], human body configuration was used in order to creat a graph, where the nodes correspond to the body parts and the edges to their energy function. In [65] author introduce generic energy-minimization problem as one of graph matching problems considered as an action recognition model. Action was represented as a model of graph, where nodes are geometric or appearance feature, connected by relationships.

3.3 Application of Game Theory

3.3.1 Game Theory in Science

Economics
Game theory is used to study economical behaviors applied to the markets, stocks, consumers and business.

In [2] examples can be found how game theory can be applied to economical behaviors like auctions, bargaining, mergers and acquisitions pricing . There are exist several branches of economics that use game theory: information economics [50], industrial organization [64], behavioral economics [25], and political economy [58, 57].

Political Science
Game theory is used to model relations in political economy such voting, social choice study, fair division, where players can be populations, states, parties.

In [13] one can find the evidence how game theoretical approaches were applied to the Cuban missile crisis, in the years of US presidency by John F. Kennedy.

Biology
According to Maynard Smith [60], in the preface to Evolution and the Theory of Games, ”paradoxically, it has turned out that game theory is more readily applied to biology than to the field of economic behavior for which it was originally designed”. Game theory is also applied to understand many relationships in biology. Evolutionary game theory is one of such branches.

Different types of communications between animals [61] are more understandable via game theory, for example:
in [60], fighting behavior and territoriality were modeled as a game of chicken;
in [61], evolutionary game theory was used to study mobbing behavior, when a large number of smaller animals attack a bigger predator in order to defense themselves. This phenomenon is called “spontaneous emergent organization”.

Computer Science
Game theory is applied in order to develop algorithms, multi - agent communication, robotics, cloud/distributed computing, network security, machine learning, social networks, and resource management.

[71] presented a game-theoretic method of fair resource allocation for cloud computing services. Game theory was used for modeling complex interactions between cloud providers and a number of service providers with conflicting objectives such as maximizing quality at the minimal cost. They investigated the Nash equilibrium for the proposed game-theoretical model, where no players has an incentive to change its strategy to model the relationship. There are many scenarios in cloud resource management involving spot pricing of cloud resource addressed by auction/bidding games.
In [6] K-server problem, that known as game with moving cost used in work with online algorithms and game-theoretic technique for proving lower bounds on the computational complexity of randomized online algorithms.

3.3.2 Game Theory in Movement and Behavioral Analysis

In the literature, we did not find a significant application of game theory in analysis of human movements.

Game theory has been very useful in the understanding of the behavior not only of animals, but also of people. Game theory models and the concept of evolutionarily stable strategy (ESS) provided sound explanations to a variety of phenomena that could not otherwise be fully understood. A game theoretic approach has to be used to understand the behavior of animals whenever there are reasons to believe that the strategy or the behavior of one organism is affected by the behavior of the other and vice versa [11].

The authors of [53] observed smiling as a signal of the intention to cooperate in social interaction. However, given that humans have the ability to smile falsely, the ability to detect intentions may go far beyond the ability to recognize a smile. In their study, they examined the value of a smile in a simple bargaining context. 120 subjects participate in a laboratory experiment consisting of a simple two-person, one-shot "trust" game with monetary payoffs. Each subject was shown a photograph of his partner prior to the game; the photograph was taken from a collection that included one smiling and one non-smiling image for each among 60 individuals. These photographs are also rated by a separate set of subjects who completed a semantic differential survey on affective and behavioral interpretations of the images.

Results lent some support to the prediction that smiles can elicit cooperation among strangers in a one-shot interaction. Other characteristics of faces also appeared to elicit cooperation. Factor analysis of the survey data revealed an important factor, termed "cooperation", which is strongly related to trusting behavior in the game. This factor is correlated with smiling, but is somewhat more strongly predictive of behavior than a smile alone. In addition, males were found to be more cooperative, especially towards female images, whereas females are less cooperative towards female images [53].

Other works used game theory in order to study the interaction between humans. Early engineering models which were used to predict people's movement, such as EVACNET [27], applied no behavioral rules. They relied on the physical movement of the population, and the physical representation of the building geometry to influence and determine occupant egress. Recently, social/architectural scientists, such as Passini [45], Ozel [43], Proulx [47, 48], Sime [59] and Canter [16] have pointed out that one of the dominant factors affecting evacuation patterns is the evacuees' behavioral reaction accompanying their movement. Their studies have identified the contributing factors and provide valuable information for studying wayfinding process. On the basis of these studies, some engineering models now incorporate some psychological rules to model the response pattern of evacuees. However, how the rules can be applied in a dynamic process, especially when the reaction of an individual is affected dynamically by others, has rarely been discussed.

Game theory proved to be useful in understanding the behaviour and movements of human in various situations. Nevertheless, according to the authors', the application of game theory is novel. Therefore, game theory applied to analysis of human movement is a novel and interesting technique to explore.

3.4 Combining Graph and Game Theory

Despite the fact that the combination of graph and game theory techniques is already used in several research fields, to the best of the authors' knowledge, approaches aimed at the analysis of expressive gestures through a combination of these methods have not yet been developed.

Graph-restricted games, first introduced by Myerson [42], model naturally-occurring scenarios, the coordination between any two agents within a coalition of players is only possible if there is a communication channel (a path) between them. Recent research studies provide several examples of combinations of graph and game theory techniques. For instance, in [21] these two approaches were combined to investigate how game-like interactions influence the cellular topology of a planar tissue. In network economics, these tools were used to study decentralized networks of agents. An example can be found in [34], which deals with a general model of exchange in decentralized markets.

Furthermore, a lot of the work in evolutionary game theory came down to considering how large populations of myopic agents play over time. Many of these models rely on limiting the history of remembered play by agents to some finite number of "records", based on graph-theoretic arguments [72]. Moreover, there is a branch of game theory that explicitly deals with games that have a sort of graphical structure (network games). For instance, in [35] the relation between pure-strategy Nash equilibria in network games of anti-coordination and graph coloring problems was discussed.

Part II

Proposed Approach

Chapter 4

Problem Formulation and Implementation

Contents

4.1 The basic Idea

Analyzing a problem through game theory means basically being able to associate to it a scheme that could describe completely its characteristics using the features of a game.

Starting from the literature from experimental psychology and Human Computer Interaction (HCI) [70], [41], [9], [15], humanistic theories and the arts [37], [36], and from meetings with dance experts, a collection of expressive movement qualities was taken into consideration to be considered, including smooth, light/heavy, fluid, impulsive, sudden/sustained, symmetric, contracted/expanded, energetic, synchronized movements. To describe such movements, we used, as an initial implementation, a single low-level feature, namely ”energy” (i.e., the energy that a cloud of points possesses due to its motion).

Our method is based on the idea that “important” joints during a specific movement are those that separate parts of the body characterized by different motion behaviors. For instance, if an arm is moving and the corresponding shoulder and the other parts of the body are at rest, then that shoulder may be considered as an “important” joint because, in a certain sense, though being at rest, it “controls” the motion of the arm.

In summary, our method tends to consider as the most important vertex one that connects different clusters (here identified by spectral clustering), where each cluster corresponds to connected joints with similar movement qualities.

We expect the method to be relevant for the analysis of movement qualities, because, differently from other approaches:

1. its game-theoretical component makes it possible to identify the most important joint;
2. its graph-theoretical component allows one to take into account the skeletal structure of the body in the analysis, and also the possible presence of "bridge" edges.

4.2 Implementation

1. To implement the ideas above, we first define a graph G representing the skeletal structure of the body, having its joints as vertices. The edges belonging to this skeletal structure are called *physical* edges.

   For each frame of a recording session, the similarity in the current values of a specific motion-related feature is used to assign positive weights to these physical edges. In more detail, the edge weight is inversely proportional to the sum of a small positive constant (to prevent division by 0) and the absolute value of the difference of the feature values associated with the two vertices joined by the edge.

   Additional *nonphysical* edges (or bridge edges) are inserted between vertices not directly connected by physical edges. Positive weights are assigned to these nonphysical edges, too, which are proportional to the current similarity in feature values of the associated vertices, with a constant of proportionality chosen to be much smaller than for the physical edges (5 times smaller, in the present implementation of the method).

   Indeed, compared with physical edges, nonphysical edges should have a large weight only in case of a very large similarity of feature values.

2. Then, we cluster the resulting weighted graph applying spectral clustering to its weighted edge set (see, e.g., [63] for some details on this clustering method, and [68] for the specific implementation considered in the paper). In such a way, vertices that belong to the same cluster are expected to have similar feature values, whereas edges between different clusters should be associated with vertices having significantly different feature values.

3. At this point, for each frame we construct an auxiliary graph, whose vertices are the same as the ones of the original graph, and whose edges form the subset of the physical edges of the original graph that connect joints belonging to different clusters (i.e., in this phase, bridge edges are not considered any more).

Then, we attribute weights to these physical edges, which are proportional to the dissimilarity of the feature values of the associated joints (in the specific case, they are equal to the absolute value of the difference of the feature values associated with the two joints).

The weighted auxiliary graph is denoted by $G^{aux} := (V, E^{aux}, w^{aux})$. The set of neighbours of a vertex v in the weighted auxiliary graph is denoted by $N^{aux}(v)$.

4. As a further step, we construct a cooperative TU game on the weighted auxiliary graph.

   Its characteristic function is defined as follows: for each coalition V', the value $c(V')$ is defined as the sum of the weigths (in the weighted auxiliary graph G^{aux}) of all the physical edges contained in the subgraph induced by V'. In other words, we set

$$c(V') = \sum_{v,\hat{v} \in V', \hat{v} \in N^{aux}(v)} w^{aux}(e_{v,\hat{v}^{aux}}) \tag{4.1}$$

5. For the above-defined cooperative transferable utility game, we compute its Shapley value, and we use it to rank the joints. The "most important" joint in each specific frame is defined as the one with the largest rank, if there is only one joint with that property. In case of more than one joint with the largest rank, the one with the smallest index (according to an a-priori given labeling of the joints) is considered (a random selection is not used, to avoid introducing noise in the model).

   In a similar way, as a by-product of the procedure, one can compute also the joint with the second-largest Shapley value (again, the vertex with the smallest index is considered in case of multiple choices), because it could occur that the two vertices with the two largest Shapley values are connected by an edge in the weighted auxiliary graph (in that case, that edge could be considered as the "most important" edge in the graph). This would occur, e.g., if the spectral clustering step produced only two clusters, separated by that edge.

6. Finally, a filtering step is applied to the computed Shapley values, in such a way to keep only the vertices that were automatically evaluated to be the most important ones for some number of consecutive frames (51, in the current implementation of the method).

Chapter 5

Theoretical Analysis

Contents

5.1 Monotonocity, Superadditivity, Convexity

In this section, we discuss some theoretical properties of the cooperative TU game on the weighted auxiliary graph presented in Chapter 4.

First, it can be proved (see, e.g., [23]) that, for our choice of the characteristic function c and for any player i ($i = 1, \ldots, |V|$) associated with the vertex $v_i \in V$ in the weighted auxiliary graph, one has

$$\phi_i(c) := \sum_{\hat{v} \in N^{aux}(v_i)} w^{aux}(e^{aux}_{v_i,\hat{v}}) \,. \tag{5.1}$$

In other words, our approach of ranking the joints based on the Shapley value of the cooperative TU game with the characteristic function (4.1) is equivalent to ranking them according to their *weighted degree centrality* on the weighted auxiliary graph.

We recall that a characteristic function c of a cooperative TU game (in this chapter we denote it by c, to avoid making confusion with the symbol v, used here to denote a vertex) is *monotonic* if and only if, for every two subsets $V_1 \subseteq V_2 \subseteq V$, one has $c(V_2) \geq c(V_1)$.

It is *superadditive* if and only if, for every two disjoint subsets $V_1 \subseteq V_2 \subseteq V$, one has $c(V_1 \cup V_2) \geq c(V_1) + c(V_2)$. Monotonocity and superadditivity are quite common and reasonable properties for the characteristic function of a cooperative TU game. Indeed, monotonicity implies that, among all possible coalitions, the grand coalition is the one with the maximum utility, whereas superadditivity guarantees that the Shapley value is *individually rational*, i.e., that the Shapley value of each player is larger than or equal to the value of the coalition made only of that player (hence, the player has interest to join the grand coalition). Moreover, a cooperative TU game is *convex* if and only if, for every two subsets $V_1 \subseteq V_2 \subseteq V$ and every $v \in V \setminus V_2$, one has $c(V_1 \cup \{v\}) - c(V_1) \leq c(V_2 \cup \{v\}) - c(V_2)$. Convexity of a cooperative TU game implies nonemptiness of the *core* (which is a more general solution concept for a cooperative TU game, see [63] for its definition and for related theoretical properties), and in particular, the belonging of the Shapley value to the core.

5.2 Properties of the Characteristic Function of the Game

The following proposition with their proofs, shows some properties of the characteristic function c of the cooperative TU game on the weighted auxiliary graph, provided by formula (4.1), and of the cooperative TU game itself.

Proposition 1. For the cooperative TU game whose characteristic function c is defined by formula (4.1), the following properties hold:

- c is monotonic;
- c is superadditive;
- the cooperative TU game is convex.

Proof of Proposition 1

(i) This follows from formula (4.1) and the fact that, for every two subsets $V_1 \subseteq V_2 \subseteq V$, one has $E^{aux}(V_1) \subseteq E^{aux}(V_2)$, and the weights of the edges of the weighted auxiliary graph G^{aux} are non-negative.

(ii) For every two disjoint subsets $V_1 \subseteq V_2 \subseteq V$, one has

$$
\begin{aligned}
& c(V_1 \cup V_2) \\
= & \sum_{v,\hat{v} \in V_1 \cup V_2, \hat{v} \in N^{aux}(v)} w^{aux}(e^{aux}_{v,\hat{v}}) \\
= & \sum_{v,\hat{v} \in V_1, \hat{v} \in N^{aux}(v)} w^{aux}(e^{aux}_{v,\hat{v}}) \\
& + \sum_{v,\hat{v} \in V_2, \hat{v} \in N^{aux}(v)} w^{aux}(e^{aux}_{v,\hat{v}}) \\
& + \sum_{v \in V_1, \hat{v} \in V_2, \hat{v} \in N^{aux}(v)} w^{aux}(e^{aux}_{v,\hat{v}}) \\
& + \sum_{v \in V_2, \hat{v} \in V_1, \hat{v} \in N^{aux}(v)} w^{aux}(e^{aux}_{v,\hat{v}}) \\
= & \; c(V_1) + c(V_2) \\
& + \sum_{v \in V_1, \hat{v} \in V_2, \hat{v} \in N^{aux}(v)} w^{aux}(e^{aux}_{v,\hat{v}}) \\
& + \sum_{v \in V_2, \hat{v} \in V_1, \hat{v} \in N^{aux}(v)} w^{aux}(e^{aux}_{v,\hat{v}}) \\
\geq & \; c(V_1) + c(V_2) \,,
\end{aligned} \tag{5.2}
$$

due to the non-negativeness of the weights of the edges of the weighted auxiliary graph G^{aux}.

(iii) Let $V_1 \subseteq V_2 \subseteq V$ and $v \in V \setminus V_2$. One has

$$
\begin{aligned}
& c(V_1 \cup \{v\}) - c(V_1) \\
= & \; 2 \sum_{\hat{v} \in V_1, \hat{v} \in N^{aux}(v)} w^{aux}(e^{aux}_{v,\hat{v}}) \\
\leq & \; 2 \sum_{\hat{v} \in V_2, \hat{v} \in N^{aux}(v)} w^{aux}(e^{aux}_{v,\hat{v}}) \\
= & \; c(V_2 \cup \{v\}) - c(V_2) \,,
\end{aligned} \tag{5.3}
$$

since $V_1 \subseteq V_2$, and the insertion of the vertex v in V_1 (respectively, in V_2) causes an increase of the value of the characteristic function equal to two times the sum of the (non-negative) weights of the edges in the weighted auxiliary graph linking v with the vertices of V_1 (respectively, of V_2).

Part III

Experimental Setup and Software

Chapter 6

Dataset and Data Acquisition

Contents

6.1 Data Acquisition Objectives

The main purpose of this work is the development of a novel approach and the associated computational method for the analysis of expressive full-body movement qualities, combining tools and methods from graph theory and cooperative game theory. In order to have the possibility to exploit the proposed techniques, one of the first and most important aspects is the collection of the data that can be used in an experimental setup.

In order to study motion, one must develop the methodology and design how and what to capture in a first place. There is only very limited possibility to capture motion in an analogue way. Thus, differently from sound, image, video and other multimedia only digital processing of motion is applicable. The lack of an analogue equivalent for digitized motion makes processing such data unique and novel. However, motion digitalization opens a new research area with both theoretical and practical challenges.

The most suitable method for our purposes was considered to be the Motion Capture System data acquisition. In the following, the abbreviation MoCap data will be used to refer to data acquired using this system.

Motion digitalization is done via various motion capture devices which produce Motion Capture data. A universal format of MoCap data is not fixed, and several proprietary formats could be proposed (e.g., asf/amc, bvh, c3d, csv). Generally, MoCap data represent motion sampled according to some known sampling frequency (mostly constant during the acquisition process) in a form that allows reconstruction of each sampled pose.

Even though modern capturing devices can detect and track multiple persons in this thesis we consider MoCap data of single individual, i.e., this thesis does not discuss person interaction activities.

The following sections in this chapter present the methodology which was followed in order to design the Motion Capture Sessions, in a way that maximized the impact of the technologies used, to optimize the time management, select the movement sequences that we need to capture and store data, and build a repository of movement which covers the needs of the testing purposes of the method.

Sections 1 and 2 outline the technical process and provide details on the equipment and techniques used to record sessions, whereas section 3 presents a summary of the outcomes, both in terms of content produced and infrastructure.

In particular, in section 1 (Motion Capture System), we describe the steps which we have followed to organize the work-flow with the Motion Capture System, and we describe the process of the Motion Capture sessions, including a short description of the technologies used, the locations, and the context of capturing and quality control.

The third part summarizes the outcome of the Motion Capture sessions and describes in details one of the dataset that became a dataset for experimental purposes for proposed method of this work.

6.2 Data Acquisition

A multimodal system platform at Casa Paganini Infomus, Research Center of the Universita degli studi di Genova, based on the EyesWeb XMI research platform and Qalysis Motion Capture System, was used to allow synchronized recording and playback.

The platform supported several research activities such as: the creation of a multimodal repository of recordings; the fine-grain synchronization of multimodal data; the playback and testing of the repository; the real-time processing and application of the models for MoCap.

For the purpose of the multimodal recordings, several configurations of low-level input devices were tested. First, an overview and analysis of wearable sensors offered on the market was carried out, which allowed us to choose a set of the devices to be used in multimodal recordings.

The selected devices are: the Qualisys system (a high precision system with 13 cameras Oqus), 2 high Resolution video cameras, and several wearable IMU sensors: xOSC, and wireless microphones. Next, all the devices were integrated in the EyesWeb software platform. Particular attention was paid to the development of the techniques to support synchronized recordings of the multimodal data obtained from the different devices, characterized by different frame rates (e.g., video - variable framerate, Qualisys – 100Hz, IMUs – 25/50Hz). The synchronization details are described in the following section.

1. Eyesweb Platform

   The platform is based on EyesWeb XMI, allowing users to perform synchronized recording, playback, and analysis of a multimodal stream of data. EyesWeb XMI is a modular system that allows both expert (e.g., researchers in computer engineering) and non-expert users (e.g., artists) to create multimodal installations in a visual way. EyesWeb provides software modules, called blocks, that can be assembled intuitively (i.e., by operating only with mouse) to create tools and programs, called patches, that exploit system's resources such as multimodal files, webcams, sound cards, multiple displays and so on.

2. Oqus cameras by Qualisys

   Motion Capture Cameras Oqus. The main feature of the Oqus cameras is the ability to calculate marker positions with impressive accuracy and speed. Hundreds of markers can be measured at thousands of frames per second with ten, fifty or even more cameras – and all this can be run off an ordinary laptop. No workstation or switch is needed, which makes the system easy to move out in the field.

3. Inertial Moment Units: xOSC

   Inertial Movement Units (IMU) are devices endowed with sensors that capture, sample and transmit data such as acceleration, angular velocity and magnetic field in real-time.

   We used 4 x-OSC sensors. x-OSC is a wireless I/O board that provides just about any software with access to 32 high-performance analogue/digital channels and on-board sensors (gyroscope, accelerometer, magnetometer) via OSC messages over WiFi. There is no user programmable firmware and no software or drivers to install, making x-OSC immediately compatible with any WiFi-enabled platform. All internal settings can be adjusted using any web browser.

   Data is captured at 50 frames per second; each frame consists of 9 values: (x, y, z) of accelerometer, gyroscope, and magnetometer;

4. Microphone

We used a wireless microphone with radio frequency transmitter (Mono, 48 kHz), head-mounted. In the framework of this thesis we did not used the audio data, however it is available in the dataset and will be used in the future developments of the approach.

5. Video cameras

   We used 2 JVC video cameras from frontal and lateral side, for different point of view capture (1280x720, at 50fps).

6.2.1 Motion Capture System

the Motion Capture System available at premises of Casa Paganini Infomus consists of 13 high resolution and high presicision infrared cameras Oqus plus native integrated software Qualisys QTM, a Qualisys proprietary tracking software, which designed to work seamlessly with any model of Qualisys camera, ensuring fast and precise data collection. The system allows users to perform 2D, 3D and 6DOF capture of data in real-time, with minimal latency. QTM meets the needs of advanced as well as less experienced users in a range of applications – from medical sciences to industrial ergonomics. Flawless integration and synchronization with force plates and EMG devices as well as real-time streaming into 3rd party analysis software, make QTM a state-of-the art software. QTM is built and developed around a set of advanced motion capture algorithms ensuring high performance, accuracy and low latency.

The Mocap System Qualisys has a number of valuable technical features:

1. 2D and 3D Tracking

   When the data (x, y) from the cameras are collected in 2D, QTM calculates the 3D (x, y, z) positions. The Qualisys tracker is a real-time tracker that combines marker occlusion and merging detection techniques with an extreemely fast and highly accurate tracking algorithm.

2. 6 Degree of Freedom - 6DOF

   The 6DOF tracking function provides 6-degrees-of-freedom (pitch, roll, yaw, x, y, z) data from user-defined rigid bodies. The 6DOF data gives information about the position and rotation of a moving body. QTM can both save 6DOF data and send 6DOF data over UDP in real-time.

3. Automatic Identification of Markers - AIM

The AIM model is created from an identified file and can then be applied to any measurement that captures similar motions compared to the model, or just a part of the motions in the model. The AIM does not have requirements on human body model mapping for each frame in real-time. After the model has been identified in real-time, the tracker keeps tracking the model with frame rate independent performance.

4. Passive and Active Markers

   Qualisys offers the only system on the market that can measure passive or active markers and high-speed video, with the same software.

The data available after the recordings are a set of unidentified trajectories of the movements of markers. Therefore these data had to be labeled, cleaned from noise, and bones needed to be created, in order to have a convenient representation of the model.

6.2.1.1 Work-flow pipeline with the Motion Capture System

There are 8 main parts of work-flow with the MoCap system, that served as a guideline for all the scenarios considered in the thesis.

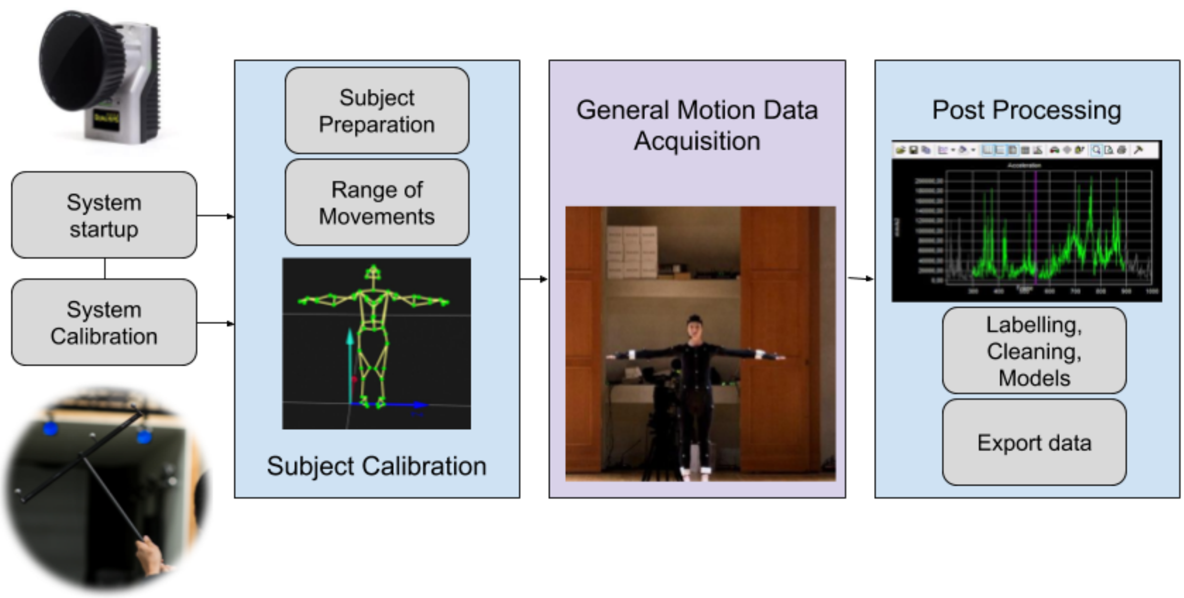

Figure 6.1: Motion Capture Workflow

1. System Startup

   First, we have to boot up the system and start all necessary components of the integrated platform. Due to specifics of the MoCap systems and settings, this step can be different in other motion capture laboratories depending what kind of hardware is available. For Casa Paganini-Infomus the components to be powered on and checked for a correct functionality are the following ones: 13 MoCap cameras Oqus, 2 video JVC cameras, 4 working stations, PC Audio, PC Qualysis, PC Video, PC Sensors and Kinect.

   Then the Qualysis Track Manager software has to be executed and connected to the Qualysis realtime engine. After starting the QTM software, all mocap cameras are initialized and recognized by the software automatically. All components of the Recording platform have to be checked for the correct synchronization time as it is explained in details in the following pages.

2. System Calibration

   The QTM software must have information about the orientation and position of each camera in order to track and perform calculations on the 2D data transforming them into 3D data. The calibration is done by a well-defined measurement procedure in QTM. Calibration is, however, not needed for a 2D recording with only one camera.

   For 3D and 6DOF measurements, the system needs to be calibrated. QTM uses a dynamic calibration method. A wand is simply moved around in the volume, while a stationary reference object in the volume defines the coordinate system for the motion capture. All settings for the calibration are controlled by QTM,and the result of the calibration is visualized in a quick and intuitive way. The calibration is done within 60-120 seconds.

3. Subject Preparation

   When all the necessary Hardware/Software setup has been completed one can begin the subject calibration. The subject is the person whose motions will be recorded. First of all, the participant of the recordings has to be dressed into a MoCap Suit. The MoCap suit consists of pants, jacket and a hat made of elastic material with a velcro surface for attaching the infrared passive markers.

   In general, a tight suit is better, as long as it still allows the subject to move around uninhibited. If some parts of the suit are loose, velcro can be used to tighten it.

   Then, one proceeds with attaching markers to the subject as required by the marker set. There are different types of markers that have a different size (diameter of the sphere). The size of the marker does not matter in terms of subject definition; however, one should use smaller markers for smaller parts of the body, such as the feet and hands. After numerous recordings sessions and experimentation with different marker sets, we have developed the most effective positioning of the markers in order to avoid occlusion and loss of them. The marker set for the recordings in the framework of this thesis is made of 64 markers (see figure)

After markerization, the subject is equipped with accelerometers, attached to the hands and legs with elastic bands, together with a microphone and a transmitter.

4. Range of Motion (ROM)

   Range of Motion is a special type of motion that is used to process the subject calibration.

   in Figure 6.2 , some of the that were used in the ROM trial are presented. ROM is useful to capture the spatial range of a subjects motion

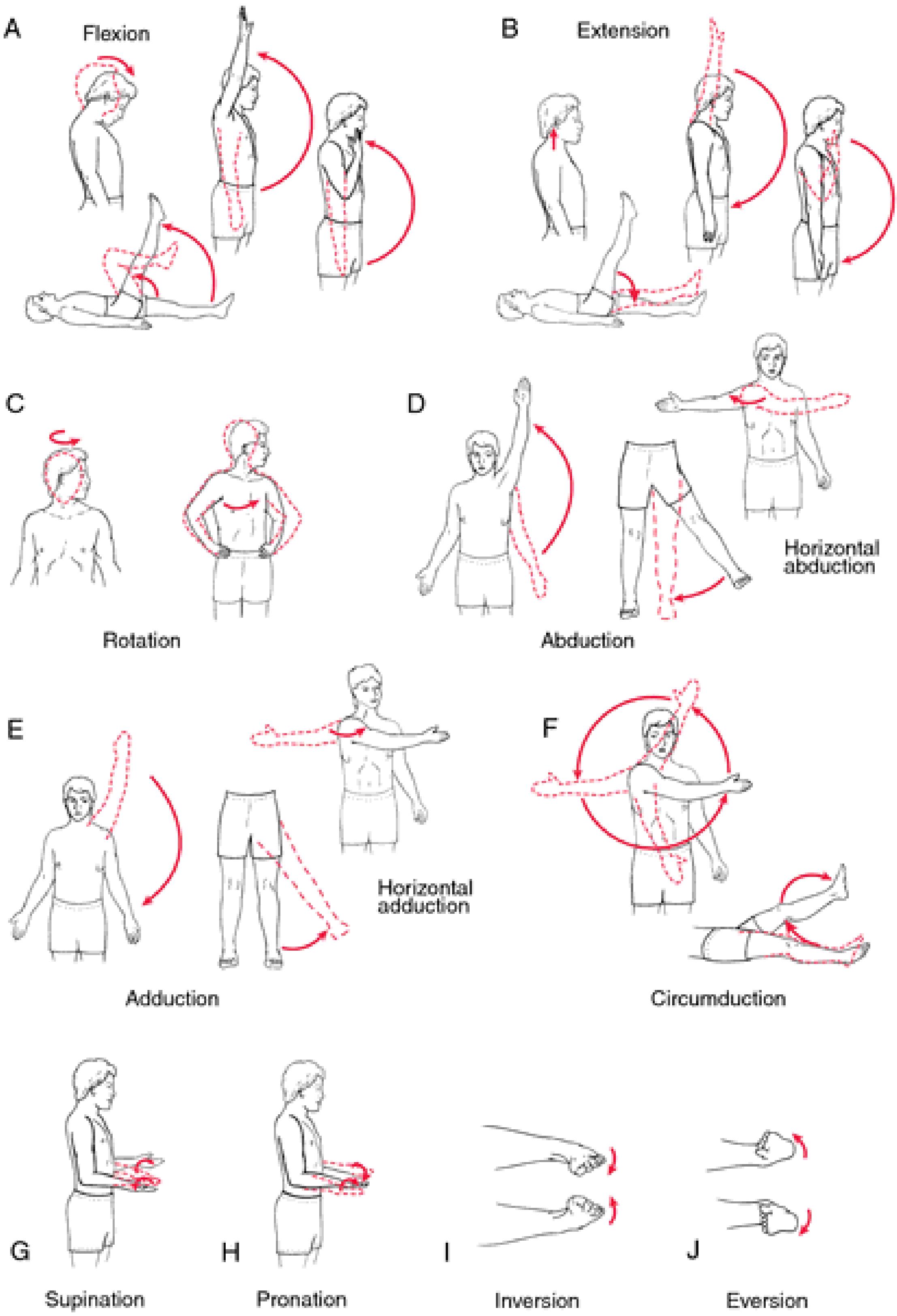

Figure 6.2: Some movements performed in the Range of Motion (ROM) phase

Every session of motion capture needs its own ROM capture for the subject calibration. Here we refer to ”session” as the motion capture data acquisition that occurs without any changes to the subject. For instance, if a subject takes off the suit for some reason and wears it again later, another session would begin, since the markers' positions may have changed.

The objective of ROM is to find the largest possible range of subject motion, so that the system can calculate all the possible range of variations for each marker and joint based on a template configuration. This information is used to process the rest of the motion capture data.

Every ROM trial starts should start with the T-Pose and end with the T-Pose (see Figure 6.3).

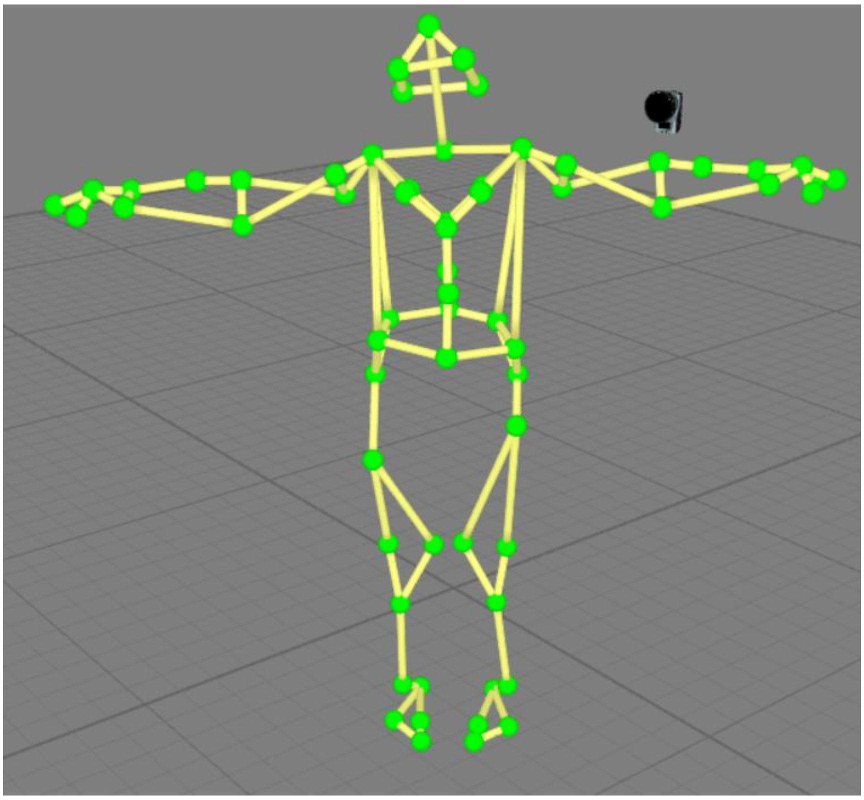

Figure 6.3: T-Pose and Markerized Automatioc Identification of Markers Model

The subject should stands in a T-Pose at the center of the capture volume, facing toward the Y-Axis. The subject should now moves all the joints as much as possible.

5. Subject Calibration

Subject Calibration is a special type of post-processing task for the ROM data. The goal of subject calibration is to generate a calibrated Automatic Identification Model (AIM) that includes all the details for subject specific data: markers and rigid body location for a particular subject.

Once we have finished with the ROM motion acquisition (usually 1 or 2 trials), we have to re-process the files with QTM in order to transform raw 2D data into tracked 3D measurements.

Before re-processing step described above, we cannot see any marker data in the 3D workspace, because these data only exit at first as a 2D image. One can see these 2D data in the camera view mode.

Next we have to proceed with generating an AIM. This model is applied in further data acquisitions of a particular subject in real time. Shortly, to create an AIM we have to follow the following steps: load the label list (including names and colors coding, and location of the bones), label (identify) the markers starting from the T-pose, remove noise, and finally AIM is generated.

6. General Motion Acquisition

   After the previous steps are completed, one can finally procees with the MoCap data acquisition for a specific scenarioì. All motion trials should still start with the T-Pose and end with the T-Pose.

   Motion Acquisition is a standard process, when all the components of the MoCap system are ready and calibrated, as well as the subject prepared. The system automatically names each trial and records the data on the hard disk.

7. Post Processing

   Post processing for motion data (not ROM data) is a little bit different from the ROM processing since at this step one has already an AIM model file that can be applied to the following motion trials for a particular subject. The AIM model file can help the system to process the rest of the motion data fairly accurately. This is why the Subject Calibration is important for each session.

   Nevertheless, this automated task sometimes does not provide the necessary quality and accuracy needed. If this is the case, we re-do labeling manually as instructed in the ROM processing section.

   There are five main steps in this process: 1) reconstructing the markers, 2) trajectory labeling 3) eliminating the noise and jittering markers, 4) filling the gaps of trajectories with linear or polynomial interpolation 5) saving the trial file.

8. Export Data QTM gives several possibilities to export the processed data.

.C3D format - this is used for analysis in Visual3D. C3D file is a format for programmers, manufacturers and end-users. The C3D format is a public domain, binary file format that has been used in Biomechanics, Animation and Gait Analysis laboratories to record synchronized 3D and analog data since the mid 1980's. It is supported by all 3D major Motion Capture System manufacturers, as well as other companies in the Biomechanics, Motion Capture and Animation Industries.

.TSV format – This format is used to export 3D/6DOF/force/analog data to tab separated value files, for analysis via Microsoft Excel.

Export to Matlab and LabView formats - One can also export 3D/6DOF/force/analog data to MATLAB and LabVIEW for further analysis.

6.2.1.2 Identification of trajectories and Automatic Identification of Markers (AIM) model

Automatic identification of trajectories in the QTM software is performed by a module called AIM (Automatic Identification of Markers). The AIM model is created from identified files and can then be applied to any measurement that captures similar motion with the same marker set. When the automatic identification fails, identification has to be performed manually. To get the best AIM model, it is important to continue adding QTM files with the same marker setup to the AIM model, if the automatic identification fails.

In order to create a model, a manual identification of all recorded trajectories has to be done, because only identified files can be used for creating the model. When the data are identified, the Automatic Identification of Markers (AIM) model can be created and applied to other files. This procedure reduces the identification time of MoCap data files, however the applied model does not give a perfect result, since 30-35 % of unidentified trajectories still has to be identified by hand.

Several problems may occur in the process of identifying trajectories. If there are unwanted reflections tracked in the file, they have to be deleted. Also ”ghost” markers may be tracked. These markers appear in a place where no marker is supposed to be, usually very far from the performer, and they have to be deleted as well. Another problems arise when the movement is too fast and positions of two markers are near to each other, since in this case they are often tracked as one single marker. Therefore, it is important to distinguish which marker generated each trajectory. In some cases, the trajectory can be split into two trajectories, in order to have one part that matches one marker and the other part that matches the other marker.

The following list of marker labels, that was created separately, was used for all the recordings:

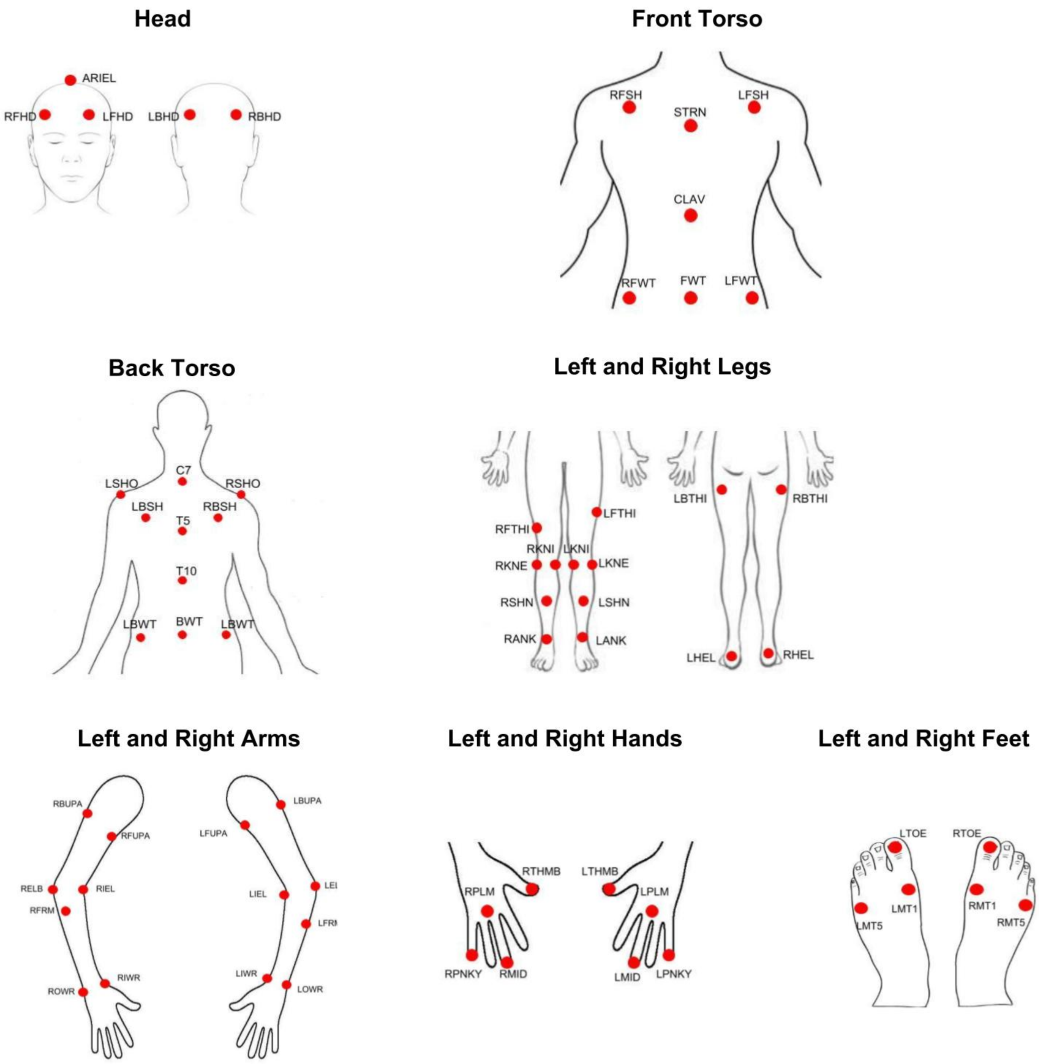

Figure 6.4: Marker set, 64 infrared markers

Head:

RFHD - Right Front Head
LFHD - Left Front Head
RBHD - Right Back Head
LBHD - Left Back Head
ARIEL - Top Head

Torso Back:

C7 - C7
LSHO - Left Shoulder
RSHO - Right Shoulder
LBSH - Left Scapula
RBSH - Right Scapula
T5 - T5 (on the 5th thoracic vertebrae)
T10 - T10 (on the 10th thoracic vertebrae)
BWT - Sacrum Bone
LBWT - Pelvic bone left back
RBWT - Pelvic bone right back

Torso Front:

STRN - Sternum
CLAV - Xiphoid process
LFSH - Left deltoid muscles
RFSH - Right deltoid muscles
FWT - Below Navel
RFWT - Pelvic bone right front
LFWT - Pelvic bone left front

Left and Right Arma:

LFUPA - Left front upper arm
LBUPA - Left back upper arm
LIEL - Left lateral elbow
LELB - Left medial elbow
LFRM - Left forearm
LOWR - Left outside wrist
LIWR - Left inside wrist
RFUPA - Right front upper arm
RBUPA - Right back upper arm
RIEL - Right lateral elbow
RELB - Right medial elbow
RFRM - Right forearm
ROWR - Right outside wrist
RIWR - Right inside wrist

Left and Right Hands:

LPLM - Left Palm
LTHMB - Left thumb finger

LMID - Left middle finger
LPINKY - Left pinky finger
RPLM - Right Palm
RTHMB - Right thumb finger
RMID - Right middle finger
RPINKY - Right pinky finger

Left and Right Legs:

LFTHI - Left thigh front
LBTHI - Left thigh back
LKNE - Left knee (outer side)
LKNI - Left knee (inner side)
LSHN - Left tibia
LANK - Left ankle
RFTHI - Right thigh front
RBTHI - Right thigh back
RKNE - Right knee (outer side)
RKNI - Right knee (inner side)
RSHN - Right tibia
RANK - Right ankle
LHEL - Left heel
RHEL - Right heel

Left and Right Feet:

LTOE - Left toe
LMT1 - Left 1st meta tarsal
LMT5 - Left 5th meta tarsal
RTOE - Right toe
RMT1 - Right 1st meta tarsal
RMT5 - Right 5th meta tarsal

Moreover, the data of each trajectory should be as good as possible. This is especially important if our model includes trajectories that are close to each other. Then a small erratic noise on a trajectory can make two trajectory paths come very close to each other, which makes identification difficult.

Therefore if the data is erratic, i.e. if the trajectory movement does not look smooth when the file is played, the erratic data should be deleted, and the gap filled with a suitable approximation.

Since multiple subjects were captured, a separate AIM model for each subject had to be created.

Gap fill trajectories For several markers, trajectories were missing in some segments of the recordings. The reason is that the movements of the participant were such that these markers were covered by other parts of the body or jitter. The most problematic areas were hips, knees, and both hands, due to specific movements in karate. In these parts of the recordings the missing trajectories were filled in using a feature of the Qualisys Track Manager – Gap fill Trajectory with Preview, which uses Linear or Polynomial interpolation.

6.2.2 Synchronized Multimodal Recording and Playback

6.2.2.1 Architecture

The overall architecture for multimodal recordings is shown in the Figure 6.5:

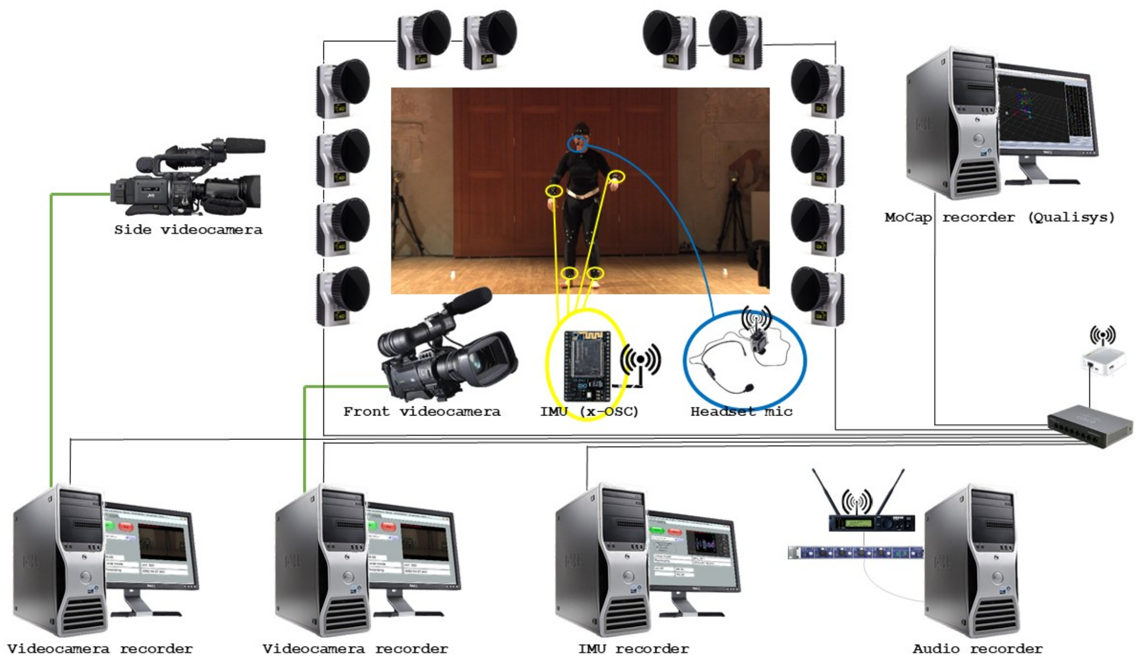

Figure 6.5: The architecture of the recording platform

The performer movements are captured by a Motion Capture system (MoCap recorder in the figure). The performer also wears a headset microphone, which is used to record breathing noise for possible further analysis. Moreover, the performer also wears four Inertial Measurement Units (IMU): x-OSC, two on the wrists and two on the ankles. Finally, two broadcast quality video cameras are observing the scene, one from the front and one from a side.

Synchronization is guaranteed by the EyesWeb platform. On the MoCap recorder computer, EyesWeb is used to generate the reference clock used by all other recorders. The generated reference clock is sent to the other devices in a format compatible with each specific device. As an example, the Qualisys Motion Capture system receives such clock encoded in an audio stream, in SMPTE format.

SMPTE timecode is a set of cooperating standards to label individual frames of video or film with a time code defined by the Society of Motion Picture and Television Engineers in the SMPTE 12M specification. SMPTE revised the standard in 2008, turning it into a two-part document: SMPTE 12M-1 and SMPTE 12M-2, including new explanations and clarifications. Timecodes are added to film, video or audio material, and have also been adapted to synchronize music and theatrical production. They provide a time reference for editing, synchronization and identification. Timecode is a form of media metadata. The invention of timecode made modern videotape editing possible, and led eventually to the creation of non-linear editing systems.

Also the two broadcast video-cameras and the Audio recorder use SMPTE encoded as an audio signal. The IMU recorder receives the reference clock via network, through the OSC protocol.

To guarantee synchronization, EyesWeb keeps track, for every recorded frame or sample, of the timestamp when the data was received. As a matter of facts, not all streams can be hardware-synchronized (e.g., with a genlock signal), thus, a software synchronization is performed by EyesWeb, by keeping track of the time at which the data were received in a separate file, and using such information when playing back the data. IMU sensors or Kinect are examples of devices which are synchronized in this way. The Syncronization tools and patches that we used are programs, written to be executed by EyesWeb, that allow the user to record, playback and analyze multimodal data (video, audio, motion capture, sensors). To run tools one needs to download the corresponding installers, launch them, and execute the tools as normal Windows applications. To run patches one needs to download and load them into the EyesWeb application.

In the following pharagraph we illustrate some example tools and patches belonging to the chategory of synchronization of multimodal devices and recordings.

6.2.2.2 Kinect recorder tool

The Kinect recorder tool is depicted in the Figure 6.6.

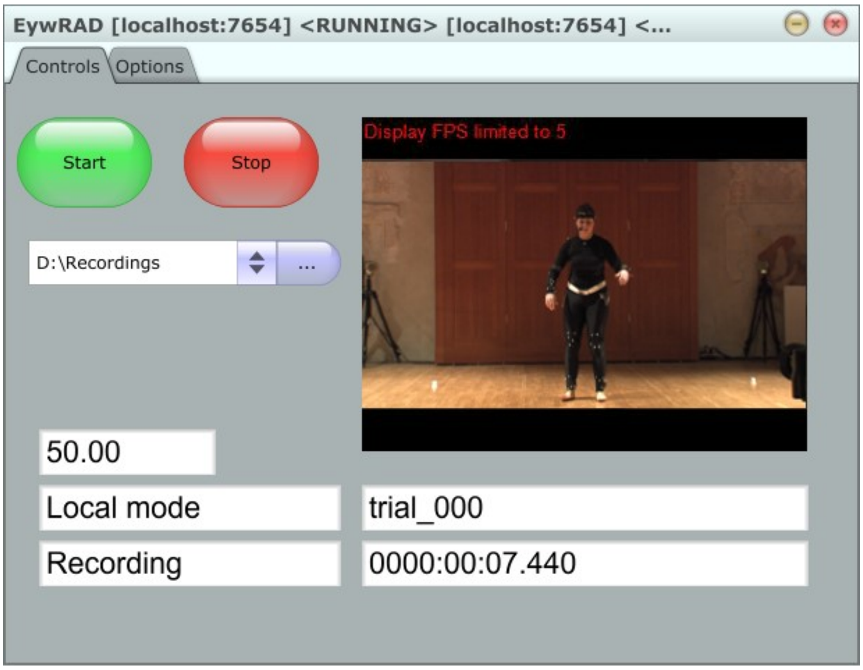

Figure 6.6: Kinect recorder tool

The tool shows the current framerate (50.00 frames per second in the example), the name used for this trial (trial_000, progressive numbers are automatically assigned to each trial), and the value of the reference clock (HHHH:MM:SS.mmm; 0000:00:07.440 in the Figure 6.6).

The recording tool records avi files. The video is encoded in MPEG4 format, the resolution is 1280x720 and the framerate is 50 fps. Audio is encoded in AAC format at 48000Hz. Two channels are recorded: the left channel contais audio from the system's audio input device (e.g., a microphone), whereas the right channel is the reference clock encoded in SMPTE audio format. Multiple instances of the video recorder tool can be started and can work standalone, or synchronized with the other recorders. The options panel allows one to configure the working mode of the recorder.

6.2.2.3 Audio recorder tool

The audio recorder tool is depicted in the Figure 6.7.

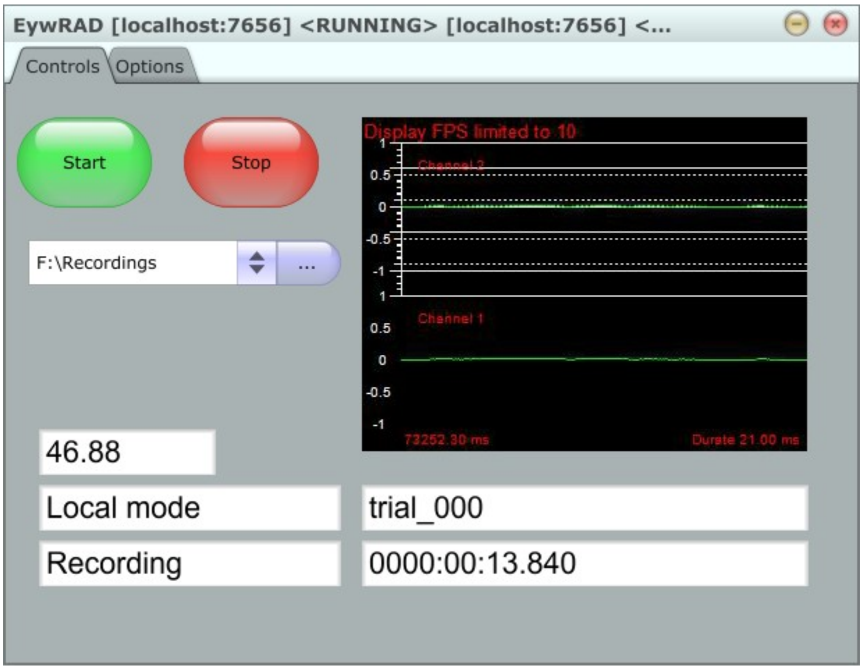

Figure 6.7: Audio recorder tool

Currently, only the headset microphone was being recorded in the recording sessions.

The headset microphone provides a monophonic signal, but it is saved as a stereo file in WAVE format, as the SMPTE is added as the right channel. Audio is sampled at 48000Hz.

The user interface is very similar to the video recorder tool, the main difference being of course the visualization part. In this tool, the audio waveform is shown instead of the video stream. The options panel allows one to configure the working mode of the recorder.

6.2.2.4 IMU recorder

The IMU recorder tool is depicted in the Figure 6.8.

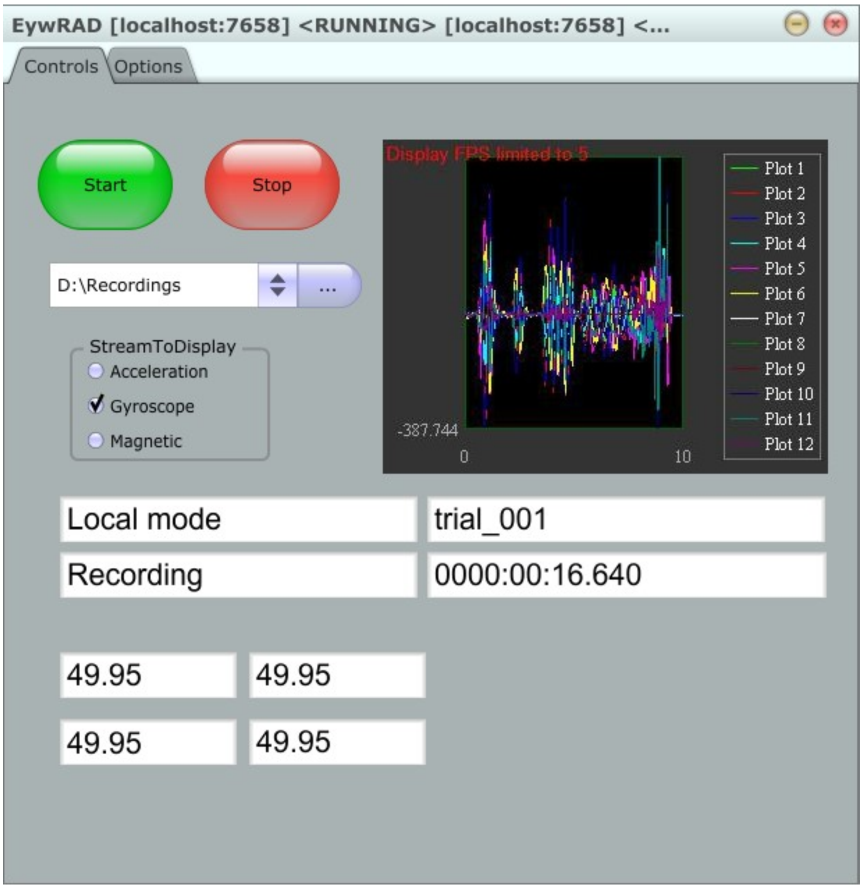

Figure 6.8: IMU recorder tool

The graph shows the values selected by the user (acceleration, gyroscope, or compass) for each of the four IMUs. In the lower left one can read the current framerate of each of the four sensors (49.95 samples per second). Below the graph one can see both the trial name and the reference clock.

The data is saved by the recording tool in txt files, in a format which is easy to be read by external software (e.g., Matlab), and can be of course read by EyesWeb itself for playback or analysis purposes.

The following is the format of the OSC addresses used to communicate data extracted by sensors (IMUs) embedded in wearable devices, x-OSC sensors messages: /imu: 10 floating-point values corresponding to the data captured by the sensors integrated on the IMU (gyro-x, gyro-y, gyro-z, acc-x, acc-y, acc-z, mag-x, mag-y, mag-z, gyro-temp).

The options panel, which is the same as for audio and video recorder, allows one to configure the working mode of the recorder.

6.2.2.5 Options panel

All the recording tools share the same options panel depicted on the Figure 6.9:

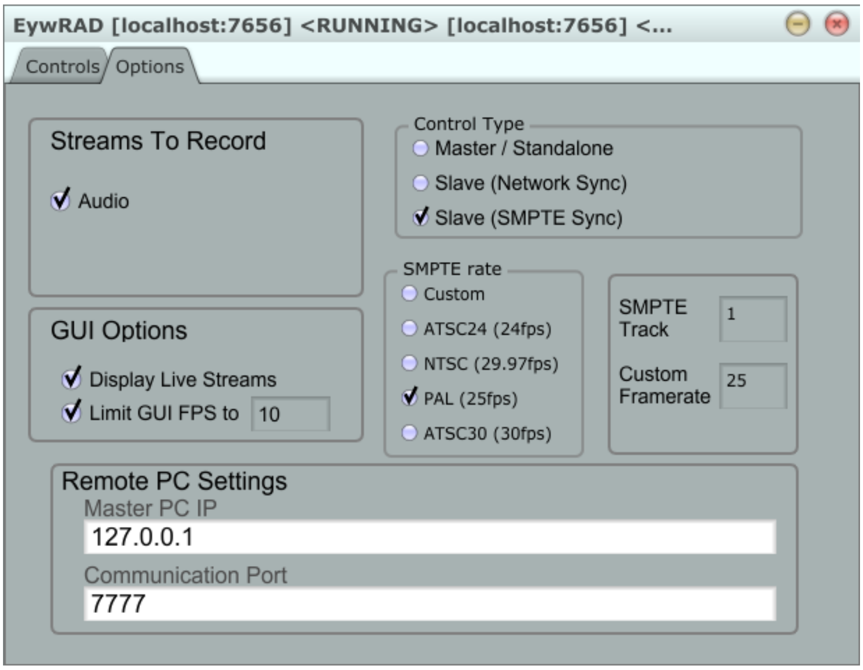

Figure 6.9: IMU recorder tool

The control type section controls the synchronization mode. If set in master/standalone mode, the tool works with its own clock, without synchronizing with the other devices. In slave mode, the tool receives the clock from an external device (the master). The clock can be received both via Network (OSC protocol) and via audio (SMPTE Sync).

6.2.3 Genoa Experimental Recordings

Casa Paganini-InfoMus research center at UNIGE, consists of an ancient monumental building endowed with a 230-seat auditorium and some museum rooms. The main lab room offers a direct view of the auditorium's stage. This configuration is exemplary of the interaction between artistic, scientific and technological research.

Moreover, it enables the development of experiments, prototypes, and demonstrations in an almost real-world setting with ecological extensions. The whole monumental building has a technological infrastructure, such as fast network, audio and video connection, including multichannel audio and video.

As was described in the previous section, the EyesWeb XMI platform, used at Casa Paganini Infomus, integrates Qualisys with other sensor systems, including accelerometers, RGB and RGBD video cameras, smartphones, microphones, and biometric sensors, for recordings of movements, gestures, audio, video, and physiology. EyesWeb enables synchronized recording, analysis, and playback of data.

In this section we present the recordings that have been done in the framework of H2020 Wholodance Project, and were used for the testing purposes of our method in the framework of this project.

The main part of the session has been dedicated to the multimodal recordings. The performers and dance partners have been prepared for the session working on specific exercises before arriving to Genoa. During these sessions, the focus was given to Movement Qualities, i.e., expressive aspects of movement such as Origin of Movement, Fluidity, Coordination, Light vs. Weighty movement etc.

The recordings focused on contemporary dance, since, as compared with other forms of dance, it is more open to experimentation and to the exploration of different qualities of movement. During the recordings and data acquisitions, and while watching the videos of the capturing process, discussions took place, aiming among others at the common understanding of the Movement Qualities and at the conceptual framework for WhoLoDancE.

The following process has been followed to record a series of short fragments (1 to 2 min each) of synchronized multimodal data (MoCap, audio, IMUs (Inertial Measurement Units), 2 video cameras), focusing on specific movement qualities related to contemporary dance scenarios. The recording sessions have been done without music because the partners agreed that music can impose specific movement qualities and can affect the way the dancer performs a movement.

A microphone has been placed on the dancers in order to record the breath for possible further processing. For the recordings, 13 cameras + 5 accelerometers + 2 high definition cameras have been used + 1 microphone + 1 kinect camera. Two contemporary dancers have been captured, following a list of exercises on Movement Qualities / High Level Features.

The analysis and discussion took place only on the first day, to concentrate the work on the capturing sessions in the following ones. The partners of the project had the chance to have a general discussion and set the deadlines for the analysis and annotation to be completed.

After the Post Processing phase, the recordings with the best qualities of trajectories were selected, then these MoCap recordings were segmented into small segments lasting about 10-15 s (100 segments in total). These were used for the testing purposes of our method.

Chapter 7

Implementation

Contents

We tested our method using a data set extracted from a larger data set of Motion Capture data of subjects performing expressive movements, recorded in the framework of the WholoDance EU project. The main topics of the recordings were fluidity, weight, external and internal propagation of movement, balance, equilibrium, coordination, and suddenness. In order to create the data set, first, we started acquiring data on volunteer dancers, with the aim of defining and building a reliable and effective Motion Capture setup. Then, we proceeded capturing data on professional dancers. In summary, the multimodal data set of the project consists of 127 trials (recordings), acquired with the aim of investigating movement, defining movement features, and developing techniques for their computation.

Two professional dancers took part in the recordings, using a Motion Capture system with 2 video cameras. We recall that the dancers were equipped with 64 infra-red reflective markers, 5 accelerometers, and a microphone. After the recording sessions, the data were post-processed and cleaned, and the origin, path, and destination of the movement were annotated by the experts. In our initial implementation of the proposed method, the following steps were performed.

7.1 Setup

1. The initial motion capture data set (x; y; z positions of the 64 reflective markers for each frame of the recordings, see Figure 6.3 for a sample frame) was transformed into a reduced data set associated with a smaller number (20) of joints: more precisely, head, hip center, left ankle, left elbow, left foot, left hand, left hip, left knee, left shoulder, left wrist, right ankle, right elbow, right foot, right hand, right hip, right knee, right shoulder, right wrist, shoulder center, spine.

   Compared with the initial 64-joints model, using this 20-joints skeletal structure allowed a faster implementation, useful for testing purposes. Moreover, using the 64 initial markers allows us to interpolate the reduced skeleton of 20 markers with better precision in case some of the data of 64 markers is missing.

2. Then, for each frame, starting from the position of each among these 20 joints, we computed its energy as the motion-related feature needed by the method. This choice was motivated by simplicity reasons, and also by the ease of visualization of such a feature.

3. For each frame, we computed the Shapley values of all these 20 joints, using the method detailed in Chapter 4. Then, we extracted the "most important" joint, according to the proposed method.

   In order to conduct our preliminary tests, the whole method was implemented in MATLAB. Due to the reduced number of joints, we fixed 4 as the maximum number of clusters to be detected by spectral clustering (such a small number of clusters was chosen due to the small number of vertices of the skeletal structure).

4. To ease the evaluation of the results, we visualized them, highlighting in red, frame-by-frame, the most important joint extracted by the method inside the 20-markers skeletal graph (see Figure 8.1 for an example of visualization of the results).

   The EyesWeb XMI1 open platform was used to implement this visualization step. The preliminary results obtained so far (see Figure 8.1 for a sample) showed that, on the data set used for the evaluation, the method was able to extract meaningful joints, i.e., it identified, as important joints, ones also identified in preliminary annotations. Additional and more quantitative validations of the method will be considered in future works.

7.2 Software Modules

7.2.1 EyesWeb Patches

This Section describes the EyesWeb applications (Patches) developed to: (i) read and conver MoCap data from the full marker set used during the recordings to a simplified set of joints (i.e., matching the Kinect sdk joint set) and (ii) read Shapley analysis results and MoCap data from file, and render them on a point-light display highlighting, frame by frame, the joints corresponding to the maximum Shapley values in order to generate stimuli for algorithm evaluation.

Section 7.2.1 describes the Eyesweb Patch developed to produce the simplified joint set from the full motion captured marker set, while Section 7.2.2.1 describes the patch developed to produce the point light displays showing the evolution of the extracted Shapley values trough time.

7.2.2 Generating a Simplified Marker Set

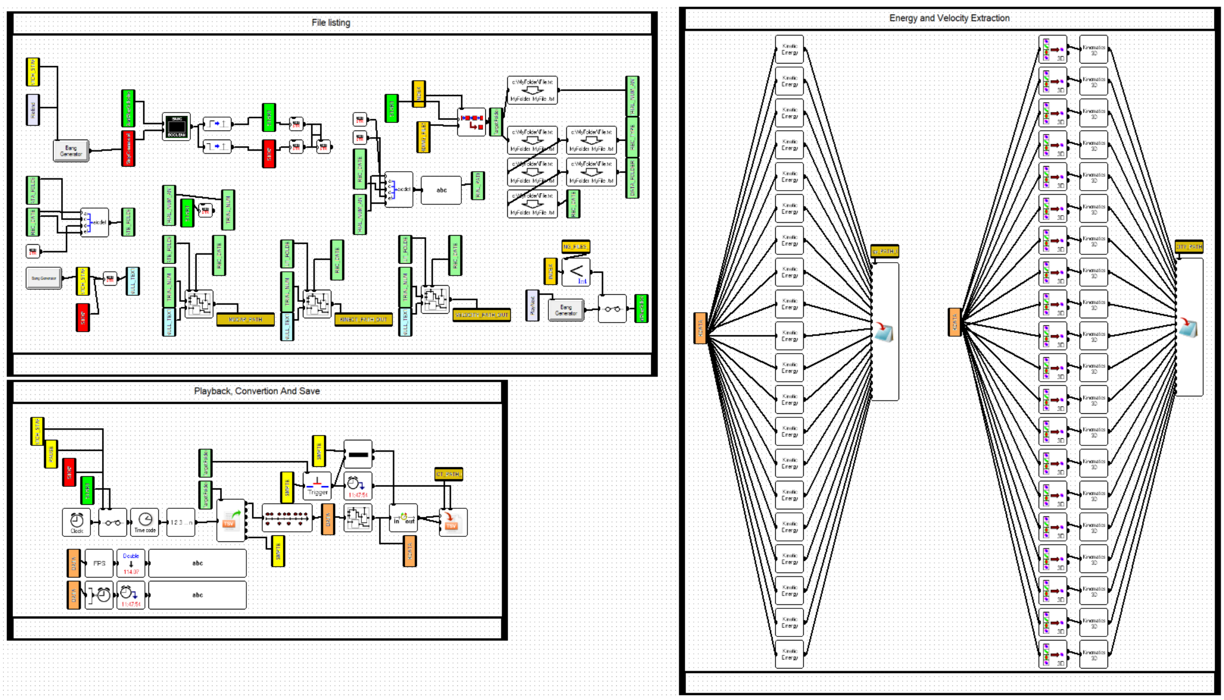

Figure 7.1: Overview of the patch that converts the initial set of joints to a simpler set of joints

The EyesWeb patch described in this section was developed to convert the marker set used during Motion capture to a smaller set, in particular to the marker set used by Microsoft SDK [1]. This was done with the purpose of making the algorithms compatible with low-cost capture devices.

Figure 7.2 summarizes the main components of the patch for remapping coordinates of MoCap data.

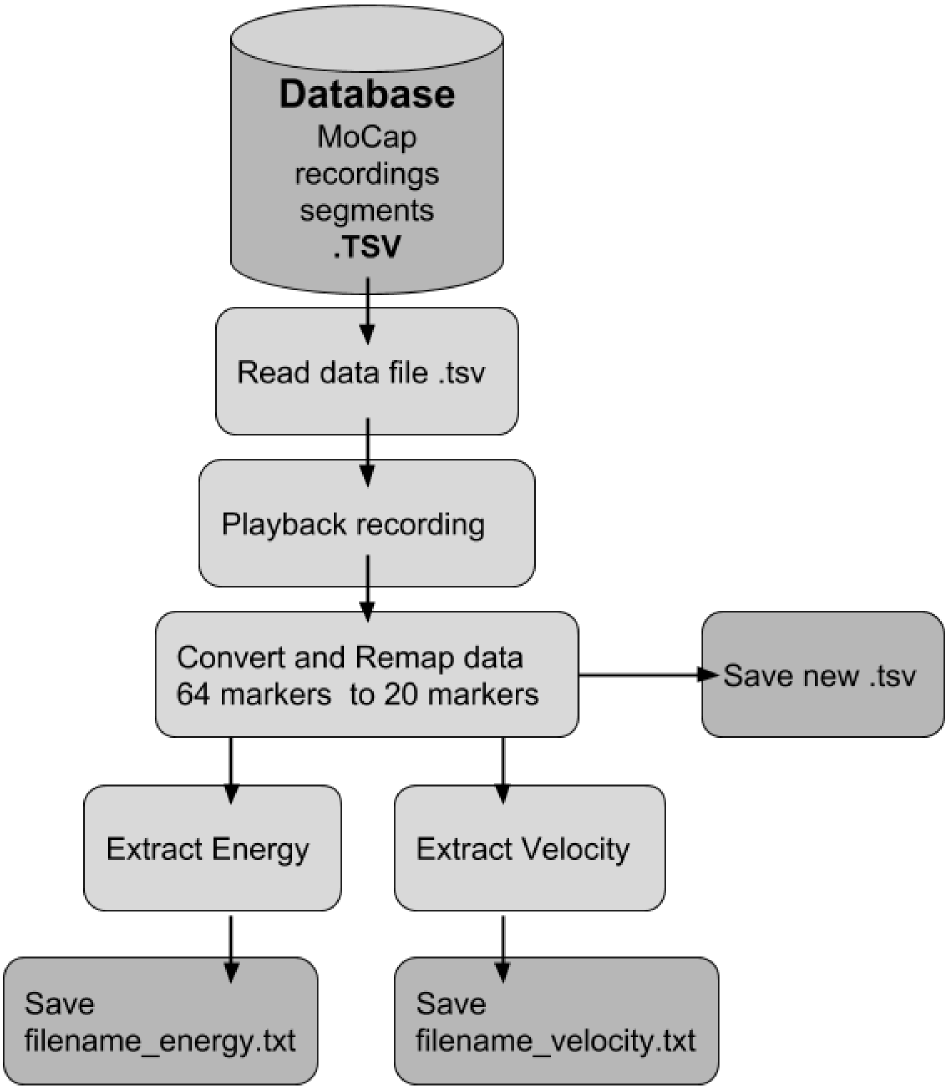

Figure 7.2: General Plan of the Work

[1] https://www.microsoft.com/en-us/download/details.aspx?id=44561

Figure 7.1 shows an overview of the patch that performs the following operations:

- it lists and reads the available recordings from hard drive automatically and sequentially (top left);
- it playback the recording, converts original coordinates to the simplified marker set, and saves the new coordinates to new TSV file (bottom left);
- it extracts Kinetic Energy and Velocity for each simplified marker. These two quantities were used in the process of computing the Shapley values (kinetic energy as weight of edges) and evaluate the results (see Chapter 8), respectively;

Each portion of the patch enclosed in the black boxes performs one of the described tasks (from left to right, top to bottom).
Figure 7.3 shows the portion of the patch that reads the list of available recordings, manages automatic file playback, processes automatically each element in the list of found recordings and finally generates file names for the TSV files that contain the simplified marker set and the files that store Kinetic Energy and Velocity values for each joint at each frame.

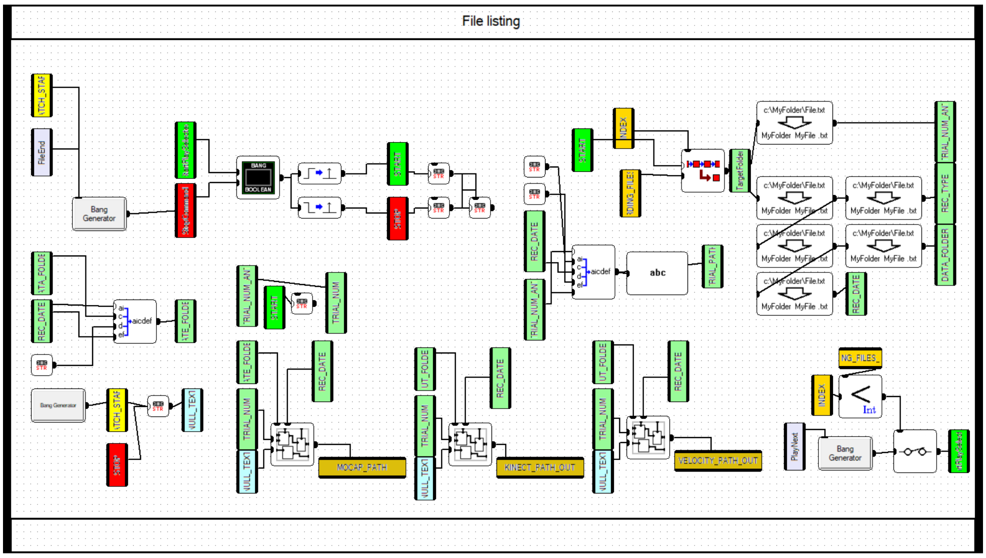

Figure 7.3: Portion of the patch that generates the list of files available on the hard drive.

Figure 7.4 shows the part of the patch that performs the playback of the motion capture recordings, converts the marker set to the simplified marker set; and saves the new generated marker set to file; the conversion process is detailed in Section 7.2.2.

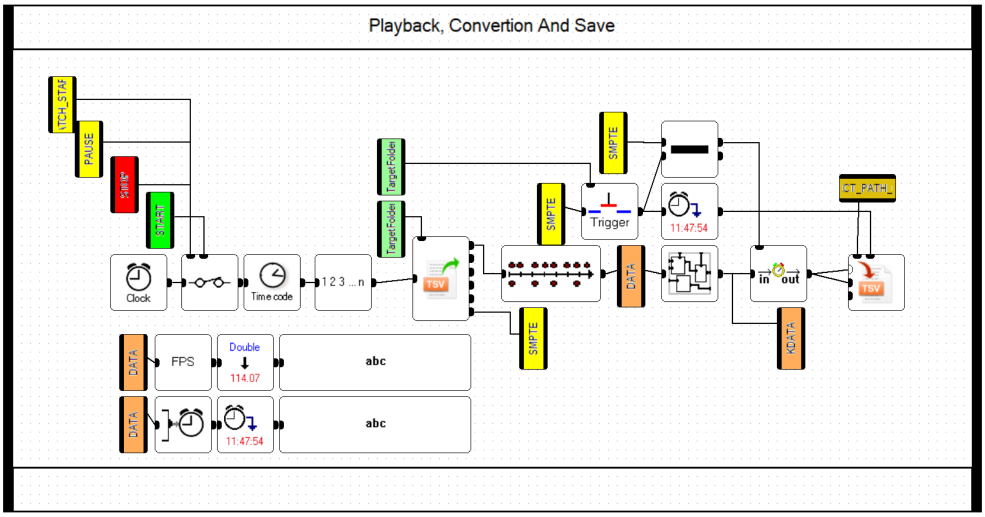

Figure 7.4: Portion of the patch that reads a single recording, generates the simplified marker set, and saves it to a new TSV file.

Figure 7.5 depicts the portion of the patch that extracts Kinetic Energy (left) and Velocity (right) of each joint of the simplified set and saves it to file.

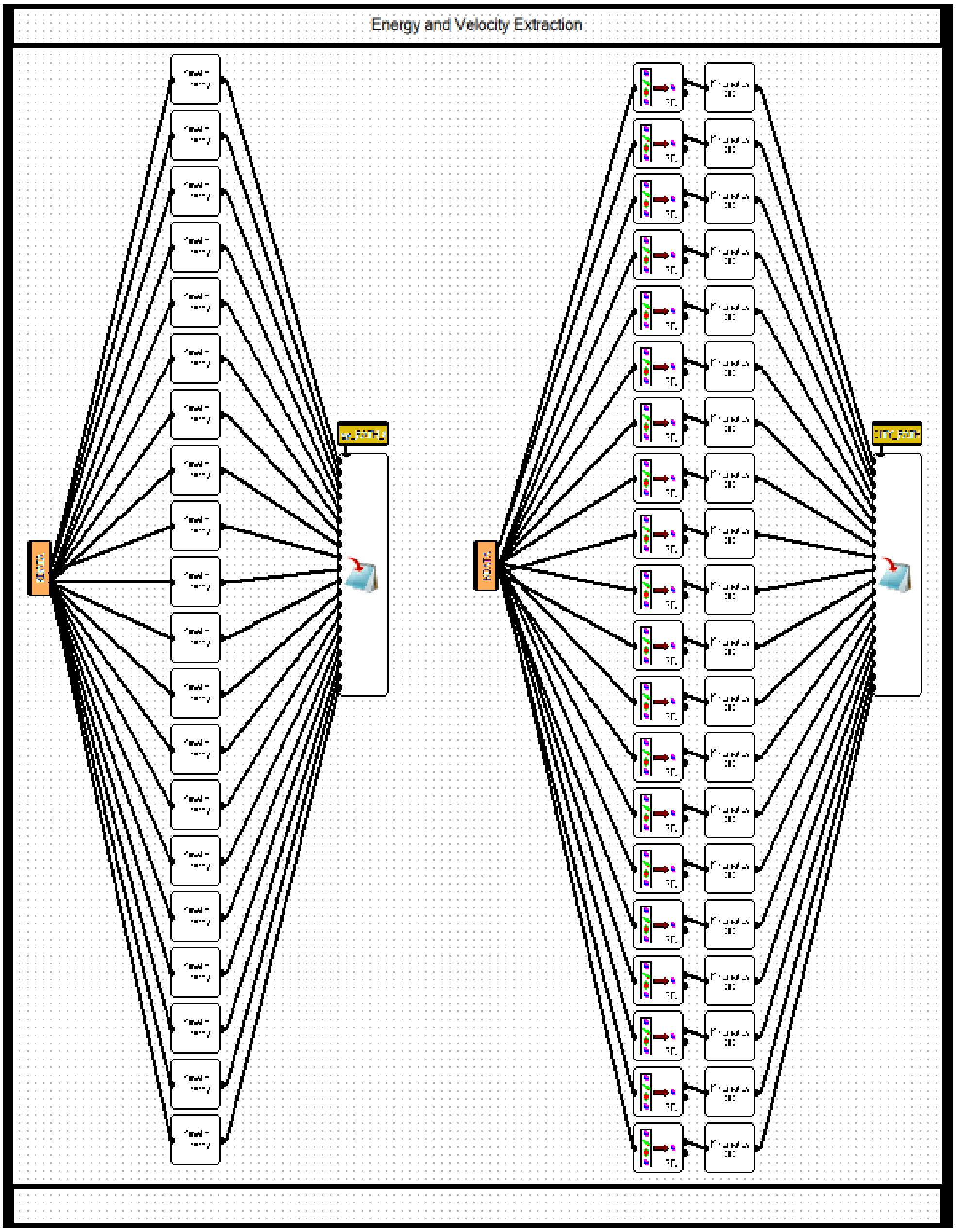

Figure 7.5: Portion of the patch that computes Kinetic Energy and Velocity for each joint of the simplified set

7.2.2.1 Generation of a simplified marker set from full body motion capture

As explained in Chapter 6, the motion capture process produced three dimensional recordings of full-body dance sequences composed by 64 markers. In order to simplify the computation of the Shapley value, reduce the amount of missing trajectories of MoCap data, and enable the use of the developed algorithms with low-cost devices, we decided to reduce the marker set to match the one used by Microsof Kinect [2]. This process is performed in the patch described in the previous Section, in particular the conversion process is performed by the portion of the patch depicted in Figure 7.6, where each of the depicted blocks performs the conversion of a set of original joints (64) to a single joint of the simplified set (20).

[2] https://developer.microsoft.com/en-us/windows/kinect

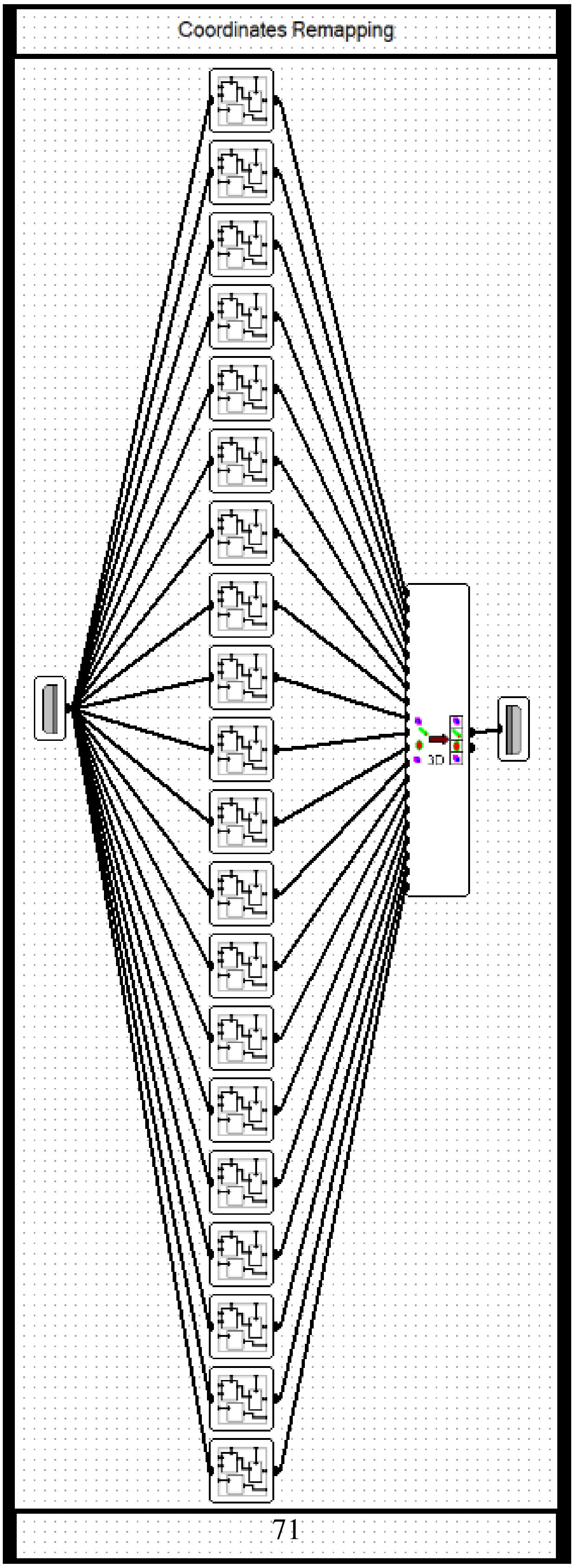

Figure 7.6: Portion of the patch that converts the original set of joints to the simplified 20 joints set

Each one of the 20 blocks shown in Figure 7.6 is actually a nested patch (an example is shown in Figure 7.7) that performs the conversion of a group of markers of the original set to a single marker of the simplified chosen marker set. The conversion process involves the selection of specific markers of the original set through their labels, the computation of the geometric barycentre of the group, and the generation of a new label for the originated barycentre that will represent the marker in the simplified set. In particular the barycentre computation is performed according to the following formula 7.2.2.1, where x_b, y_b and z_b represent the coordinates of the barycentre, while x_j^o, y_j^o, z_j^o are the coordinates of the $j-$th marker of the n chosen to be used to compute the new marker of the simplified set. Table 7.1 shows how the markers of the original set were used to generate the new simplified set.

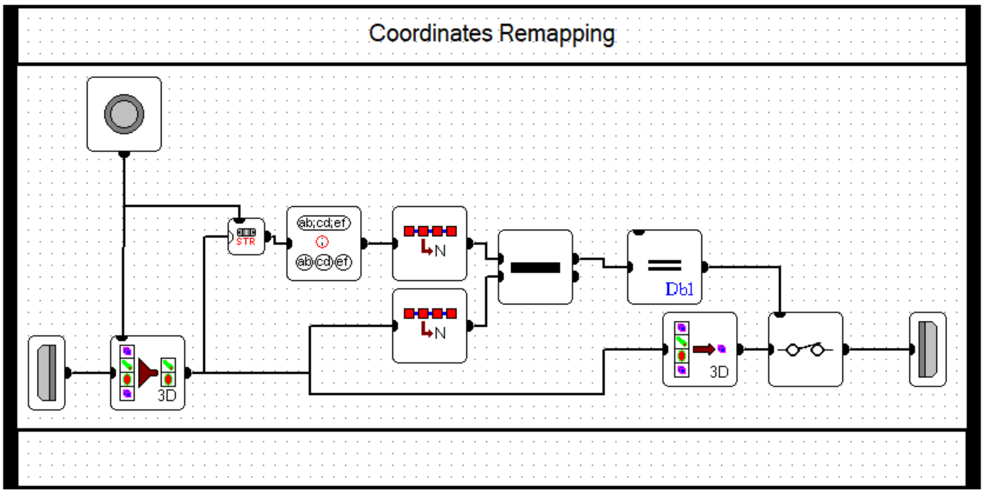

Figure 7.7: The nested patch that converts a group of markers of the original set to a single one of the simplified set

$$x_b = \frac{\sum_{j=1}^{n} x_j^o}{n} \qquad , y_b = \frac{\sum_{j=1}^{n} y_j^o}{n} \qquad , z_b = \frac{\sum_{j=1}^{n} z_j^o}{n} \qquad . \tag{7.1}$$

Table 7.1: Correspondence between original marker set labels and simplified marker set labels.

Original Set Marker Labels	Simplified Set Marker Label
ARIEL, RFHD, LFHD, LBHD, RBHD	head
C7, CLAV	shoulder_center
T10	spine
BWT, LBWT, RBWT, RFWT ,LFWT	hip_center
LSHO, LBSH, LFSH	left_shoulder
RSHO, RBSH, RFSH	right_shoulder
LIEL, LELB	left_elbow
RIEL, RELB	right_elbow
LOWR, LIWR	left_wrist
ROWR, RIWR	right_wrist
LPLM, LINDX, LMID, LPNKY	left_hand
RPLM, RINDX, RMID, RPNKY	right_hand
LFWT, LBWT	left_hip
RFWT, RBWT	right_hip
LKNE, LKNI	left_knee
RKNE, RKNI	right_knee
LANK, LHEL	left_ankle
RANK, RHEL	right_ankle
LMT1, LMT5	left_foot
RMT1, RMT5	right_foot

7.2.3 Stimuli Saving

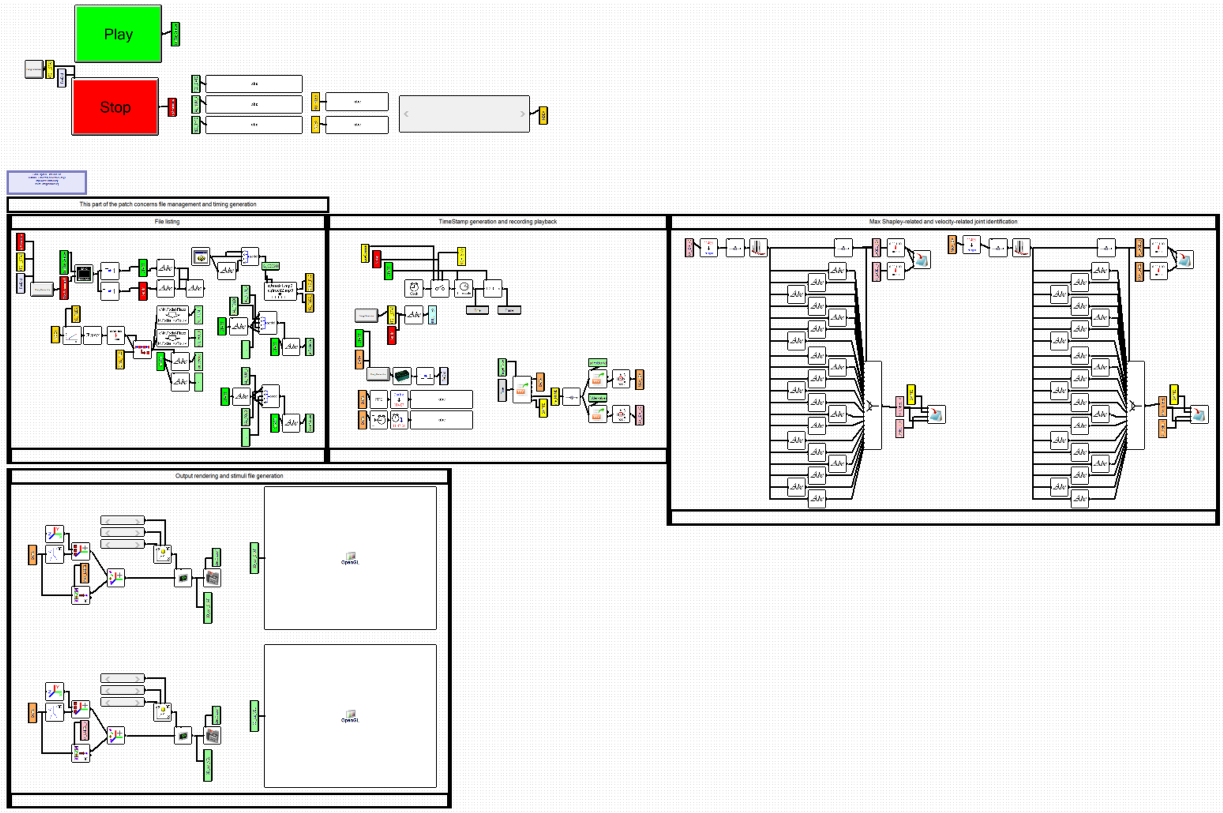

Figure 7.8: Overview of the 3D rendering patch

The EyesWeb patch described in this section was developed to render 3D point light displays of the extracted maximum Shapley value-related and maximum velocity-related joints, to generate stimuli for the evaluation of the technique proposed in this dissertation.

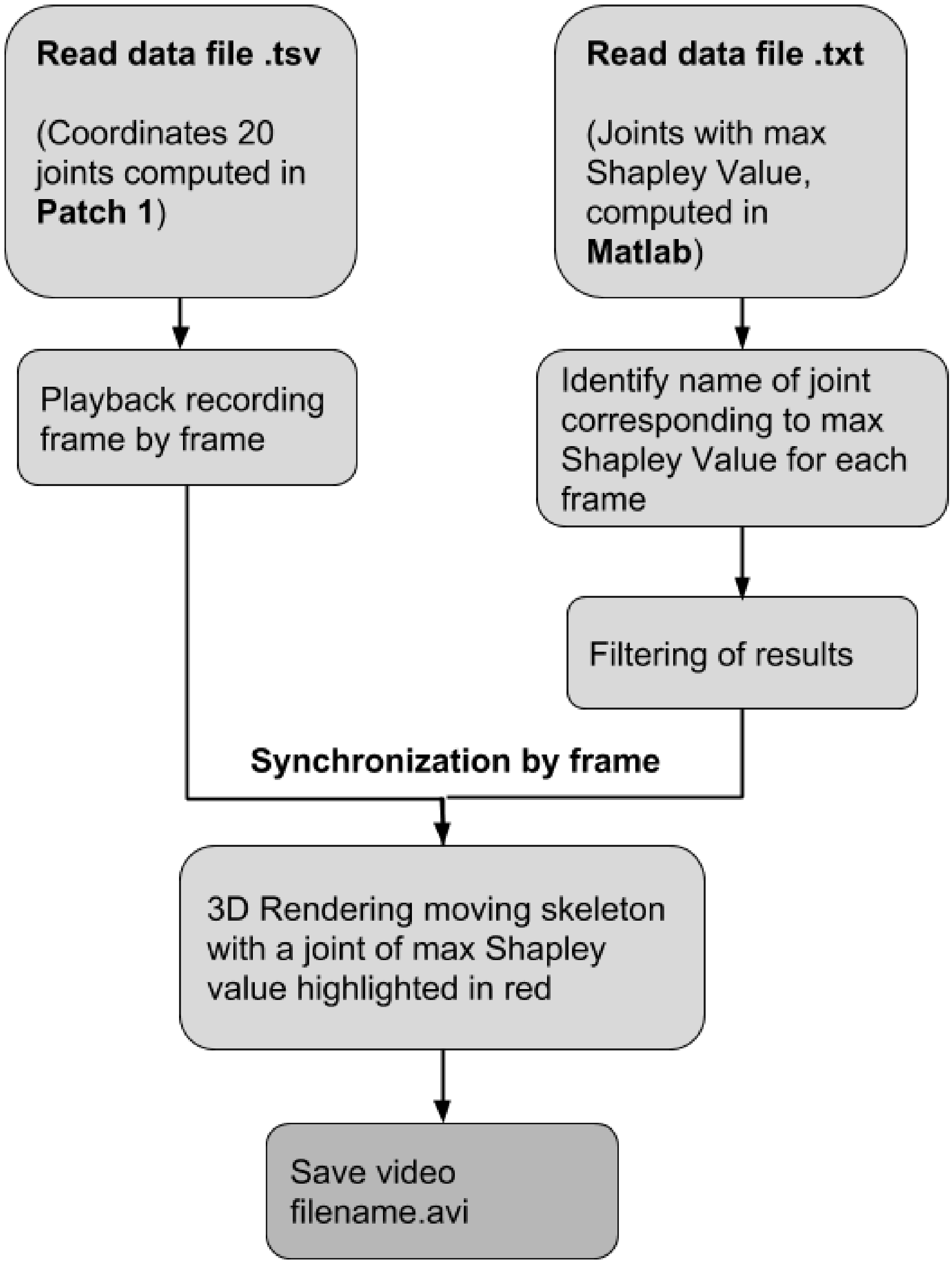

Figure 7.9: General Plan of the Work

Figure 7.9 summarizes the work flow to compute the Shapley value and visualize the results of the computation on a human-like skeleton

Figure 7.8 shows an overview of the patch that performs the following operations:

- it lists and reads the available recordings from hard drive automatically and sequentially;
- it playbacks the recording and Shapley values and Maximum velocities extracted using the modules described in Chapter 6 in a synchronous way;
- it identifies the joint corresponding to the highest Shapley values and the highest velocity value;
- it generates a 3D Rendering of the tracked joints and saves it to a file to provide video stimuli for evaluation purposes.

Each portion of the patch enclosed in the black boxes performs one of the described tasks (from left to right, top to bottom).
Figure 7.10 shows the portion of the patch that reads the list of available recordings (top right portion), manages automatic file playback (top left), process automatically each element in the list of found recordings (middle left), and finally generates file names for the 3D rendered videos (bottom right).

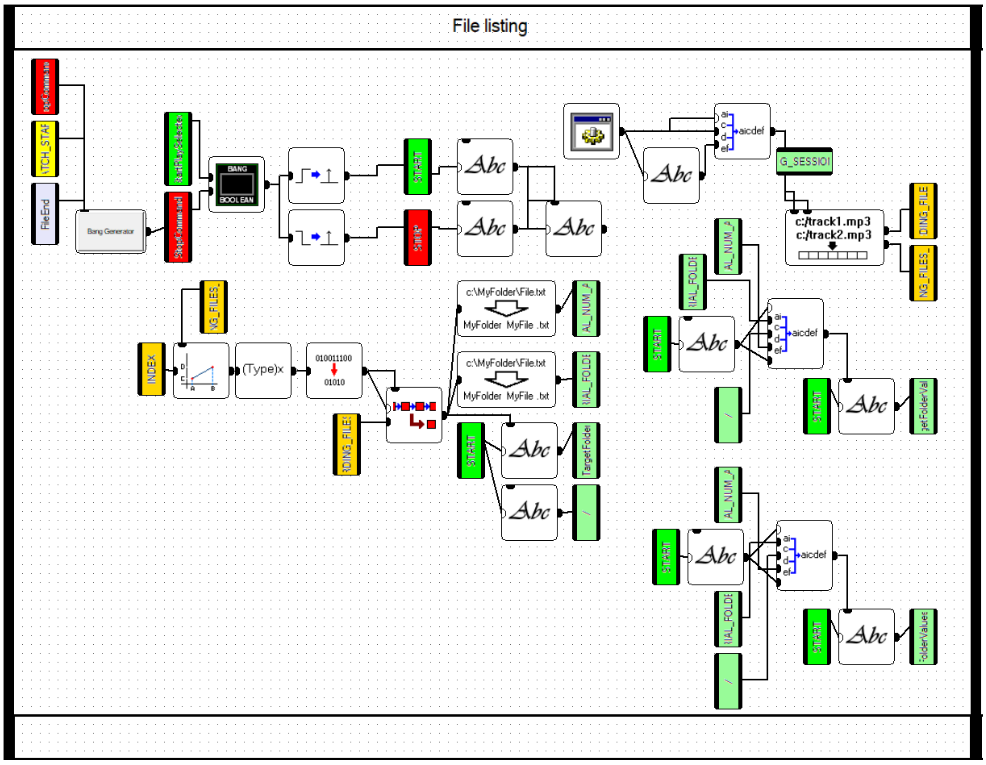

Figure 7.10: Portion of the patch that generates the list of files available on the hard drive.

Figure 7.11 shows the part of the patch that performs the playback of motion capture recordings and of the files containing extracted maximum Shapley values and maximum velocity associated to each joint: the top-left part generates a time-code at a fixed frame rate (100 fps) corresponding to the frame-rate at which the motion capture recordings were captured; the time-code is then used in the bottom right portion to play the recording back: three TSV file reader blocks read respectively (from left to right): the TSV file containing the simplified set of user's joints; the maximum Shapley values extracted at the corresponding frame, and the maximum velocity for the joints at the corresponding frame.

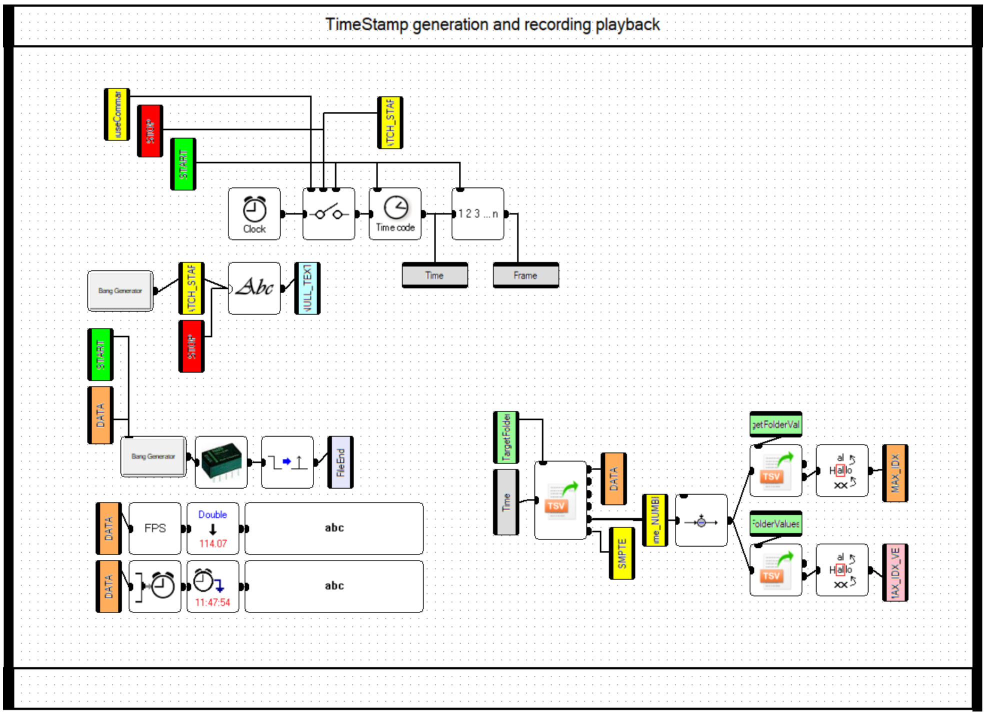

Figure 7.11: Portion of the patch that reads a single recording and the corresponding Shapley values and velocity values

Figure 7.11 depicts the portion of the patch that extracts the names and identification numbers of the joints that correspond to the maximum Shapley values (left) extracted at a given frame, and the maximum velocity (right) for the joints at the corresponding frame. The associated joint names and identification numbers, including the corresponding frame number and time-code, are also saved in a file for further subsequent analysis.

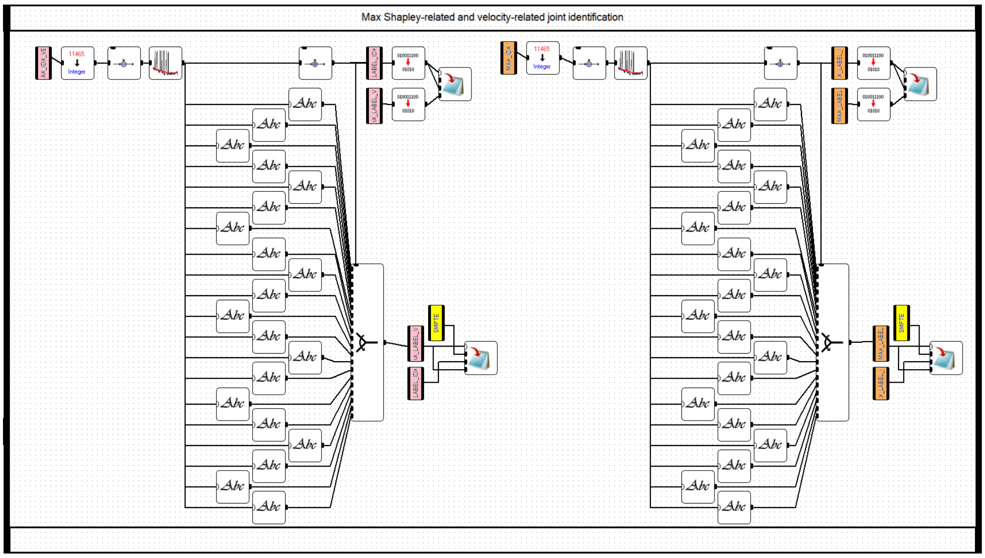

Figure 7.12: Max Shapley-related and velocity-related joint identification

Finally Figure 7.13 depicts the part of the patch that renders 3D displays of a skeletal representation of the motion captured sequence, and highlights the joint corresponding to the maximum Shapley values (top) extracted at a given frame, and the maximum velocity (bottom) at the corresponding frame. In particular, two different displays show the same skeleton (represented by interconnected spheres) having a single joint rendered in a different colour (i.e., red), which is the joint having the maximum value of the desired feature). The 3D renderings are saved in a file (1280X720@25fps mpeg4 encoded videos, avi format).

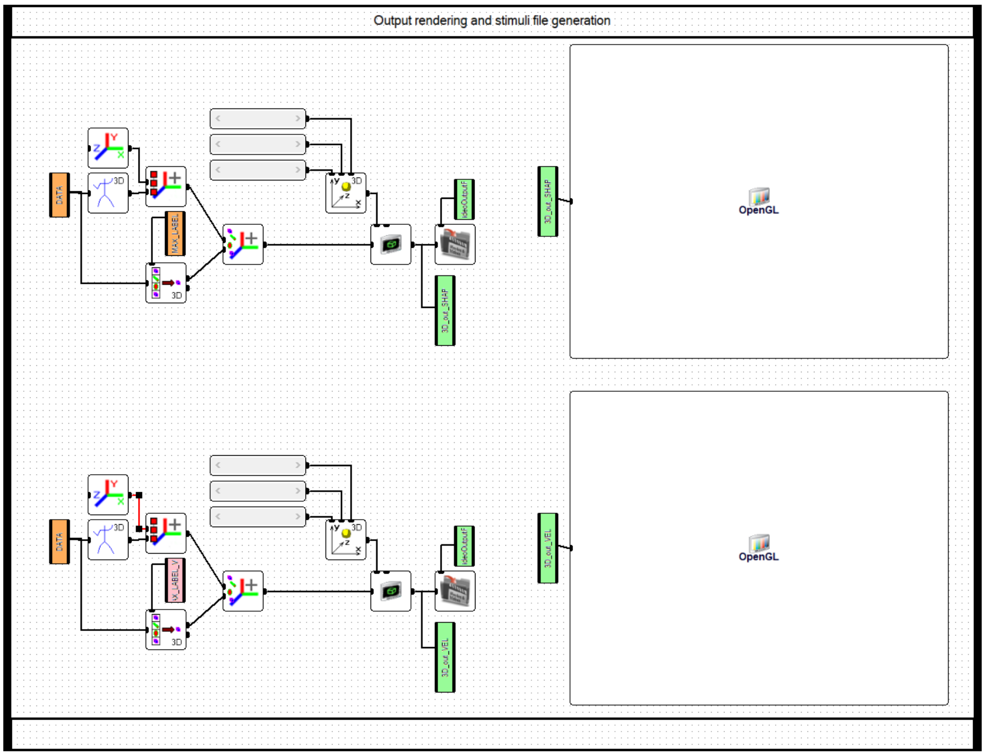

Figure 7.13: Portion of the patch that renders 3D videos of the user's skeleton and saves them to files

7.3 Matlab

7.3.1 General Structure of the Software Module

The proposed method has been implemented in MATLAB as a separate software module. Figure 7.14 presents the main parts of the Matlab software module. In this subsection we describe the steps performed.

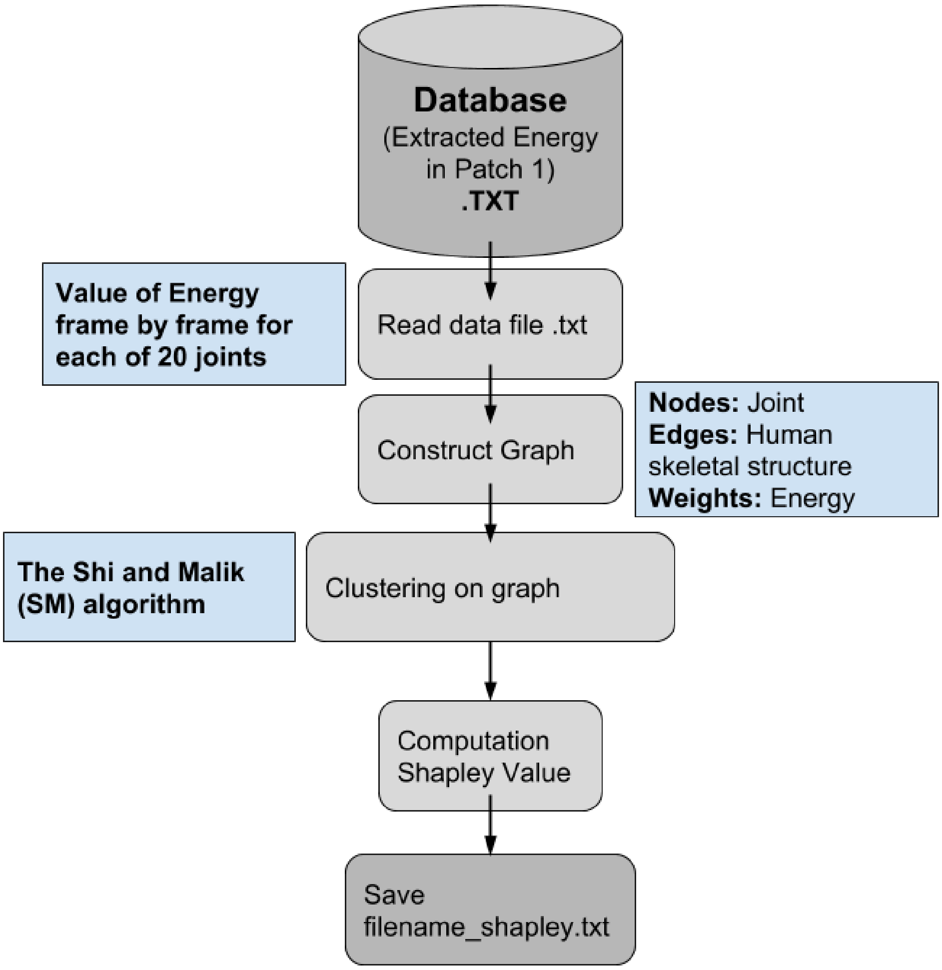

Figure 7.14: Workflow chart of part of the work developed in Matlab

The general plan is the following:

1. Data Loading

   First, we load the data .txt file with the values of velocity that we calculated and extracted previously with the Eyesweb patch. The data file consists of a matrix of velocity values for 20 joints, for each frame of the recording segment.

2. Graph Construction
   After loading the data .txt file with the velocity for each joint, we specify the name of the joint that correspond to each column in the data file.

These 20 joints of the human skeleton represent the nodes of the graph that we construct. After specifying the nodes, we introduce the edges, which represent the connections. We use the loaded data file with the extracted energy in order to specify the weight of the edges of the graph.

3. Spectral Clustering

   Then, we cluster the resulting weighted graph applying spectral clustering to its weighted edge set (see more details in the following section). In such a way, vertices that belong to the same cluster are expected to have similar feature values, whereas edges between different clusters should be associated with vertices having significantly different feature values.

4. Computation of the Shapley value

   Then for each frame, we compute the Shapley values of all the 20 joints, using the method detailed in Chapter 4. The computation of Shapley values is implemented in separate MATLAB function Max_Shapley.m.

5. Saving data to file

   After computing maximum Shapley values, we save the data for each segment of recording into a .txt file with first maximum Shapley value, second maximum Shapley value, and the numbers of the two associated joints, for each frame.

The Graph construction and Spectral Clustering presented in details in the following sections.

7.3.2 Creating the Graph with MATLAB

Briefly, we remind that a graph is a collection of nodes and edges that represents relationships: nodes are vertices that correspond to objects, edges are the connections between objects. The graph edges sometimes have weights, which indicate the strength (or some other attribute) of each connection between the nodes.

In MATLAB, graphs are created using **graph** and **digraph**, and can have self-loops (an edge connecting a node to itself). However, graphs cannot have multiple edges with the same source and target nodes.

The primary ways to create a graph are to use an adjacency matrix or an edge list.

<u>Adjacency Matrix</u>

One way to represent the information in a graph is through a square adjacency matrix. The nonzero entries in an adjacency matrix indicate the presence of an edge between the corresponding two nodes, and the value of the entry indicates the weight of the edge. The diagonal elements of an adjacency matrix are typically zero, but a nonzero diagonal element indicates a self-loop, i.e., a node that is connected to itself by an edge.

The adjacency matrix of an undirected graph created by **graph** must be symmetric. However, in practice the matrices are frequently triangular to avoid repetitions of elements.

For large graphs, the adjacency matrix typically is a sparse matrix, i.e, it contains many zeros.

We can use adjacency matrices to create a graph using the **graph** function, then we can use the **adjacency** function to find the unweighted sparse adjacency matrix of a pre-existing graph.

Edge List

Another way to represent the information in a graph is by listing its edges. For connected graphs, the graph nodes are implied by the list of the graph edges. However, some graphs have disconnected nodes, which need to be listed separately.

In MATLAB, the list of edges is separated by column into source nodes and target nodes. For directed graphs, the edge direction (from source to target) is important, but for undirected graphs the source and target node are interchangeable. One way to construct a graph using the edge list is to use separate inputs for the source nodes, the target nodes, and the edge weights.

By default, all the nodes in a graph created using **graph** or **digraph** are numbered. Thus, it is always possible to refer to the nodes in a graph by their numeric node index.

If the graph has node names (that is, **G.Nodes** contains a variable Name), then one can also refer to the nodes in a graph using their names. Thus, named nodes in a graph can be referred to by either their node indices or node names. For example, node 1 can be called also node, 'A'.

Graph Node IDs

The term node ID encompasses both aspects of node identification. The node ID of a node refers to both the node index and the node name.

For convenience, MATLAB remembers which type of node ID is used in calling most graph functions. So if one refers to the nodes in a graph by their node indices, most graph functions return a numeric answer that also refers to the nodes by their indices.

However, if one refers to the nodes by their names, then most graph functions return an answer that also refers to the nodes by their names (contained in a cell array of character vectors).

We can use the **findnode** function to find the numeric node ID for a given node name. Conversely, for a given numeric node ID, one can use **G.Nodes.Name** to determine the corresponding node name.

After loading the data .txt files with the velocity for each joint, we specify the names of each joint that correspond to the associated columns in data file.

The order of the joints is the following: head - 1, hip_center – 2, left_ankle - 3, left_elbow - 4, left_foot - 5, left_hand - 6, left_hip - 7, left_knee - 8, left_shoulder - 9, left_wrist - 10, right_ankle - 11, right_elbow - 12, right_foot - 13, right_hand - 14, right_hip - 15, right_knee - 16, right_shoulder - 17, right_wrist - 18, shoulder_center - 19, spine - 20

We rexall, that these 20 joints of the human skeleton represent the nodes of the graph that we construct. After specifying the nodes, we introduce the edges, which represent the connections between two nodes. We use the loaded data file with the extracted energy in order to specify the weight of the edges of the graph.

Then we use the function **graph** of MATLAB in order to construct the graph.

7.3.3 Spectral Clustering

Then, we cluster the resulting weighted graph applying spectral clustering to its weighted edge set (see, e.g., [63] for some details on this clustering method, and [68] for the specific implementation). In such a way, vertices that belong to the same cluster are expected to have similar feature values, whereas edges between different clusters should be associated with vertices having significantly different feature values.

Due to the reduced number of joints, we fixed 4 as the maximum number of clusters to be detected by spectral clustering (such a small number of clusters was chosen due to the small number of vertices of the skeletal structure).

Spectral Clustering has become quite popular over the last few years, and several new algorithms have been published. To choose a specific Spectral Clustering algorithm, we used the results of the comparison, made in [68], of several well-known such algorithms from the point of view of clustering quality over artificial and real datasets. In that work, the authors implemented many variations of existing Spectral Clustering algorithms, and compared their performance to see which features are more important. They also demonstrated that Spectral Clustering methods show competitive performance on real datasets with respect to other existing methods.

Based on the results of the experiments reported in [68], we selected the Shi and Malik algorithm to perform the Spectral Clustering on the previously constructed graph.

Notation

The set of data points to be clustered is denoted by I, with $|I| = n$. For each pair of points $i, j \in I$ a similarity $S_{ij} = S_{ji} \geq 0$ is given. The similarities S_{ij} can be viewed as weights on the undirected edges (ij) of a graph G over I. The matrix $S = (S_{ij})$ plays the role of a real-valued adjacency matrix for G. Let $D_i = \sum_{j \in I} S_{ij}$ be the degree of node i, and the volume of a set

$A \subset I$ be $VolA = \sum_{i \in A} D_i$. The set of edges between two disjoint sets A, $B \subseteq I$ is called the edge cut or in short the cut between A, B (donated $Cut(A, B)$). A clustering $C = C_1, C_2, ..., C_K$ is a way of partitioning I into the nonempty mutually disjoint subsets $C_1, ...C_K$.

<u>The Shi and Malik (SM) algorithm</u>

This algorithm was introduced by [56] as a heuristic algorithm aimed to minimize the Normalized Cut criterion proposed by the same authors. The normalized cut between two sets $A, B \subseteq I$ is defined as

$$NCut(A, B) = Cut(A, B)(\frac{1}{VolA} + \frac{1}{VolB})$$

The set I is partitioned into two clusters $C, C' = I \backslash C$ that approximately minimize $NCut(C, C')$ over all possible two way partitions of I.

Algorithm SM

1. Compute
$$P = D^{-1} S$$

2. Let $1 = \lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$ be the eigenvalues of P, and v_1, v_2, . . ., v_n the corresponding eigenvectors (If the eigenvalues are not distinct, we pick the eigenvectors such that $(v^i)^T D v^j = 0$ for $i \neq j$. This is always possible and the Matlab implementation that we used does it automatically.)

   Compute v^2.

3. Min-Ratio-Cut

   (a) Sort the elements of v^2 in increasing order. Denote by v_i^2 the i-th element in the sorted list.

   (b) For $i = 1, ...n - 1$
   Compute $NCut(C_i, C_i')$ where $C = \{1, ...i\}, C' = \{i + 1, i + 2...n\}$

   (c) Partition I into the two clusters C_{i_0}, C_{i_0}' where $i_0 = \text{argmin}_i NCut(C_i, C_i')$

4. Repeat steps 1–3 recursively on the cluster with the largest λ_2 until K clusters are obtained.

Part IV

Future Work and Conclusion

Chapter 8

Evaluation Platform

Contents

This chapter provides ...

8.1 Visualization

In order to simplify the evaluation of the algorithm and make it visually observable, we generated video sequences portraying a skeletal representation of a performer, where at each video frame, one of the joints is highlighted with a red colour, so that the highlighted joint is the joint with the maximum Shapley value among the joints set extracted by the proposed method for that frame.

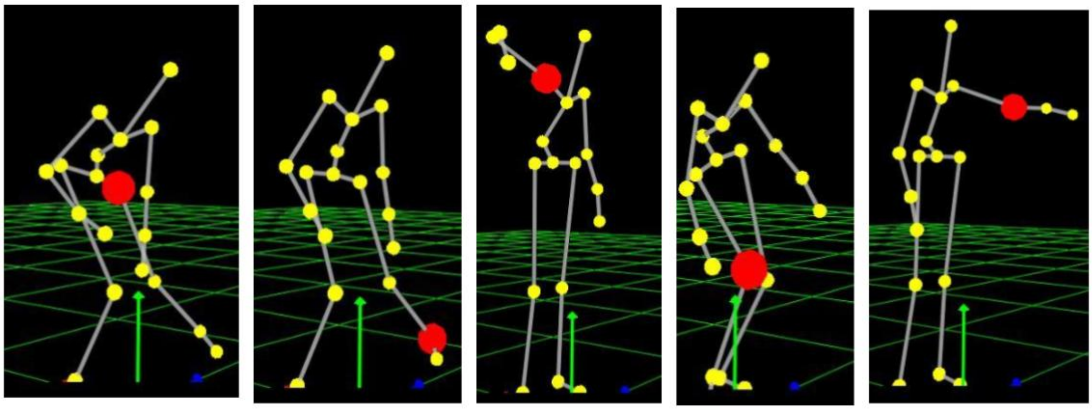

Figure 8.1: These sequence of images represents a dance sequence and the corresponding transitions of the highlighted joint, which represents the joint with the maximum Shapley value at each frame.

Figure 8.1 shows an example of a sequence of frames taken from one of the rendered videos. The videos were then used to develop a survey website described in the following Section.

8.2 Survey Website for Evaluation

In order to evaluate the proposed method, we designed a survey website[1] to collect user ratings on the developed method. As the user visits the page, a description of the task is given (visible in Figure 8.2), then the user can choose to proceed with the task or quit it at any moment. No sensible data is collected during the procedure.

[1] http://www.infomus.org/Tools/OriginOfMovementThreeVideos/index.php

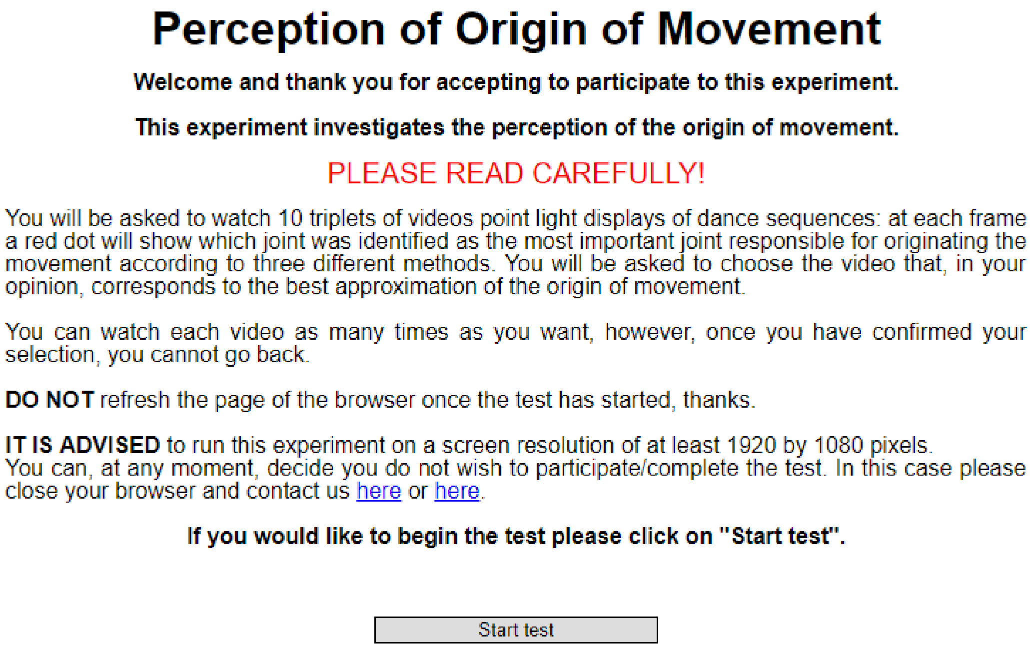

Figure 8.2: Website for the evaluation of method, introductory page

Once the user accepts to participate and perform the task, a series of triplets of videos is proposed (Figure 8.3). Each among the three videos displays a skeletal representation of a dancer performing the same dance sequence. Each video has one highlighted joint (in red). Such a joint corresponds to the most important joint according to one of the following criteria: (i) maximum Shapley value (ii) maximum velocity, and (iii) random choice. The order of the three criteria is randomized among the three videos, so that the specific criterion applied to each video is unpredictable.

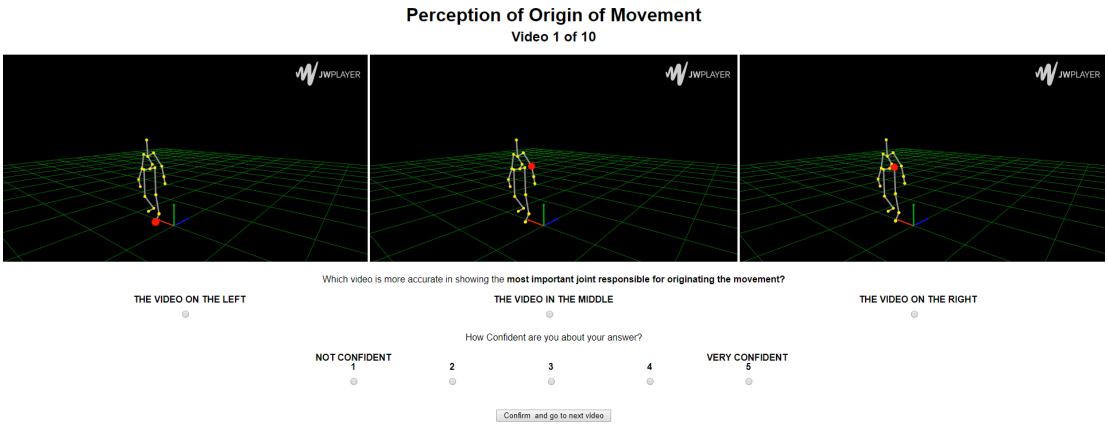

Figure 8.3: Website for the evaluation of the proposed method

The participant is asked to pick the video that mostly represents the evolution of the most important joint responsible for the origination the dancer' movement. Once a user has selected one video, she is also asked to declare how much she is confident about her choice by selecting a value from 1 to 5 on a Likert scale. The participant can see all the videos as many times as desired, and has to answer both questions (video choice and confidence) before proceeding to the next triplet of videos. Each participant has to rate ten triplets of videos proposed from a selection of one hundred triplets, using a Latin square selection method.

Results of the questionnaire are saved on a .csv file, including a unique Id of the participant. The file is generated when the participant accepts to perform the task. For each of the ten proposed triplets of videos, the following information is collected:

- Id of the dance sequence proposed;
- order used to propose the three alternative videos, and corresponding extraction method used (Shapley, velocity, random);
- participant' choice among the three videos;
- participant' confidence.

The website was submitted to dance experts and choreographers, but results are still not ready, due to a low availability of participants. Therefore the evaluation of the methods will performed in the final phase of the H2020 ICT WhoLoDanCe project. The results of the evaluation will be considered for a future submission of a journal article.

Chapter 9

Conclusion

In this thesis, we presented a novel computational method for the analysis of expressive full-body movement qualities, which exploits concepts and tools from graph theory and game theory. We believe that the novel point of view adopted by this method can allow substantial advancements in movement analysis and in understanding the mechanisms of social affective communication, also providing new and effective computational models for that.

Moreover, the method may be applicable to other fields, as well. It is also worth noting that, since in the proposed method the Shapley value reduces to a weighted degree centrality on the auxiliary graph, its computation is feasible not only off-line but also on-line, since for its evaluation it is not necessary to consider all possible coalitions in following formula:

$$\phi_i(c) := \sum_{V' \subseteq V \setminus \{i\}} \frac{|V'|!(|V| - |V'| - 1)!}{|V|!} (c(V' \cup \{i\}) - c(V')) . \tag{9.1}$$

The results of application of proposed method to a dataset of movement segments obtained in the thesis are promising, and show that, on the data set used for its preliminary evaluation, the method was able to extract meaningful joints.

By observing the visualized results on the moving skeleton, we can notice the relevance of the extracted joint with respect to the concept of *"Origin of movement in dance"*, which was defined by dancers and choreographers in the framework of the H2020 Wholodance Project, which aims at developing and applying breakthrough technologies to dance. Origin of movement is the point at which a movement is said to originate. This refers to specific body parts, which can be either distal (in the limbs or head) or central (in the torso).

We observed differences between joints with maximum Shapley value and joints with maximum velocity. In the opinion of dance experts, who participated in discussions during the WholoDance Project, the joint where movement originates is different from the joint with maximum velocity. In general, body parts with maximum velocity showed to be in the extremities of the body: left hand, right hand or left foot, right foot. Instead, the body joints with maximum Shapley value are related to the same body parts, but not in their ends, meaning left shoulder, right shoulder, left knee, right knee etc.

As a future step, for a further validation, we aim to apply the method to features different from speed and to a larger data set, allowing for a statistical evaluation of its results, and for a more quantitative comparison with the annotations. Moreover, the output of the model (location of the most important joint in each frame) could be also used to construct higher-level features to train supervised classifiers.

Bibliography

[1] AARDAL, K. I., VAN HOESEL, S. P., KOSTER, A. M., MANNINO, C., AND SASSANO, A. Models and solution techniques for frequency assignment problems. *Annals of Operations Research 153*, 1 (2007), 79–129.

[2] AGARWAL, N., AND ZEEPHONGSEKUL, P. Psychological pricing in mergers & acquisitions using game theory. *School of Mathematics and Geospatial Sciences, RMIT University, Melbourne* (2011).

[3] AGGARWAL, J. K. Motion analysis: past, present, and future. In *Distributed Video Sensor Networks*, B. Bhanu, C. V. Ravishankar, A. K. Roy-Chowdhury, H. Aghajan, and D. Terzopoulos, Eds. Springer-Verlag, London, UK, 2011, pp. 27–39.

[4] BALABAN, A. T., AND SCHLEYER, P. V. R. Systematic classification and nomenclature of diamond hydrocarbons—i: Graph-theoretical enumeration of polymantanes. *Tetrahedron 34*, 24 (1978), 3599–3609.

[5] BARDADYM, V. A. Computer-aided school and university timetabling: The new wave. In *international conference on the practice and theory of automated timetabling* (1995), Springer, pp. 22–45.

[6] BEN-DAVID, S., BORODIN, A., KARP, R., TARDOS, G., AND WIGDERSON, A. On the power of randomization in on-line algorithms. *Algorithmica 11*, 1 (1994), 2–14.

[7] BERGE, C. *Graphs and Hypergraphs*. North-Holland mathematical library. Amsterdam, 1973.

[8] BERNHARDT, D., AND ROBINSON, P. Detecting affect from non-stylised body motions. In *Affective Computing and Intelligent Interaction*, A. Paiva, R. Prada, and R. Picard, Eds., vol. 4738 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Lisbon, 2007, pp. 59–70.

[9] BOONE, R., C. J. Children's decoding of emotion in expressive body movement: The development of cue attunement. *Developmental psycology 34* (1998), 1007–1016.

[10] BORZESHI, E. Z., XU, R., AND PICCARDI, M. Automatic human action recognition in videos by graph embedding. In *Image Analysis and Processing – ICIAP 2011* (Berlin, Heidelberg, 2011), G. Maino and G. L. Foresti, Eds., Springer Berlin Heidelberg, pp. 19–28.

[11] BOUSKILA, A. Games played by predators and prey.

[12] BOUSMALIS, K., MORENCY, L., AND PANTIC, M. Modeling hidden dynamics of multimodal cues for spontaneous agreement and disagreement recognition. In *2011 IEEE Int.l Conf. on Automatic Face Gesture Recognition Workshops (FG 2011)* (Santa Barbara, 2011), IEEE, pp. 746–752.

[13] BRAMS, S. J. Game theory and the cuban missile crisis. *Plus Magazine: Living Mathematics* (2001).

[14] BROCAS, J., GIELEN, M., WILLEM, R., ET AL. *permutational approach to dynamic stereochemistry*. McGraw-Hill International, 1983.

[15] CAMURRI, A. Recognizing emotion from dance movement: Comparison of spectator recognition and automated techniques. *Int. J. of Human-Computer Studies 59* (2003), 213–225.

[16] CANTER, D. V. *Studies of human behaviour in fire: empirical results and their implications for education and design*. Building Research Establishment, 1985.

[17] CARIDAKIS, G., CASTELLANO, G., KESSOUS, L., RAOUZAIOU, A., MALATESTA, L., ASTERIADIS, S., AND KARPOUZIS, K. Multimodal emotion recognition from expressive faces, body gestures and speech. In *Proc. 4th IFIP Int. Conf. on Artificial Intelligence Applications and Innovations (AIAI2007)* (Athens, 2007), C. Boukis, A. Pnevmatikakis, and L. Polymenakos, Eds., Springer, pp. 375–388.

[18] CASTELLANO, G., VILLALBA, S., AND CAMURRI, A. Affective computing and intelligent interaction. In *Affective Computing and Intelligent Interaction*, A. Paiva, R. Prada, and W. Picard, Eds., vol. 4738 of *Lecture Notes in Computer Science*. Berlin: Springer-Verlag, Lisbon, 2007, pp. 71–82.

[19] CHALKIADAKIS, G., ELKIND, E., AND WOOLDRIDGE, M. *Computational Aspects of Cooperative Game Theory*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2011.

[20] CORMEN, T., LEISERSON, C., RIVEST, R., TORELLI, M., MEREGHETTI, C., AND FABRIZIO, A. *Introduzione agli algoritmi*. Jackson Libri, 1999.

[21] Csikász-Nagy, A., Cavaliere, M., and Sedwards, S. Combining game theory and graph theory to model interactions between cells in the tumor microenvironment. In *New Challenges for Cancer Systems Biomedicine* (Milano, 2012), A. d'Onofrio, P. Cerrai, and A. Gandolfi, Eds., Springer Milan, pp. 3–18.

[22] Cutillo, L. A., Molva, R., and Onen, M. Analysis of privacy in online social networks from the graph theory perspective. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE* (2011), IEEE, pp. 1–5.

[23] Deng, X., and Papadimitriou, C. H. On the complexity of cooperative solution concepts. *Mathematics of Operations Research 19*, 2 (1994), 257–266.

[24] Diestel, R. *Graph Theory*. Springer-Verlag, 2005.

[25] Durlauf, S. N., Blume, L., et al. *The new Palgrave dictionary of economics*, vol. 6. Palgrave Macmillan Basingstoke, 2008.

[26] Erk, K., and Padó, S. A structured vector space model for word meaning in context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (2008), Association for Computational Linguistics, pp. 897–906.

[27] Francis, R. L., and Saunders, P. B. Evacnet: Prototype network optimization models for building evacuation. Tech. rep., 1979.

[28] Freedman, M., Lovász, L., and Schrijver, A. Reflection positivity, rank connectivity, and homomorphism of graphs. *Journal of the American Mathematical Society 20*, 1 (1 2007), 37–51.

[29] Gao, W., Wu, H., Siddiqui, M. K., and Baig, A. Q. Study of biological networks using graph theory. *Saudi Journal of Biological Sciences* (2017).

[30] Hung, H., and Gatica-Perez, D. Estimating cohesion in small groups using audio-visual nonverbal behavior. *IEEE Trans. on Multimedia 12*, 6 (2010), 563–575.

[31] Jeribi, A. *Spectral theory and applications of linear operators and block operator matrices*. 2015.

[32] Kagel, J. H., and Roth, A. E. *The Handbook of Experimental Economics, Volume 2: The Handbook of Experimental Economics*. Princeton university press, 2016.

[33] Kao, J. Y., Ortega, A., and Narayanan, S. S. Graph-based approach for motion capture data representation and analysis. In *2014 IEEE International Conference on Image Processing (ICIP)* (Oct 2014), pp. 2061–2065.

[34] Kranton, R. E., and Minehart, D. F. A theory of buyer-seller networks. *American Economic Review 91*, 3 (2001), 485–508.

[35] KUN, J., POWERS, B., AND REYZIN, L. Anti-coordination games and stable graph colorings. In *Proc. of Algorithmic Game Theory: 6$^{\text{th}}$ Int. Symp., SAGT 2013, Aachen, Germany* (Berlin, Heidelberg, 2013), B. Vöcking, Ed., Springer Berlin Heidelberg, pp. 122–133.

[36] LABAN, R. *Modern Educational Dance*. Mcdonald Evans Ltd, London, 1963.

[37] LABAN, R., AND LAWRENCE, F. *Effort*. Mcdonald Evans Ltd, London, 1947.

[38] LAKENS, D., AND STEL, M. If they move in sync, they must feel in sync: Movement synchrony leads to attribution of rapport and entitativity. *Social Cognition 29*, 1 (2011), 1–14.

[39] LINDFORS, E., ET AL. *Network biology: applications in medicine and biotechnology*. VTT, 2011.

[40] LIU, J. Learning semantic features for visual recognition. Dept. Electr. Eng. Comput. Sci., Univ. Central Florida.

[41] MEIJER, M. The contribution of general features of body movement to the attribution of emotions. *J. of Nonverbal Behaviour 13*, 4 (1989), 247–268.

[42] MYERSON, R. B. Graphs and cooperation in games. *Mathematics of Operations Research 2*, 3 (1977), 225–29.

[43] OZEL, F. The computer model "bgraf": a cognitive approach to emergency egress simulatin.

[44] PANDEY, R. Application of graph theory to translation from english to hindi. *Global Journal of Computational Science and Mathematics 6*, 1 (2016), 19–38.

[45] PASSINI, R. *Wayfinding in architecture*, vol. 4. John Wiley & Sons Inc, 1992.

[46] PETER, C., AND BEALE, R. *Affect and Emotion in Human-Computer Interaction: From Theory to Applications*. Springer-Verlag, Berlin, Heidelberg, 2008.

[47] PROULX, G. A stress model for people facing a fire. *Journal of Environmental Psychology 13*, 2 (1993), 137–147.

[48] PROULX, G. Evacuation time and movement in apartment buildings. *Fire safety journal 24*, 3 (1995), 229–246.

[49] RAJA, K., LAPTEV, I., PÉREZ, P., AND OISEL, L. Joint pose estimation and action recognition in image graphs. In *2011 18th IEEE International Conference on Image Processing* (Sept 2011), pp. 25–28.

[50] RASMUSEN, E., AND BLACKWELL, B. Games and information. *Cambridge, MA 15* (1994).

[51] REHM, M. Non-symbolic gestural interaction for ambient intelligence. In *Human-Centric: Interface for Ambient Intelligence*, H. Aghajan, R.-C. Delgado, and J. Augusto, Eds. ACM Press, Amsterdam, The Netherlands, 2010, pp. 327–345.

[52] SALIH, Y. K., HANG SEE, O., IBRAHIM, R. W., YUSSOF, S., AND IQBAL, A. A novel noncooperative game competing model using generalized simple additive weighting method to perform network selection in heterogeneous wireless networks. *International Journal of Communication Systems 28*, 6 (2015), 1112–1125.

[53] SCHARLEMANN, J. P., ECKEL, C. C., KACELNIK, A., AND WILSON, R. K. The value of a smile: Game theory with a human face. *Journal of Economic Psychology 22*, 5 (2001), 617–640.

[54] SEIDMAN, S. B. Network structure and minimum degree. *Social Networks 5*, 3 (1983), 269 – 287.

[55] SHAPLEY, L. S. A value for n-person games. In *Contributions to the Theory of Games II*, H. W. Kuhn and A. W. Tucker, Eds. Princeton University Press, Princeton, 1953, pp. 307–317.

[56] SHI, J., AND MALIK, J. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence 22*, 8 (2000), 888–905.

[57] SHUBIK, M. *Game theory: Economic applications*. Cowles Foundation for Research in Economics at Yale University, 1968.

[58] SHUBIK, M. Game theory models and methods in political economy. *Handbook of Mathematical Economics 1* (1981), 285–330.

[59] SIME, J., AND KIMURA, M. The timing of escape: Exit choice behavior in fires and building evacuations. *Safety in the built environment* (1988), 55.

[60] SMITH, J. M. The theory of games and the evolution of animal conflicts. *Journal of theoretical biology 47*, 1 (1974), 209–221.

[61] SMITH, J. M., AND HARPER, D. *Animal signals*. Oxford University Press, 2003.

[62] SONDEREGGER, M. Applications of graph theory to an english rhyming corpus. *Computer Speech & Language 25*, 3 (2011), 655–678.

[63] TIJS, S. *Introduction To Game Theory*. Hindustan Book Agency, New Delhi, 2003.

[64] TIROLE, J. *The theory of industrial organization*. MIT press, 1988.

[65] Torresani, L., Kolmogorov, V., and Rother, C. Feature correspondence via graph matching: Models and global optimization. In *Computer Vision – ECCV 2008* (Berlin, Heidelberg, 2008), D. Forsyth, P. Torr, and A. Zisserman, Eds., Springer Berlin Heidelberg, pp. 596–609.

[66] Turocy, T. L. Texas a&m university. *Bernhard von Stengel, London School of Economics "Game Theory" CDAM Research Report (October 2001)* (2001).

[67] Varni, G., Camurri, A., Coletta, P., and Volpe, G. Toward a real-time automated measure of empathy and dominance. In *Int. Conf. on Computational Science and Engineering, 2009. CSE '09* (Vancouver, 2009), vol. 4, IEEE, pp. 843–848.

[68] Verma, D., and Meila, M. A comparison of spectral clustering algorithms. *University of Washington Tech Rep UWCSE030501 1* (2003), 1–18.

[69] Von Neumann, J., and Morgenstern, O. *Theory of games and economic behavior (commemorative edition).* Princeton university press, 2007.

[70] Wallbott, H. Bodily expression of emotion. *European J. of Social Psychology 28* (1998), 879–896.

[71] Wei, G., Vasilakos, A. V., Zheng, Y., and Xiong, N. A game-theoretic method of fair resource allocation for cloud computing services. *The journal of supercomputing 54*, 2 (2010), 252–269.

[72] Young, H. P. The evolution of conventions. *Econometrica 61*, 1 (1993), 57–84.

[73] Zegura, E. W., Calvert, K. L., and Donahoo, M. J. A quantitative comparison of graph-based models for internet topology. *IEEE/ACM Trans. Netw. 5*, 6 (Dec. 1997), 770–783.