Chapter 3

Motion Capture

When working with music and body motion it is essential to be able to convey information about how someone or something moves. In daily speech we use words such as ‘walking’, ‘rolling’, ‘turning’, etc., to achieve this. These words, however, do not provide precise descriptions of motion. More detailed representations of motion can be gained through visualisation techniques, such as a video recording, or through a sequence of photographs, drawings or storyboards [Jensenius, 2007a].

Motion capture (mocap) involves the use of a sensing technology to track and store movement. In principle, a pencil drawing on a piece of paper can be called motion capture, since the pencil lead is testimony of the hand motion of the person that made the drawing. However, the most common use of the term refers to tracking and representation of motion in the digital domain.

3.1 Motion Capture Basics

Figure 3.1 shows how motion capture may be divided into three main parts: (1) sensing the motion, (2) processing the sensor data, and (3) storing the processed data. Together, parts 1 and 2 are referred to as *motion tracking*. Rather than being stored, tracking data may be used directly, for instance in realtime interactive applications. Most commercial implementations of tracking technologies include the option of storing data, and so the terms *motion tracking system* and *motion capture system* are often used interchangeably.

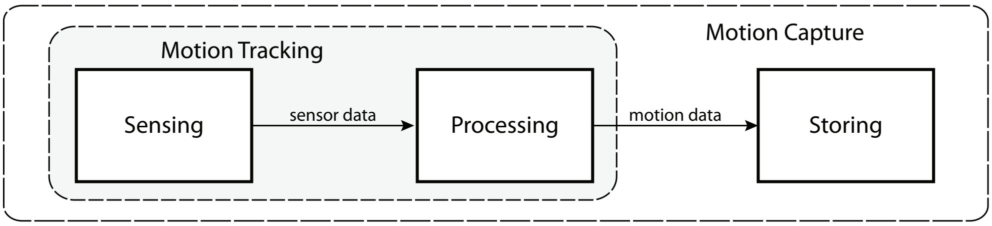


Figure 3.1: Motion tracking involves sensing motion and processing the sensor data. When motion data are stored in order to apply post-processing later, the process is known as *motion capture*.

3.1.1 From Sensor Data to Motion Data

The sensing part of a motion capture system involves measuring some aspect of the motion. This could be done by a large variety of sensors, such as a simple potentiometer or an array of advanced video cameras. In principle, the sensor data can be stored or used directly. However, these data are rarely interesting in themselves, as they typically provide sensor-specific measurements, e.g., resistance in a potentiometer or colour information of camera pixels. Consequently the processing part of a motion capture system translates the raw sensor data into information that describes the motion more significantly, for instance as low-level measures of position or orientation or derivatives of these, such as velocity, acceleration or rotation. Furthermore, certain systems provide motion data specific to the object that is tracked, such as joint angles in a human body.

For positional and orientational measurements the term *degrees of freedom*[1] (DOF) denotes the number of dimensions that are tracked. For instance, 2DOF position would mean the position on a planar surface, and 3DOF position would be the position in three-dimensional space. The description 6DOF is normally used to denote a measurement of an object's three-dimensional position and three-dimensional orientation. 6DOF-tracking is sufficient to represent any position and orientation.

3.1.2 Tracked Objects

Tracking can be applied to point-like objects, such as small spherical *markers*. These are treated as points without volume, and as such only their position (not orientation) can be tracked. A fixed pattern of several markers can be used to identify a *rigid object*. Rigid objects are non-deformable structures whose orientation and position can be tracked. Furthermore, by combining multiple rigid bodies and defining rules for the rotations and translations that can occur between them it is possible to create a *kinematic model*. Such a model may, for instance, represent the human body with the various constraints of the different joints. Such models can even fill in missing data: say, if the data from the lower arm are missing, but the data from the hand and the upper arm are present, the missing data can be estimated by following the kinematic model. Kinematic models might not need position measurements of the different parts: a set of joint angles for the body can be sufficient for a well-defined model. Examples of a marker, a rigid object and a kinematic model are shown in Figure 3.2.

A more formal discussion of how position and orientation can be represented will follow in Section 3.3. First, we shall have a look at the different technologies that are used in motion tracking.

3.2 Motion Tracking Technologies

There is a large variety of motion tracking technologies. The most advanced technologies are capable of tracking motion with very high precision at very high sampling rates. The largest

[1] This should not be confused with the statistical variable *degrees of freedom* (*df*), which is used to denote the size of a tested data set in standardised statistical tests such as t-tests and ANOVAs (see Section 4.2). Furthermore, in biomechanics and robotics degrees of freedom (DOF) is usually used to denote the number of rotary and linear joints in kinematic models [Rosenbaum, 2001, Spong et al., 2006].

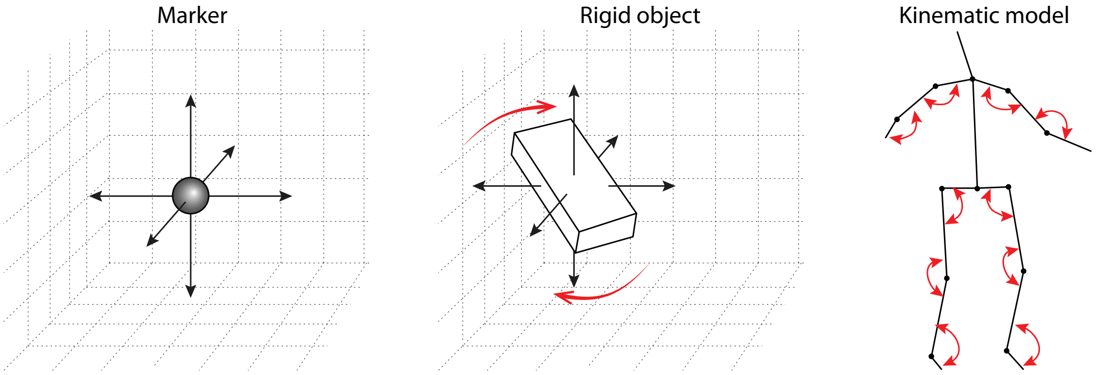


Figure 3.2: The position of a marker can be tracked in three dimensions. A rigid object also allows tracking of orientation. A kinematic model describes the relative position and orientation of connected rigid objects, for instance by joint angles.

appliers of these are the film and gaming industries where they are used for making life-like animations, and researchers who study biomechanics for rehabilitation and sports purposes. At the other end of the scale are ubiquitous low-cost sensor technologies that most people use daily in their mobile phones, laptops, game controllers, and so forth.

This section will give an overview of tracking technologies. The presentation below follows a classification of tracking technologies used by Bishop et al. [2001] where the different systems are sorted according to the physical medium of the technology. The technologies presented in this section include acoustic, mechanical, magnetic, inertial and optical tracking.

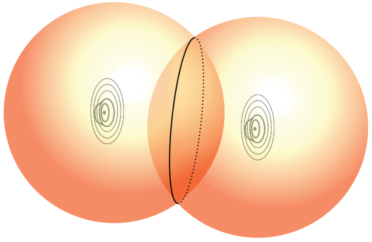
Several aspects of each technology will be presented. A description of the sensor technology as well as the algorithms involved in processing the sensor data constitute the technical details of the technology. Furthermore, the technologies differ in use and should be described in terms of the data they provide to the user, as well as their limitations and advantages in various tracking settings. What is more, in the context of this thesis it is interesting to discuss the use of the technologies in musical settings, such as the study of music-related motion or in interactive music systems.

3.2.1 Acoustic Tracking

Acoustic tracking systems calculate position upon the wavelength of an acoustic signal and the speed of sound. Systems based on *time of flight* measure the time between the sending of a signal from a transmitter and its being picked up by a receiver, and systems based on *phase coherence* measure the phase difference between the signal at the transmitter end and the receiver end [Bishop et al., 2001]. The speed of sound in air at 20 °C is about 343 m/s, but it varies with air pressure and temperature. It may therefore be difficult to acquire precise measurements from acoustic tracking systems. A single transmitter combined with a single receiver gives the distance between the two, or in other words the position of the receiver in a sphere around the transmitter. By adding more transmitters the 3D position of the receiver can be found.[2] Figure 3.3 shows how combined distance measurements from two transmitters narrows the possible positions of the receiver down to a circle.

[2] In addition to tracking the receiver position it is also possible to track the position of the transmitter. In this case adding more receivers would enable finding the 3D position.

Figure 3.3: Distance measurements from two acoustic transmitters can determine the position of a receiver to be somewhere along a circle.



Acoustic systems usually work in the ultrasonic range and can therefore be used in music-related work without interfering with the musical sound. Still, these systems are not widely used in this area. Among the few examples of those using acoustic tracking are Impett [1994], Vogt et al. [2002] and Ciglar [2010], who included ultrasound sensors in the development of digital musical instruments.

3.2.2 Mechanical Tracking

Mechanical tracking systems are typically based on some mechanical construction which measures angles or lengths between the mechanical parts by using bend sensors or potentiometers. These systems can be worn on the body, for instance by implementing sensors in an exoskeleton or a glove, to obtain a model of the joint angles in the whole body or the hand.

There are other implementations of mechanical tracking systems in which the system is not placed on the body but rather contains a base unit placed at a fixed position in the room. Two examples are input devices such as the ‘Phantom Omni’ and the ‘Gametrak’ game controller, sketched in Figure 3.4. The Phantom Omni consists of a movable arm with several joints whose angles are measured by encoders. The Gametrak measures the position of a satellite unit which is attached to the base by a nylon cord. The extension of the nylon cord as well as the angle of the cord are measured, providing positional information for the end of the cord.

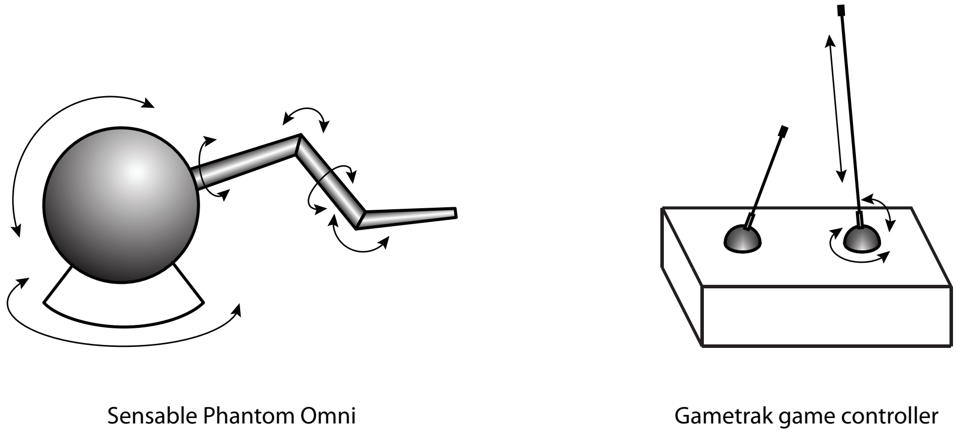


Figure 3.4: Two mechanical motion tracking devices. Left: The Phantom Omni senses the position of the tip of the arm. Right: the Gametrak game controller senses the position of the tip of the nylon cord. The arrows show the measured angles and lengths.

Mechanical tracking has been popular in music-related work, particularly for the purpose of developing new musical interfaces. Various exoskeleton implementations have been developed [e.g., de Laubier, 1998, Jordà, 2002, de Laubier and Goudard, 2006] and also a number of glove-instruments [e.g., Fels and Hinton, 1993, Ip et al., 2005, Hayafuchi and Suzuki, 2008, Fischman, 2011, Mitchell and Heap, 2011]. Furthermore, Zadel et al. [2009] implemented a system for solo laptop musical performance using the Phantom Omni, and Freed et al. [2009] explored a number of musical interaction possibilities for the Gametrak system.

3.2.3 Magnetic Tracking

Magnetic tracking systems use the magnetic field around a sensor. Passive magnetometers can measure the direction and strength of the surrounding magnetic field, the simplest example being a compass which uses the Earth's magnetic field to determine the orientation around the Earth's radial vector. The field varies slightly across the Earth's surface, but this can be compensated for without much effort [Welch and Foxlin, 2002]. Passive magnetometers are widely used in combination with inertial sensors, which will be covered in the next section.

More advanced magnetic systems use an active electromagnetic *source* and a *sensor* with multiple coils. These systems are based on the principle of induction, which explains how an electric current is induced in a coil when it is moved in a magnetic field. To obtain 6DOF tracking a magnetic source with tree coils is used, each perpendicular to the two others [Raab et al., 1979]. Similarly, each sensor consists of three perpendicular coils. The position and orientation of each sensor can be calculated as a function of the strength of the induced signal in each sensor coil [Bishop et al., 2001]. An illustration of the Polhemus Patriot system is shown in Figure 3.5.

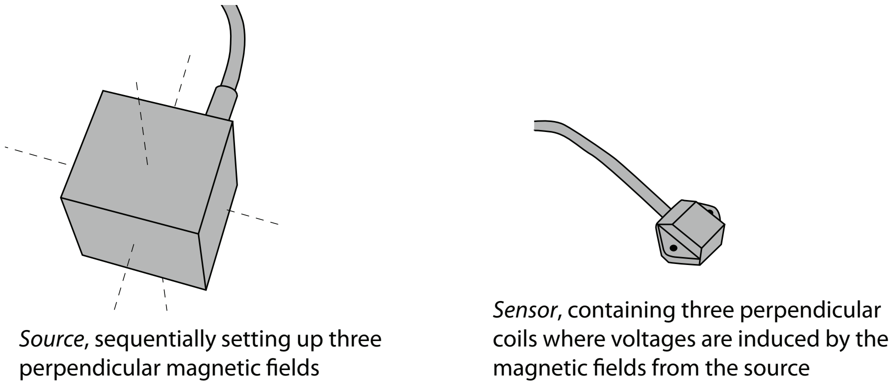

Source, sequentially setting up three perpendicular magnetic fields
Sensor, containing three perpendicular coils where voltages are induced by the magnetic fields from the source

Figure 3.5: The Polhemus Patriot system sets up three perpendicular magnetic fields and tracks the position and orientation of up to two sensors.

Magnetic trackers are able to operate at high sampling rates (more than 200 Hz) with high theoretical accuracy.[3] However, the systems are sensitive to disturbances from ferromagnetic objects in the tracking area. Vigliensoni and Wanderley [2012] showed that the distortion is acceptably low at close distances from the magnetic source. But if a larger area is to be covered, it is necessary to compensate for the distortion of the tracking field [Hagedorn et al., 2007]. This, as concluded by Vigliensoni and Wanderley, may be particularly true for spaces used for musical performance, which often contain ferromagnetic objects. On the positive side, these trackers do not require a clear line-of-sight between the source and the sensor, meaning that the sensors can be hidden under clothes etc.

[3] According to the technical specifications of the Polhemus Liberty system the positional and orientational resolution decrease with increased distance between the source and the sensor. As long as the distance between the sensor and the source is less than 2 m, the system displays submillimeter accuracy [Polhemus Inc.].

Magnetic trackers have been used for analysis of music-related motion by a number of performers and researchers. Trackers from Polhemus have been the most popular, used by e.g. Marrin and Picard [1998], Lin and Wu [2000], Marshall et al. [2002], Ip et al. [2005], Marshall et al. [2006], Maestre et al. [2007] and Jensenius et al. [2008].

3.2.4 Inertial Tracking

Inertial tracking systems include those based on accelerometers and gyroscopes. These sensors are based on the physical principle of *inertia*. Accelerometers measure acceleration based on the displacement of a small "proof-mass" when a force is exerted to the accelerometer. Gravity will contribute to displacement of the proof-mass, and thus the data measured by accelerometers contain the acceleration that is due to gravity (9.8 m/s^2) and any acceleration applied by a user [Bishop et al., 2001]. Gyroscopes apply a similar principle but measure rotational changes. Vibrating parts in the gyroscope resist any torque that is applied to it, and by using vibrating piezoelectric tuning forks in the gyroscopes an electrical signal is emitted when torque is applied [Bishop et al., 2001]. To obtain 6DOF tracking three accelerometers and three gyroscopes are used, with each sensor mounted perpendicularly to the other two.

Inertial tracking systems have certain strong advantages over all the other tracking technologies. Firstly, they are completely self-contained, meaning that they do not rely on external sources such as acoustic ultrasound sensors or cameras which require line-of-sight. Secondly, the sensors rely on physical laws that are not affected by external factors such as ferromagnetic objects or light conditions. Thirdly, the sensors are very small and lightweight, meaning that they are very useful in portable devices; and finally, the systems have low latencies and can be sampled at very high sampling rates [Welch and Foxlin, 2002].

Orientation is gained from inertial tracking systems by integrating the data from the gyroscopes. Any change in orientation also means a change in the direction of the gravity force vector. Position is calculated by first adjusting for any change in the gravity vector, and then integrating the accelerometer data twice [Bishop et al., 2001].

Estimating position from accelerometer data leads us to the downside of inertial sensors; namely *drift*. Even a minor error in data from the gyroscope or the accelerometer will cause a large error in positional estimates. As noted by Welch and Foxlin [2002], a fixed error of 1 milliradian in one of the gyroscopes would cause a gravity compensation error of 0.0098 m/s^2, which after 30 seconds would mean a positional drift of 4.5 metres. For this reason, Welch and Foxlin [2002] conclude, inertial systems work best when combined with other technologies.

Figure 3.6 shows one example of combining inertial sensors with other technologies, namely the Xsens MVN suit [Roetenberg et al., 2009]. The suit uses 17 sensors called *MTx*, fixed at predefined positions on the suit, each containing an accelerometer, a gyroscope and a magnetometer (compass). By combining the sensor signals with a kinematic model, which restricts the positions and orientations of each body segment in relation to the other segments, a full-body model is constructed.

The Xsens MVN suit has been tested and evaluated for use in musical interaction by Skogstad et al. [2011], and actual implementations of the suit in musical interactive systems have been presented by Maes et al. [2010], de Quay et al. [2011] and Skogstad et al. [2012c].

Accelerometers and gyroscopes are now implemented in smart phones and laptops every-

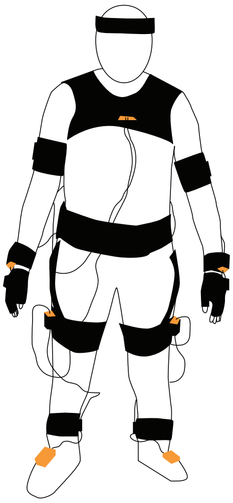


Figure 3.6: The Xsens suit consists of 17 MTx sensors combining inertial sensors and magnetometers. Full body motion capture is obtained through the use of a kinematic model.

where, and the use of inertial sensors in musical performance and research is widespread. This can be seen from the number of laptop orchestras and mobile phone ensembles that have appeared in the recent years [e.g., Trueman et al., 2006, Dannenberg et al., 2007, Wang et al., 2008, Bukvic et al., 2010, Oh et al., 2010].

3.2.5 Optical Tracking

Optical motion tracking systems are based on video cameras and computer vision algorithms. The systems of this type range more widely than do the other types in terms of quality and cost, and various implementations of optical tracking technologies can appear very different to the user.

Optical Sensing

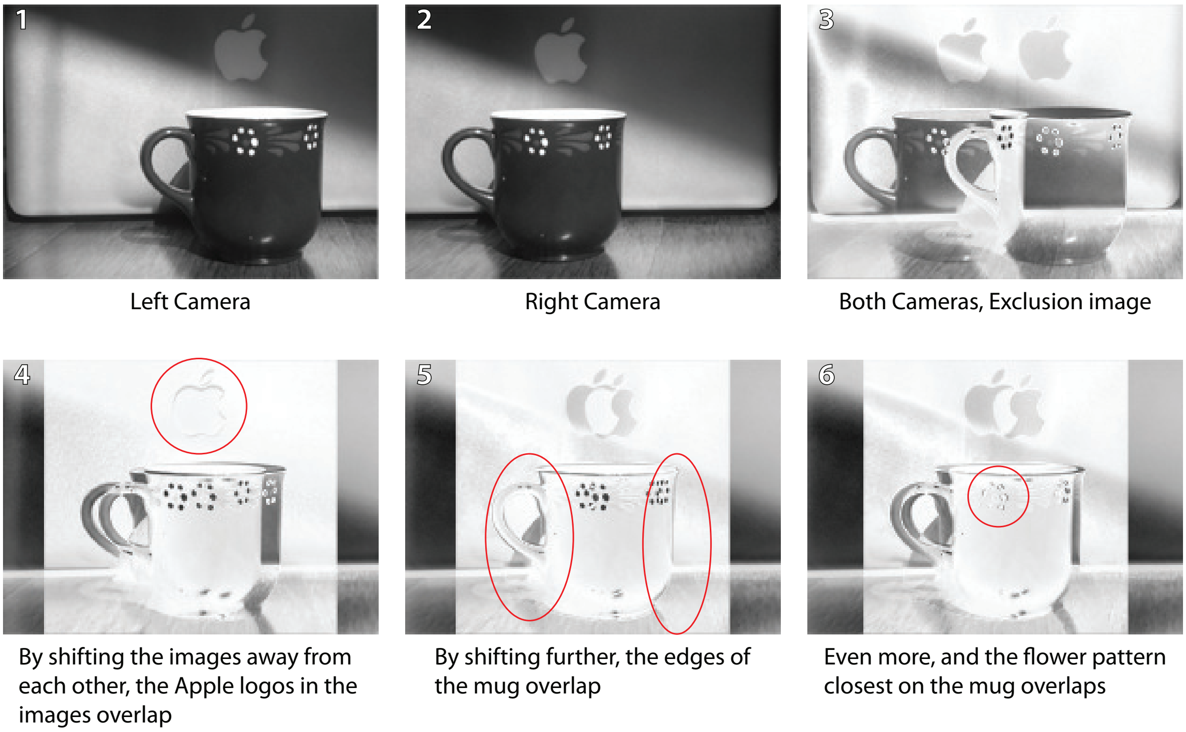
Various types of video camera are used in optical motion tracking. In principle, any digital video camera can be used — in fact, one of the most affordable sensors for conducting motion tracking is a simple web camera. Cameras used in optical motion tracking are either (1) regular video cameras, (2) infrared (IR) video cameras, or (3) depth cameras.

Ordinary video cameras sense light in the visible part of the electromagnetic spectrum. Each pixel in the camera image contains a value corresponding to the amount of light sensed in that particular part of the image. Colour information in each pixel can be represented by using multiple video planes, with the pixel values in each plane representing e.g. the levels of red, green and blue.

Infrared cameras sense light in the infrared part of the electromagnetic spectrum, meaning light with wavelengths above those visible to humans. Some infrared cameras can capture heat radiation, e.g., from humans, but the most common use of infrared cameras in tracking technologies is in a slightly higher frequency range. This is achieved by using some active infrared light source, and either capturing the light from this source directly or as reflections on the tracked objects. Typical implementations consist of a group of infrared light-emitting diodes (LEDs) positioned near the infrared camera and capturing the reflection of this light as it is reflected from small spherical markers.

Depth cameras provide a layer of depth information in addition to the regular two-dimensional image. These cameras use some technology in addition to the regular video camera. One approach is *time-of-flight* cameras, which embed an infrared emitter whose light is reflected off the objects in the field of view. The distance to each pixel is calculated on the speed of light, i.e. the infrared light returns sooner in the case of objects that are closer [Iddan and Yahav, 2001, Ringbeck, 2007]. Another approach, as used in Microsoft's Kinect sensor, is to project a fixed pattern of infrared light and analyse the deformation of this pattern as it is reflected on objects at different distances from the sensor [Freedman et al., 2010].

When not provided by the camera itself depth information can be gained through the use of *stereo cameras*. This involves two cameras mounted next to each other, providing two similar images as shown in Figure 3.7. The figure shows how depth information is found as a correlation function of sideways shifting of the images. The more shift that is required for maximum correlation, the closer to the camera are the pixels in the image. For more details on stereo vision techniques, please refer to [Siegwart and Nourbakhsh, 2004].



Left Camera

Right Camera

Both Cameras, Exclusion image

By shifting the images away from each other, the Apple logos in the images overlap

By shifting further, the edges of the mug overlap

Even more, and the flower pattern closest on the mug overlaps

Figure 3.7: Basic illustration of depth extraction from stereo vision

Computer Vision

After obtaining the video data various processing is applied to the video stream. The video processing that is performed in optical tracking systems is primarily dependent on two factors: (1) whether or not the tracking is based on markers and (2) the camera configuration. But in any case the first processing step is to remove unwanted information from the video, i.e. separate the foreground from the background.

When depth information is available the foreground can be isolated by thresholding the depth values, or if we know the colour of the tracked objects, thresholds can be set on the colour values for each pixel. Other techniques include *background subtraction*, i.e. using a prerecorded background image as reference and detecting any new objects in the image by subtracting the background image from the current image, and *frame difference*, meaning subtracting the previous video frame from the current video frame in order to observe changes in the video image. After the first segmentation step, filtering can be applied and a blob-size[4] threshold can be set in order to remove noise and constrain the tracking to objects of a certain size.

It is useful to distinguish between optical tracking systems that use markers and those that do not. *Markerless* tracking involves tracking whatever is present in the field of view of the camera, e.g. a human body or some object being moved around. The blobs that are detected can be measured in terms of size, centroid, principal axis etc., and these measures can again be matched to some predefined model such as that of a human body, in order to obtain more useful tracking data.

Marker-based tracking technology locates the position of usually spherical or hemispherical markers which can be placed at points of interest. For instance, a human arm can be captured by placing markers on the shoulder, elbow and wrist, or full-body motion tracking can be performed by using larger marker-setups such as Vicon's Plug-in Gait model. Types of marker include *active* light/IR-emitters and *passive* reflective markers which reflect light from an external source. In the case of passive markers the external light sources are typically infrared LEDs mounted around the camera lens.

In marker-based tracking each camera in the system produces a 2D black image with white pixels where markers are observed. This allows efficient separation of the markers from the background by thresholding the pixel values. Furthermore, the markers are treated as points, meaning that only the centroid of each blob is of interest. All in all, this makes the processing of video in marker-based systems quite efficient.

The use of a single camera can provide 2D tracking, or in the case of depth-cameras pseudo-3D tracking — meaning that objects that are hidden behind others in the camera's field of view are not tracked. By using more cameras positioned around the tracked objects full 3D tracking can be obtained. The tracking system is calibrated in order to determine the position and orientation of each camera, usually by moving a calibration wand, meaning a rigid structure with a predefined set of markers attached, around in the tracking area. From the points that are captured simultaneously in multiple cameras the position and orientation of each camera are calculated using so-called *direct linear transformation* [Robertson et al., 2004]. Figure 3.8 shows how the 3D-positions of markers that are seen by multiple cameras can be calculated.

Music-Related Applications

Several systems have been developed for conducting markerless motion capture aimed at music research and musical performance, such as EyesWeb [Camurri et al., 2000], The Musical Gestures Toolbox [Jensenius et al., 2005], and the cv.jit library for Max [Pelletier]. Max objects have also been developed to estimate periodicity in a video image [Guedes, 2006] and create a skeleton model based on video input [Baltazar et al., 2010]. For analysis of marker-based motion capture data Toiviainen's *MoCap Toolbox* is very useful [Toiviainen and Burger, 2011]

[4] A blob is a group of adjacent pixels in an image matching some criterion. In this case the pixels in the blob would match the criterion of having colour values within a certain range.

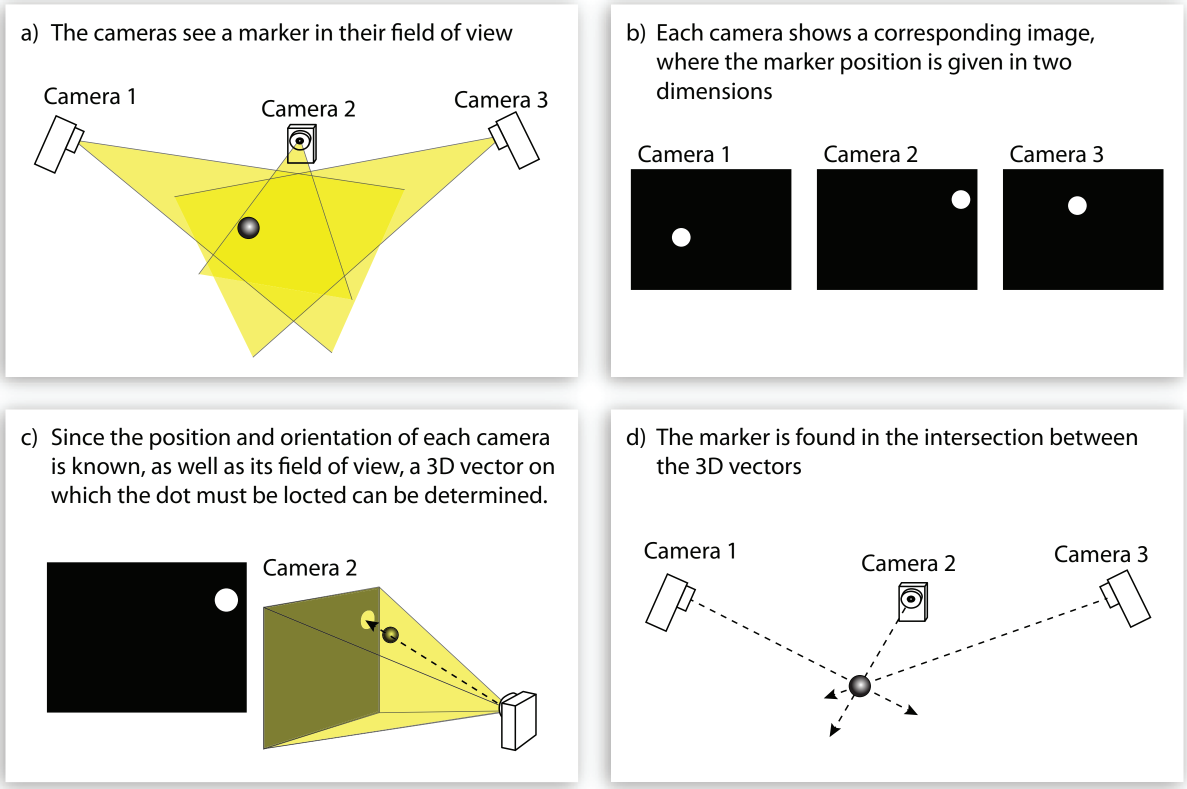

a) The cameras see a marker in their field of view
Camera 1
Camera 2
Camera 3
b) Each camera shows a corresponding image, where the marker position is given in two dimensions
Camera 1
Camera 2
Camera 3
c) Since the position and orientation of each camera is known, as well as its field of view, a 3D vector on which the dot must be locted can be determined.
Camera 2
d) The marker is found in the intersection between the 3D vectors
Camera 1
Camera 2
Camera 3

Figure 3.8: Illustration of how 3D marker positions can be calculated by an optical marker-based system.

and includes methods of feature extraction and visualisation which will be further presented in Sections 3.5 and 4.1.

Optical tracking has been popular in analysis of music-related motion. Sofia Dahl [2000, 2004] and later Bouënard et al. [2008] used marker-based motion capture of drummers to observe details of accents in percussion performance. Furthermore, Marcelo M. Wanderley and others studied how musical performance of clarinettists was perceived in different movement conditions [Wanderley, 2002, Wanderley et al., 2005, Nusseck and Wanderley, 2009]. Marker-based motion capture has also been applied in studies of string performance [Ng et al., 2007, Rasamimanana et al., 2009, Schoonderwaldt and Demoucron, 2009] and piano performance [Godøy et al., 2010, Thompson and Luck, 2012]. There are also several examples of the use of optical motion capture to analyse the motion of listeners and dancers [e.g., Camurri et al., 2000, 2003, 2004, Jensenius, 2007a, Leman and Naveda, 2010, Luck et al., 2010a, Toiviainen et al., 2010, Burger et al., 2012, Jensenius and Bjerkestrand, 2012].

The use of optical tracking in musical performance has also been explored. Various frameworks and guidelines for sonification of tracking data have been presented by Bevilacqua et al. [2002], Dobrian and Bevilacqua [2003], Wanderley and Depalle [2004], Kapur et al. [2005], Verfaille et al. [2006], Koerselman et al. [2007], Eckel and Pirro [2009], Grond et al. [2010], Skogstad et al. [2010] and Jensenius [2012c]. Furthermore, several implementations of optical tracking in sound installations or interactive music systems have been presented, e.g., by Leslie et al. [2010], Yoo et al. [2011], Bekkedal [2012], Sentürk et al. [2012] and Trail et al. [2012].

3.3 Tracking Data

Before discussing methods of working with tracking data, I shall briefly present some details of position and orientation representation.

3.3.1 Coordinate Systems

The data obtained from tracking systems constitute either a description of the tracked object in relation to some external reference point or in relation to its own previous state. In many cases the reference used is a *global coordinate system*[5] (GCS) which can sometimes be defined by the user during the calibration of the tracking system, or determined by the position of some hardware, such as a camera or an electromagnetic source [Robertson et al., 2004].

Rigid objects can be assigned a *local coordinate system* (LCS) as shown in Figure 3.9. The LCS is fixed on the object and the axes of the LCS follow the object when it is translated and rotated in space. As will be explained below the orientation of the rigid object can be measured as the orientation of the LCS in relation to the GCS. Similarly, joint angles in a kinematic model are given as the orientation of one rigid object relative to another.

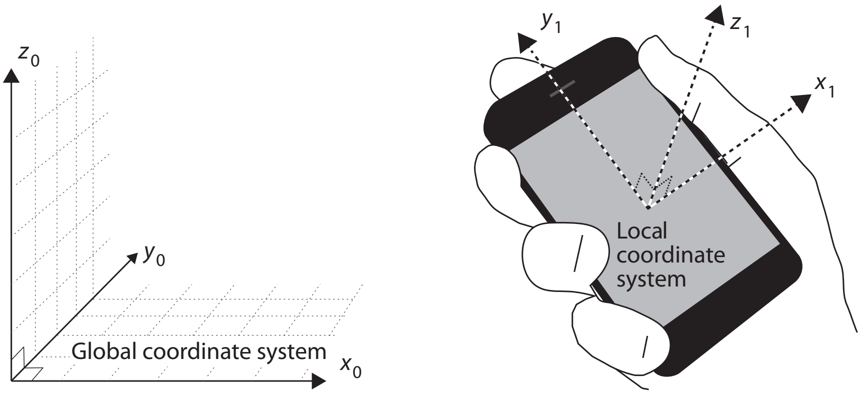


Figure 3.9: A global coordinate system (GCS) is often defined during calibration. Position and orientation measurements are given in relation to the GCS as the position and orientation of a local coordinate system (LCS) with respect to the GCS.

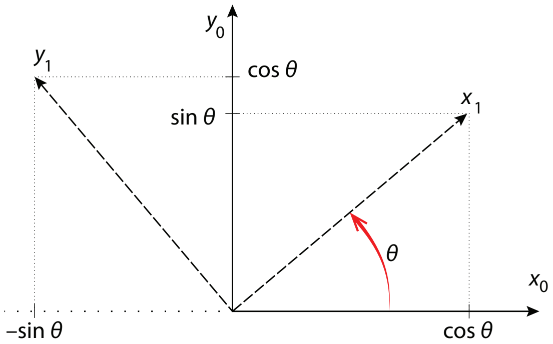
If no global coordinate system is defined, but a local coordinate system exists, the current position and orientation can be reported by reference to the previous position and orientation in a local coordinate system. In principle this also enables definition of a pseudo-global coordinate system at the start of the tracking, and estimation of trajectories in relation to this. However, as mentioned above in the section on inertial sensors, such systems are often sensitive to *drift*, which means that the error in the estimated position and orientation will increase over time.

3.3.2 Representing Orientation

We can find the position of a rigid object by the coordinates of the origin of the LCS in the GCS. Similarly, we can find the orientation of the rigid object by looking at the orientation of the axes of the LCS compared with the axes of the GCS. Figure 3.10 shows how the elements

[5] Also called a *laboratory coordinate system*, *Newtonian frame of reference*, or *absolute reference system*.

of a 2D *rotation matrix*[6] are found by projecting the axes of the LCS ($x_1 y_1$) onto the axes of the GCS ($x_0 y_0$): When the orientation is of the angle θ, the projection of the x-axis of the LCS is at point $(\cos\theta, \sin\theta)$ in the GCS, and the projection of the y-axis is at $(-\sin\theta, \cos\theta)$.



$$R_1^0 = \begin{bmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

Figure 3.10: 2D (planar) rotation. The rotation from coordinate system 0 to coordinate system 1 (written R_1^0) is found by projecting the axes of system 1 onto system 0. The notation on the right shows how this is written as a *rotation matrix*.

In case of a 3D rotation a 3 × 3 rotation matrix is used. As for the 2D rotation, the rotation matrix is found by projecting the axes of the new coordinate system onto the original system. Figure 3.11 shows how the rotation matrix is found for a rotation of θ around the z_0 axis, followed by a rotation of ψ around the x_1 axis. The rotation matrix for the first rotation (R_1^0), is found by projecting the axes x_1, y_1, z_1 onto x_0, y_0, z_0:

$$R_1^0 = \begin{bmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 & z_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 & z_1 \cdot x_0 \\ x_1 \cdot z_0 & y_1 \cdot z_0 & z_1 \cdot z_0 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and similarly R_2^1, describing the second rotation, is:

$$R_2^1 = \begin{bmatrix} x_2 \cdot x_1 & y_2 \cdot x_1 & z_2 \cdot x_1 \\ x_2 \cdot y_1 & y_2 \cdot y_1 & z_2 \cdot y_1 \\ x_2 \cdot z_1 & y_2 \cdot z_1 & z_2 \cdot z_1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & -\sin\psi \\ 0 & \sin\psi & \cos\psi \end{bmatrix}$$

Finally, the rotation matrix R_2^0, denoting a rotation from the initial state to the final state can be found by multiplying the two first rotation matrices:

$$R_2^0 = R_1^0 R_2^1 = \begin{bmatrix} \cos\theta & -\sin\theta\cos\psi & \sin\theta\sin\psi \\ \sin\theta & \cos\theta\cos\psi & -\sin\theta\sin\psi \\ 0 & \sin\psi & \cos\psi \end{bmatrix}$$

Any rotation can be represented by performing three sequential rotations around one axis of the coordinate system in this manner. This is the basis for representing orientation by *Euler angles*, where three angles are used. Euler angles require a specification of axes about which the rotations revolve. For instance, *ZYZ* Euler angles (θ, ψ, ϕ) refer to a rotation of θ around the z-axis, followed by a rotation ψ around the y-axis and a rotation ϕ around the z-axis.

[6] A rotation matrix can also be referred to as Direction Cosine Matrix (DCM) or Orientation Matrix.

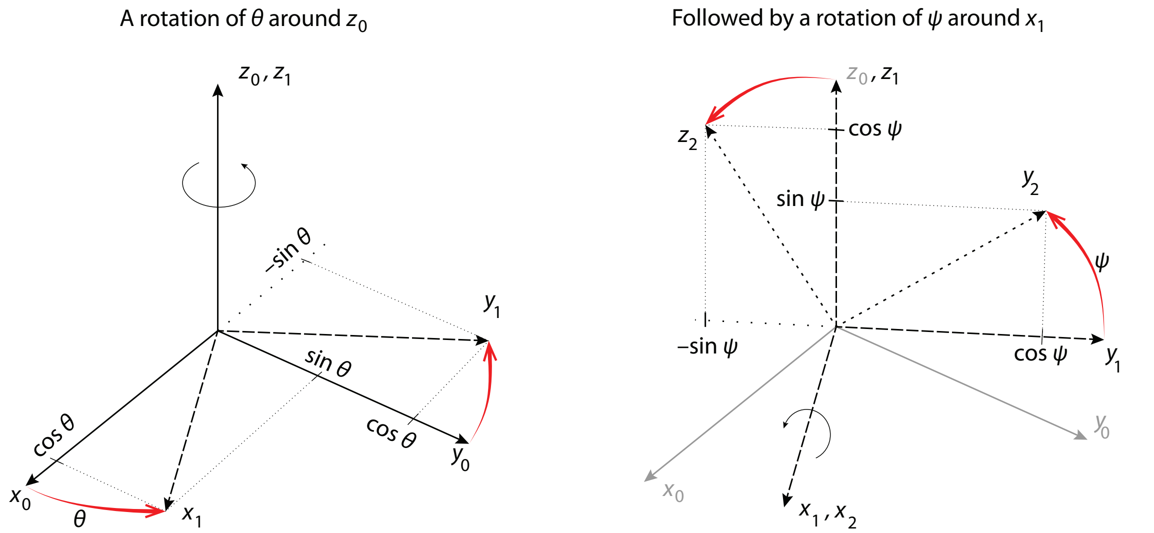


Figure 3.11: 3D rotation made up from two sequential rotations around one axis of the coordinate system. The final rotation matrix R_2^0 is found by multiplying R_1^0 and R_2^1. Any 3D rotation can be represented by three sequential rotations in this manner.

For more details of coordinate systems, representations of orientation, and working with kinematic models, please refer to [Robertson et al., 2004] and [Spong et al., 2006].

3.4 Post-Processing

3.4.1 Tracking Performance

The quality of tracking data provided by the different systems never affords a perfect representation of the real motion. As with all digital data their spatial and temporal resolutions are not infinite and depend on a number of factors related to computational power and limitations in the sensor technology. In addition to the research included in this thesis, Vigliensoni and Wanderley [2012] and Jensenius et al. [2012] have compared motion tracking systems and evaluated their use in musical interaction by measuring accuracy, precision and the temporal stability of the data rate.

The spatial resolution depends on a digitization of a continuous phenomenon. To use a familiar example, a video camera is limited by the number of subdivisions that are measured for the image, i.e. the number of pixels. Furthermore, minor errors in the calibration process can severely affect the spatial resolution [Jensenius et al., 2012]. Also, external factors such as ferromagnetic objects causing disturbance to magnetic trackers can influence the measurements.

The spatial accuracy and precision of tracking systems can be assessed by looking at *noise* and *drift*. Both can be calculated from a static measurement over a period of time. A simple linear regression can be applied to obtain an estimate of a static drift in the system. Or, if the drift is not constant, a better estimate may be obtained by filtering and downsampling the data and observing the extent of change in the data per timeframe.

The level of noise can be measured by the standard deviation (SD) of a static (i.e. without motion) measurement over a time period. If multiple dimensions are tracked, the vector norm of the SDs for each dimension is used. This value is equivalent to the root mean square (RMS) of the distance from the mean position. One example is given in Figure 3.12 where the calculated noise level is equal to the radius of the circle. Jensenius et al. [2012] also suggested other measures for noise level, including the total spatial range covered and the cumulative distance travelled by a static marker.

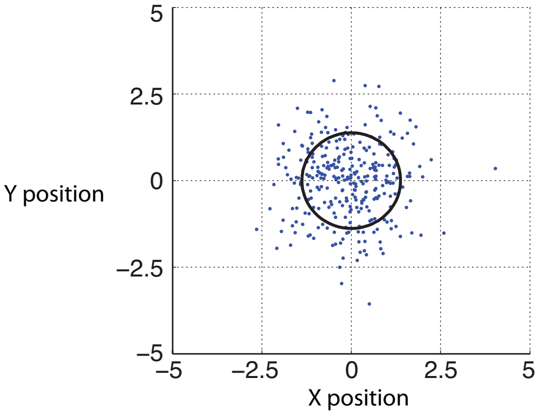


Figure 3.12: Illustration of how noise can be calculated as the standard deviation of a static position recording. The individual dots display 300 position samples (randomly generated for this example), and the circle has a radius equal to the standard deviation of the position samples.

Time is another important performance measure of tracking systems. The systems usually operate at a fixed sampling rate, ranging from a few frames per second up to several thousand frames per second for certain systems [Welch and Foxlin, 2002]. Varying amounts of processing are needed for each timeframe. This processing takes time and thus limits the sampling rate. There may also be time limitations in the sensor technology, such as a regular video camera working in low light conditions, which needs increased shutter time to capture each image.

When tracking data are to be used in real time, temporal stability is important. This is mainly evaluated by *latency* and *jitter*, which in the development of musical interfaces must be kept to a minimum to give the impression of a direct link between the motion and sound [Wessel and Wright, 2002]. The latency of an interactive system is the time delay from when a control action occurs until the system responds with some feedback, for instance the time from when a synthesiser key is pressed until sound is heard. In realtime tracking, latency will increase when processing such as filtering and feature extraction is applied. Any network connection used to stream data between devices will also induce latency. Jitter means any temporal instability in the time interval between data frames. In other words, absence of jitter would mean that the data samples are perfectly periodic.

3.4.2 Gap-Filling

Motion capture recordings may contain gaps, meaning missing frames in the data. This is mostly the case with optical systems, where a marker can be occluded by an arm or moved out of the tracking volume, but can also occur with other systems due, for instance, to packet drops when data are sent over a network.

Gaps in the data can be *gap-filled* by *interpolating* between two points, or by *extrapolating* from a single point if the missing data are at the beginning or end of the recording. Interpolation and extrapolation are achieved by calculating data values at the missing frames from a function where the measured data are used as input. Three interpolation techniques are shown in Figure 3.13. Gap-filling is useful for short gaps, but for longer gaps the trajectory within the gap may not be possible to estimate mathematically. Such recordings must be treated as incomplete and must sometimes be removed from the dataset.

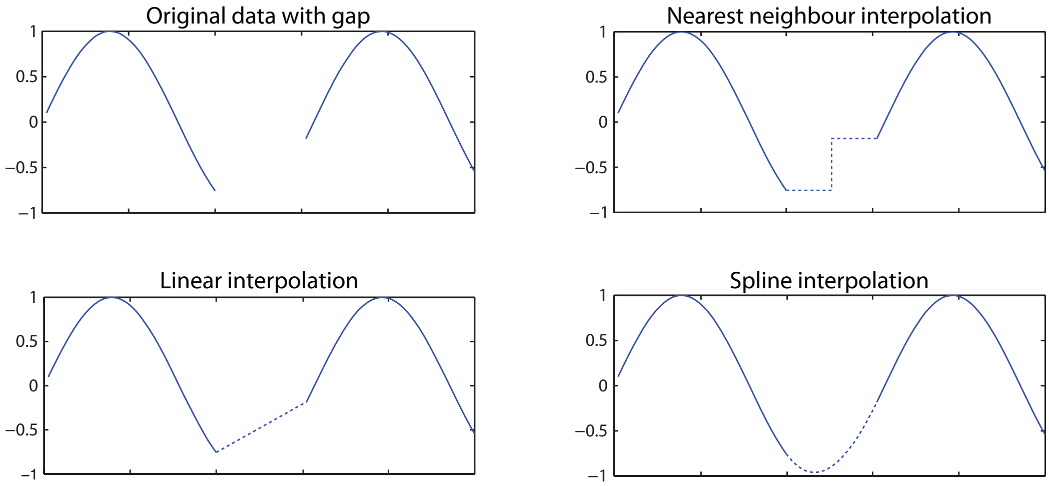


Figure 3.13: Three techniques for gap-filling: nearest neighbour, linear and spline.

3.4.3 Smoothing

Smoothing can be performed by a *moving average* or by more sophisticated *digital filters*. The moving average filter has the advantage of being easy to implement, but it may sometimes attenuate desired signal information and leave unwanted parts of the signal unchanged. The M-point moving average filter is implemented by averaging the past M samples:

$$y_i = \frac{1}{M} \sum_{k=0}^{M-1} x_{i-k}$$

where y_i is the filtered output signal at time i, x is the unfiltered input signal, and M is the number of points for which the moving average is calculated [Smith, 1997].

Better and faster smoothing can be obtained by using more advanced digital filters [Robertson et al., 2004]. Low-pass filters are used to attenuate unwanted noise in the high-frequency range of the spectrum, above the so-called *cut-off frequency*. The frequency band above the cut-off frequency is called *stopband*, and the region below this frequency is called *passband*. The cut-off is never absolute, meaning that there is a *transition band* between the stopband and passband, as shown in Figure 3.14.

Finite impulse-response (FIR) filters implement separate weights (coefficients) for each of the samples in an M-point input signal.

$$y_i = \sum_{k=0}^{M-1} a_k x_{i-k}$$

where a contains the coefficients for weighting the last M samples of x. Moving average filters are a special case of FIR filters, where all coefficients are equal to $1/M$.

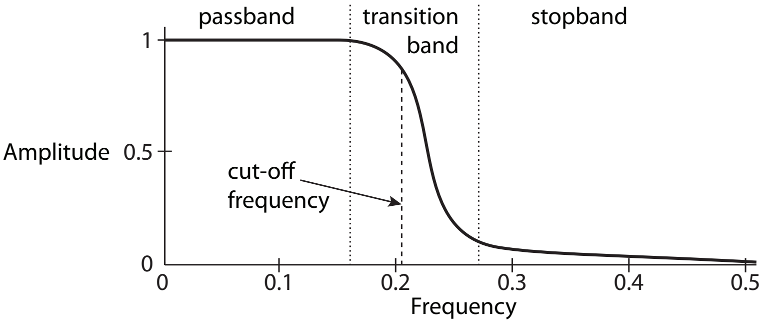


Figure 3.14: The passband, transition band, cut-off frequency and stopband of a digital low-pass filter.

In contrast to FIR filters, *infinite impulse response* (IIR) filters also include weighted versions of the filter output in the calculation. An IIR filter that considers M input samples and N output samples is given by

$$y_i = \sum_{k=0}^{M-1} a_k x_{i-k} + \sum_{k=1}^{N} b_k y_{i-k}$$

where b contains the coefficients for the last N samples of y [Smith, 1997]. IIR filters generally produce narrower transition bands but induce phase distortion, meaning that different parts of the frequency spectrum pass through the filter at different rates. Several standardised filter designs exist, and Matlab-functions for determining the filter coefficients of these are available.[7]

3.5 Feature Extraction

As presented above, there are considerable differences between tracking technologies. Nevertheless, many of the same techniques can be applied to data from different systems. As with the sound features described in Section 2.1, motion features are calculated to obtain more useful information from the raw motion data provided by the tracking system.

The use scenario of the motion data determines the preprocessing and feature extraction that can be applied to motion data. Specifically, when motion tracking is applied to interactive systems where the motion data are used in real time, it is usually important to keep the latency as low as possible. Some processing techniques require a buffering of the signal which induces latency, so trade-offs must often be made between advanced feature extraction algorithms and the amount of latency.

3.5.1 Differentiation

By using basic calculus techniques *velocity* and *acceleration* can be determined from a stream of position data. These are examples of the most basic feature extraction methods for motion data. The simplest way of estimating velocity from position data is to calculate the difference between the current and previous positions (known as the *first finite difference*), multiplied by the sampling rate:

$$v_i = \frac{s_i - s_{i-1}}{\Delta t}$$

[7] e.g. the Matlab functions fir1, fir2, butter, cheby1, cheby2, and ellip

where v_i is the velocity at time i and s is the position in metres. Δt is the time between successive samples (in seconds), and is found by 1/*f* where *f* is the sampling rate in Hz [Robertson et al., 2004]. More accurate calculations can be obtained by the *central difference* method; however, this induces one more sample delay, which could be undesirable in realtime applications:

$$v_i = \frac{s_{i+1} - s_{i-1}}{2\Delta t}$$

A similar calculation can be made to estimate acceleration from velocity data, and jerk from acceleration data. Such differentiations amplify noise that is present in the signal and therefore data smoothing should be applied before the derivatives are calculated.

3.5.2 Transformations

A stream of position data or its derivatives can be transformed in various ways. By projecting data onto new coordinate systems we can obtain information on relations between tracked objects. The position of a person's hand can, for instance, be projected onto a local coordinate system with the centre in the person's pelvis. This would provide information of the position of the hand relative to the body, independently of whether the person is standing up or lying down.

The dimensionality of the data can, furthermore, be reduced, for instance by calculating the magnitude of a multidimensional vector. The *absolute velocity* of a three-dimensional velocity stream, for instance, is given by the magnitude of the X, Y and Z components of the velocity vector. This value is useful in describing the speed of an object, without paying attention to direction of the velocity vector.

3.5.3 Motion Features

Using basic differentiation and transformation techniques on a raw motion signal is a simple way of calculating salient motion features. This is particularly useful in realtime applications, where low latency is important. Without the need to consider the motion data as representations of human body motion, we can calculate features such as *quantity of motion* by summing the absolute velocities of all the markers, or *contraction index* by calculating the volume spanned by the markers.

A different type of feature can be found by taking into account the labels of the data in the motion capture signal. If two markers represent the two hands of a person, the feature *hand distance* can easily be calculated. Similarly, three markers representing the wrist, elbow and shoulder can be used to calculate the *arm extension*. More sophisticated motion features can be found by taking into account models of the mass of various limbs. One such is the 'Dempster model' [Robertson et al., 2004] which allows calculation of the kinetic or potential energy of the body or a single limb, or estimation of the power in a joint at a certain time.

The features may be purely *spatial*, meaning that they describe positional data without considering how the motion unfolds over time. Examples of this are contraction index and potential energy. Other features are *spatiotemporal*, meaning that they describe how the motion unfolds in space over time. Difference calculations such as the derivative of hand distance are typical examples of this. Finally, a feature such as periodicity is a *temporal* feature, where the spatial aspect is not described.

Meinard Müller [2007] has proposed a robust set of 7 generic kinematic features for human full body motion. The features are based on relations between joints, which make them work independently of the global position and orientation of the body. Furthermore, the features are boolean[8] which greatly reduces the amount of processing needed to use the features e.g. for search and retrieval in motion capture databases. I present his set of generic features below, with illustrations in Figure 3.15.

F_{plane} defines a plane by the position of three joints and determines whether a fourth joint is in front of or behind this plane. This may be used, for instance, to identify the position of the right ankle in relation to a plane spanned by the centre of the hip, the left hip joint and the left ankle. If a value 1 is assigned when the foot is in front of the plane, and 0 when it is behind, a normal walking sequence would show an alternating 0/1 pattern.

F_{nplane} specifies a vector by the position of two joints and a position along the vector where a plane normal to the vector is defined. For instance, a plane that is perpendicular to the vector between the hip and the neck, located at the head, can be used to determine whether the right hand is raised above the head.

F_{angle} specifies two vectors given by four joints and tests whether the angle between them is within a given range. For instance, the vector between the right ankle and right knee and the vector between the right knee and the right hip could be used to determine the extension of the right knee joint.

F_{fast} specifies a single joint and assumes a value of 1 if the velocity of the joint is above a chosen threshold.

F_{move} defines a vector between two joints and assumes a value of 1 if the velocity component of a third joint is positive in the direction of the defined vector.

F_{nmove} defines a plane between three joints and assumes a value of 1 if the velocity component of a fourth joint is positive in the direction of the vector normal to the plane.

F_{touch} measures the distance between two joints or body segments and assumes a value of 1 if the distance is below a certain threshold.

From the 7 generic features Müller has defined 39 features which contain specific information about the joints and thresholds used. In Müller's research these are used to recognise various full-body actions such as performing a 'cartwheel' or a 'squat'. The 39 boolean features make up a *feature matrix* which describes a single recording. A computer system is used to define so-called *motion templates*, which are real-valued prototypes of the feature matrices that correspond to a certain action. The motion templates are learned by the system by inputting a number of labelled data examples. Motion templates can be used to identify new versions of the same action by using *dynamic time warping* and a distance function which matches the input data to the learned motion templates. Müller also provides a way of visualising the motion templates, which is shown in the next chapter.

[8] Boolean means that the possible values are either 0 or 1.

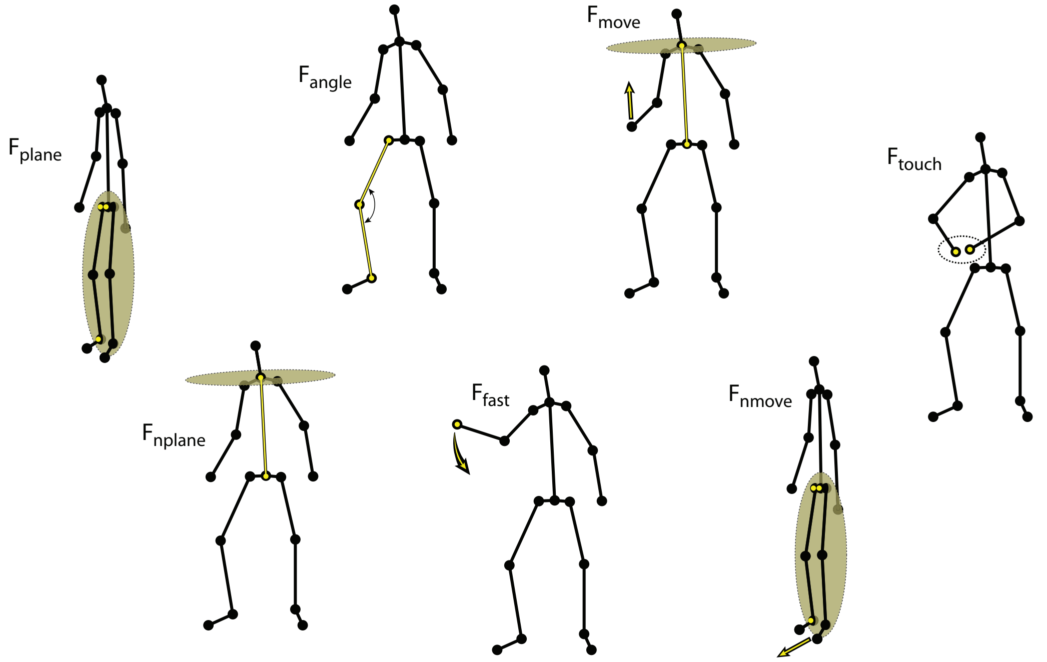


Figure 3.15: Illustrations of Müller's generic kinematic features. The yellow marks denote the joints or vectors that are used in the illustrated implementation of the feature. Refer to the main text for explanation.

3.5.4 Toolboxes

The *MoCap Toolbox* for Matlab, developed at the University of Jyväskylä, includes a variety of feature extraction algorithms for motion capture data [Toiviainen and Burger, 2011]. This includes functions for calculating derivatives, filtering, cumulative distance and periodicity, and models that take the weight of body segments into account, enabling the calculation of potential and kinetic energy. Furthermore, the toolbox has implemented algorithms for calculating the *eigenmovements* of a full body motion capture segment by using *principal component analysis* (PCA) [Duda et al., 2000]. PCA is a method of data reduction applied by projecting the original data onto a set of *principal components*. The first principal component is defined as the vector on which the data in a data set can be projected to explain as much of the variance in the data set as possible. The second principal component is perpendicular to the first and explains as much of the remaining variance as possible. Toiviainen et al. [2010] showed the utility of PCA in motion analysis for a set of motion capture recordings with 3D positions of 20 joints, equivalent to 60 data series. By keeping only the 5 highest ranked principal components, 96.7 % of the variance in the data was explained. The analysis allowed the researchers to distinguish between periodicities in various parts of the body of the subject, and to observe relations between the motion in the different segments.

Other tools have been developed for extracting features from video data and can in principle be used with sensors as simple as an ordinary web camera. Antonio Camurri's EyesWeb software is designed for extracting features from motion data in real time [Camurri et al., 2004]. The software can extract a body silhouette from a video signal, and a number of features can be calculated, most notably the *quantity of motion* and *contraction index*. These have been shown to be pertinent to the experience of emotion in dance [Camurri et al., 2003].

Quantity of Motion is calculated as the number of moving pixels in the silhouette and reflects the overall motion in the image.

Contraction Index denotes the extension of the body and can be estimated by defining a rectangular bounding region around the silhouette (area of motion) and comparing the total number of pixels within this area with the number of pixels covered by the body silhouette.

The *Musical Gestures Toolbox*, developed by Alexander Refsum Jensenius, includes some of the features that are implemented in the EyesWeb software [Jensenius, 2007a]. This software is implemented in Max as modules in the Jamoma framework [Place and Lossius, 2006], and unlike EyesWeb it is open source. The toolbox includes modules for preprocessing video, calculating features such as the quantity of motion, area of motion, the barycentre of the motion in the image, and also smoothing and scaling of the data. The toolbox also contains numerous modules for visualising motion, which will be covered in Section 4.1.

3.6 Storing and Streaming Music-Related Data

We can distinguish between two main use scenarios for tracking data. Firstly, as explained in Section 3.1, *motion capture* involves storing the tracking data in order later to apply analysis or import the data in animation software. Secondly, *realtime tracking* involves using the tracking data directly, within a very short time period after the motion occurs. Realtime tracking is used, for instance, in interactive systems such as motion-based computer games like Microsoft Kinect. When a user performs an action it is reflected in the movement of an avatar some milliseconds later, after the necessary processing has been completed.

In music-related contexts tracking data are often just one part of the total amount of data involved. In addition to motion data, music-related data include video, audio and symbolic representations of musical sound such as MIDI-data or sensor data from electronic musical instruments. Furthermore, music researchers and performers use features that are extracted from the tracking data. These may be simple, time-varying transformations, such as relations between body limbs, or distinct events such as sound-producing actions or musical phrases and also higher-level features such as descriptions of the emotive content of the music. The diversity of these data is challenging: sampling rates range typically from 44.1 kHz for audio and down to less than one event per second for event-based data such as MIDI, and dimensionality varies from a single number per sample for audio data to more than one million pixel values for one frame of video data. An overview with some typical examples of music-related data, adopted from [Jensenius et al., 2008], is presented in Table 3.1. Thus for storing data we need a format that can to handle the different data types, and for streaming we need a protocol that enables simple routing of the different types of data in realtime applications.

3.6.1 The Gesture Description Interchange Format

Most commercial mocap systems provide proprietary file formats for storing tracking data, with the option to export the data to a more open format. These solutions are sufficient in most

Table 3.1: The data types used in the experiment presented by Jensenius et al. [2008]. The different numbers of sensors, sampling rates, bit resolutions and channels per device are challenging to handle with standard protocols for storing and streaming tracking data.

Input	Sampling rate	Sensors	Channels	Bit resolution
Accelerometer	60 Hz	9	3 DOF	32
Polhemus tracking	60 Hz	2	6 DOF	32
Bioflex EMG	100 Hz	2	1 DOF	7
High-speed video	86 Hz	1	320 × 240	8
Audio	44100 Hz	1	2 (Stereo)	16
MIDI	Event-based	1	3	7

motion capture settings. However, in research on music and motion the standard formats often fall short since they are not able to handle the wide variety of data at hand [Jensenius, 2007a].

Jensenius et al. [2006b] proposed the *Gesture Description Interchange Format* (GDIF) as a multi-layered approach to structuring music-related data. The various layers in GDIF contain different representations of the data, with the most basic *acquisition layers* containing raw sensor data, and sensor data where some simple processing (e.g. filtering) has been applied. Next, the *descriptive layers* describe the motion in relation to the body, in relation to a musical instrument or in relation to the environment. Then the *functional* and *meta layers* contain descriptions of the functions of the various actions in a recording (sound-producing, communicative, etc.), and abstract representations, higher-level features and metaphors.

GDIF was mainly proposed as a concept and idea for structuring music-related data, and not as a file format *per se*. In a panel session at the International Computer Music Conference in 2007 the *Sound Description Interchange Format* (SDIF) was suggested as a possible format for the implementation of GDIF [Jensenius et al., 2007]. As shown in Figure 3.16, SDIF tackles the challenge of synchronising data with different sampling rates by organising the data into time-tagged frames in individual streams [Wright et al., 1998]. SDIF also allows data with different dimensionality in the individual streams. The use of SDIF as a storage format for music-related data has been explored by several researchers [e.g., Jensenius et al., 2008, Peters et al., 2009, Bresson and Schumacher, 2011] and is currently the most used format in GDIF development.

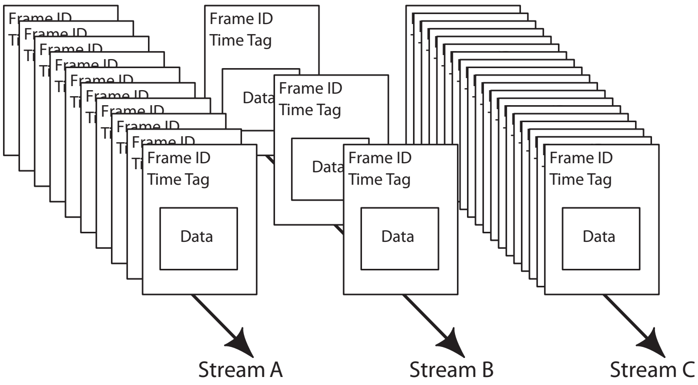


Figure 3.16: The Sound Description Interchange Format arranges data into individual streams containing time-tagged frames.

More recently researchers in the SIEMPRE EU FP7 ICT project have developed a system that allows synchronised recordings of data from several devices using SMPTE time-coding [Gillian et al., 2011]. An XML-based file format and synchronisation protocol has been developed for storing synchronised recordings of audio, video and text-based sensor and mocap data.

The system also includes a solution for uploading recordings to a server, and visualisation tools for video, motion capture, sensor data and audio using EyesWeb. A similar database solution for classifying and performing search and retrieval of music-related actions has been proposed by Godøy et al. [2012], and is currently under development at the University of Oslo.

3.6.2 Open Sound Control

One application of realtime tracking in music is in interactive systems such as digital musical instruments or other interactive sound installations. This may entail streaming the data from a tracking system and *mapping* features extracted from the data to a synthesiser. Adopting terms from Miranda and Wanderley [2006], the motion data and extracted features are referred to as *gestural variables* and the parameters available for controlling the sound output of the synthesiser are called *synthesis parameters*. With the large amount of data that is communicated, and also with different representations of the data, it is important to have a structure for communicating between gestural variables and synthesis parameters.

The Open Sound Control (OSC) protocol, introduced by Wright and Freed [1997], has become the leading protocol for communicating music-related data in research on novel musical instruments. A main idea in OSC is to structure music-related data hierarchically, for instance to facilitate mapping between gesture variables and synthesis parameters in digital musical instruments. The hierarchical structure is reflected in the so-called *OSC-address* which is sent together with the data. Each level is separated in the OSC-address by a slash "/". One example could be the following OSC-namespace for synthesis parameters in a musical instrument:

- `/synthesiser/1/oscillator/1/frequency`
- `/synthesiser/1/oscillator/1/amplitude`
- `/synthesiser/1/oscillator/2/frequency`
- `/synthesiser/1/oscillator/2/amplitude`

Here, '`/synthesiser`' is at the top level, and the '`/1`' indicates that we are referring to the first of possibly several synthesisers. The '`/frequency`' and '`/amplitude`' of two oscillators can be controlled. Thus to set the frequency of the first oscillator to 220 Hz, we would use the control message '`/synthesiser/1/oscillator/1/frequency 220`'.

Synthesis parameters are only one aspect of OSC messages. OSC is also a good way of structuring gesture variables. The Qualisys motion tracking system[9] has native support for OSC, and researchers have developed applications for interfacing with several other tracking systems via OSC, e.g. Vicon,[10] Nintendo Wii,[11] and Xsens MVN [Skogstad et al., 2011]. For full body motion capture data examples of OSC addresses might include:

- `/hand/left/velocity`
- `/head/position`

[9] http://www.qualisys.com

[10] http://sonenvir.at/downloads/qvicon2osc/

[11] http://www.osculator.net/

Various tools have been developed for using OSC-formatted data in the development of musical instruments, for instance the Open Sound Control objects for Max provided by CNMAT.[12] The *Digital Orchestra Toolbox*, developed by Joseph Malloch et al. [2007], also includes a mapping tool that simplifies mapping between OSC-formatted gesture variables and synthesis parameters. Malloch's mapping tool was later included in Jamoma which also includes several other tools for mapping between control data and sound [Place et al., 2008].

3.7 Summary

This chapter has introduced a variety of motion tracking technologies with a main focus on optical infrared marker-based motion tracking. Some general concepts in motion tracking have been introduced. Tracked objects include markers, rigid objects or kinematic models, and the type of object defines the type of tracking data provided. Positions and orientations can be described in relation to a global or local coordinate system defined by the tracked object itself or by another object.

The chapter also introduced basic processing techniques for motion data, including gap-filling and smoothing. Some feature extraction techniques were introduced, with basic differentiation and transformation, and Müller's motion features as examples of how boolean features can be extracted from relation of body limbs. Further, some features available in toolboxes for working with music-related motion were introduced. Finally, I presented some of the challenges of storing and synchronising music-related data, and basic theory on how motion tracking can be used in real time for musical applications.

[12] http://cnmat.berkeley.edu/downloads

Chapter 4

Methods of Analysis

Chapter 3 having explained how motion can be captured by various tracking technologies, this chapter will introduce the methods that have been applied in the thesis to analyse correspondences between sound and body motion. Several of the methods presented here are well-known, and more comprehensive details of these methods can be found in most textbooks on statistics. In my own analysis I have used existing software to run statistical tests and for classification, and therefore only a basic introduction to the methods is offered here, as a background to the analysis results and assessments that are made in the papers included in this thesis.

Stanley S. Stevens [1966] introduced the term *cross-modality matching*, denoting the process of matching some sensory input in two modalities. Steven's use of the technique involved an experiment in which participants were asked to adjust the sound level of a tone to match the strength of a vibration applied to their finger, and the other way around — adjusting the strength of the vibration according to the apparent loudness of the tone. The analyses presented in several of the papers included in this thesis are based on a variant of the cross-modality matching approach, in studies referred to as *sound-tracing*. Experiment participants were asked to match their body motion to some auditory input (i.e. to 'trace the sound'). Analysis of the data involves comparing features of the sound objects used as stimuli with features of the recorded motion.

Most of the sound stimuli used in the experiments have durations of less than 5 seconds and each constitutes a single sound object. The relations between sound and motion are analysed on a chunk timescale level and a sub-chunk timescale level (ref. the discussion in Section 2.3.2), but not as multiple concatenated chunks. Analysis at the sub-chunk timescale level is concerned with comparing features that contain numerical values in each timeframe. Borrowing terminology from Peeters et al. [2011], I refer to them as *time-varying features*. Other features describe an entire object; for instance, the mean acceleration of an action or the categorical labelling of a sound object as 'pitched'. These features consist of a single value or a single description for an entire object and are referred to as *global features*. Figure 4.1 displays various examples of the two main feature types involved.

4.1 Visualisation of Motion Data

A requirement of analysis of music-related data is to have good visualisation techniques. In addition to providing qualitative assessments from tracking data, good visualisation techniques

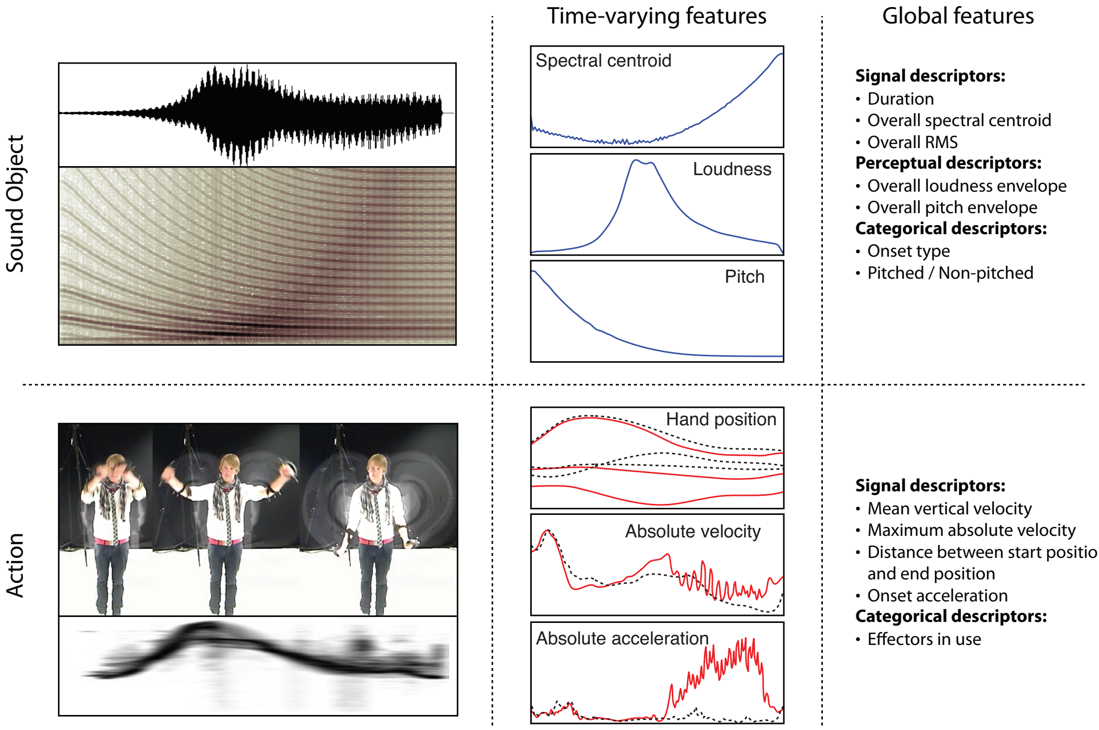


Figure 4.1: A sound object with a corresponding action (sound-tracing) and feature examples. *Time-varying features* contain separate values for each frame, and *global features* are either overall numerical calculations based on the time-varying features or non-numerical classifications of the objects.

facilitate conveying analysis results to other researchers, project funders and the general public. Further, visualisations are an essential aid in developing hypotheses that can be tested quantitatively [Moore and McCabe, 2006]. Displaying motion data over time is not trivial, particularly because of the large number of dimensions that a motion capture recording typically contains. In some cases a simple plot of absolute velocity over time is sufficient, but if 3D marker positions, velocities and accelerations are to be displayed for multiple markers, a timeline plot soon becomes unreadable. This section will cover the background of the visualisations I have used in my own work, including two techniques that I have developed.

4.1.1 The Challenge of Motion Data Visualisation

Motion data span both *time* and *space*, and it is important to have visualisation techniques that cover both of these domains. Time is one-dimensional, and spatial position three-dimensional, and in the end we want techniques that display all these dimensions on a two-dimensional medium, namely paper.

A straight forward and quite common way of plotting motion data is with time on the horizontal axis and position the vertical axis. In Figure 4.2 this is shown for a single marker on the right wrist of a pianist. The plot provides precise temporal information and when zooming in it is also easy to read the precise position of the wrist at a certain time.

Although Figure 4.2 gives precise information about the motion of the hand marker, dividing the original single trajectory into three lines seem to run counter to intuition. Furthermore, motion data usually consists of more than a single marker, and attempting to plot all the markers in a mocap recording on a timeline is in most cases too cumbersome. Figure 4.3 shows this, by

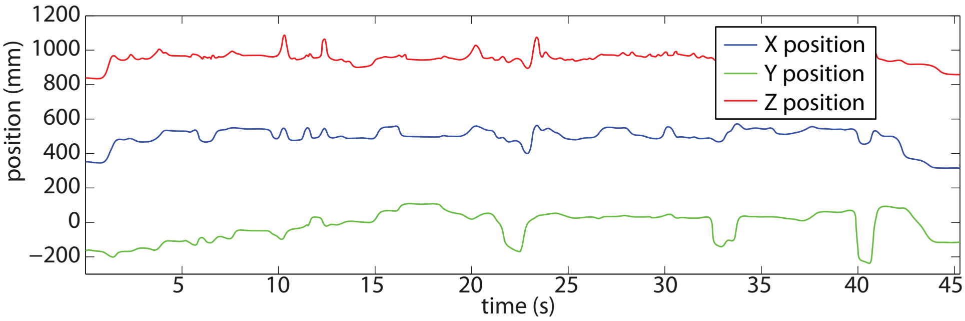


Figure 4.2: A common way of plotting three-dimensional marker data in time and space.

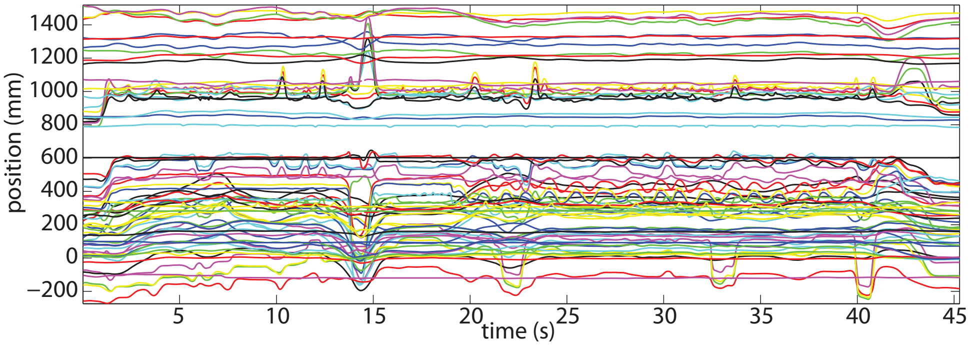


Figure 4.3: Plots of X, Y and Z positions of 24 markers from motion capture of a short piano performance. Although the plot provides some impression of salient moments (e.g. between 14 and 15 seconds), it is too complex to provide any detailed information.

plotting the X, Y and Z positions of all the 24 markers of the same piano performance.

Several visualisation techniques are able to present marker data in a more intuitive manner than Figure 4.2, and there are also techniques for displaying full-body motion without the need of plots as in Figure 4.3. There is often a trade-off between intuition and precise representations of time and space in these techniques. It takes time to become familiar with some of the methods, while others can be understood without any explanation.

Returning to the terms introduced in Section 2.3.2, we can relate the visualisation techniques to three timescale levels: *sub-chunk*, *chunk* and *supra-chunk*. Visualisations at the sub-chunk level display motion in an instant, or over a very short period of time. Such visualisations typically show a static pose and therefore the spatial aspect is important. At the supra-chunk level visualisations of long time periods may often be at the expense of spatial information. In some visualisations at the chunk level the time-span is reduced enough to be able to combine good representations of both time and space.

The relation between visualisations and the three timescale levels is particularly evident in the visualisation techniques implemented in the Musical Gestures Toolbox [Jensenius, 2007a] which was introduced in Section 3.5.3. I shall illustrate these techniques before continuing with visualisation techniques for three-dimensional motion data.

4.1.2 Motion in Video Files

Jensenius' tools for analysing motion in video contain several techniques for visualising motion [Jensenius, 2012a]. The toolbox is based on differentiating and filtering video frames, and algorithms for visualising the video as it unfolds over time. Three of the methods are listed below and are illustrated in Figure 4.4.

Motion images display the changes in the current from the previous video frame. Various filtering and thresholding techniques can be applied to remove unwanted noise from the motion image.

Motion history images display a combination of several motion images extracted from a sequence of video frames, for instance by averaging the pixel value across all of the motion images. Jensenius implemented various ways of calculating motion history images, which all show different qualities of the analysed video.

Motiongrams are displayed by collapsing each motion image frame down to one-dimensional images, either horizontal or vertical. The collapsing is done by averaging the pixel values across one of the dimensions. The one-dimensional image that is produced is plotted on a timeline and provides a visual impression of the evolution of motion in the video.

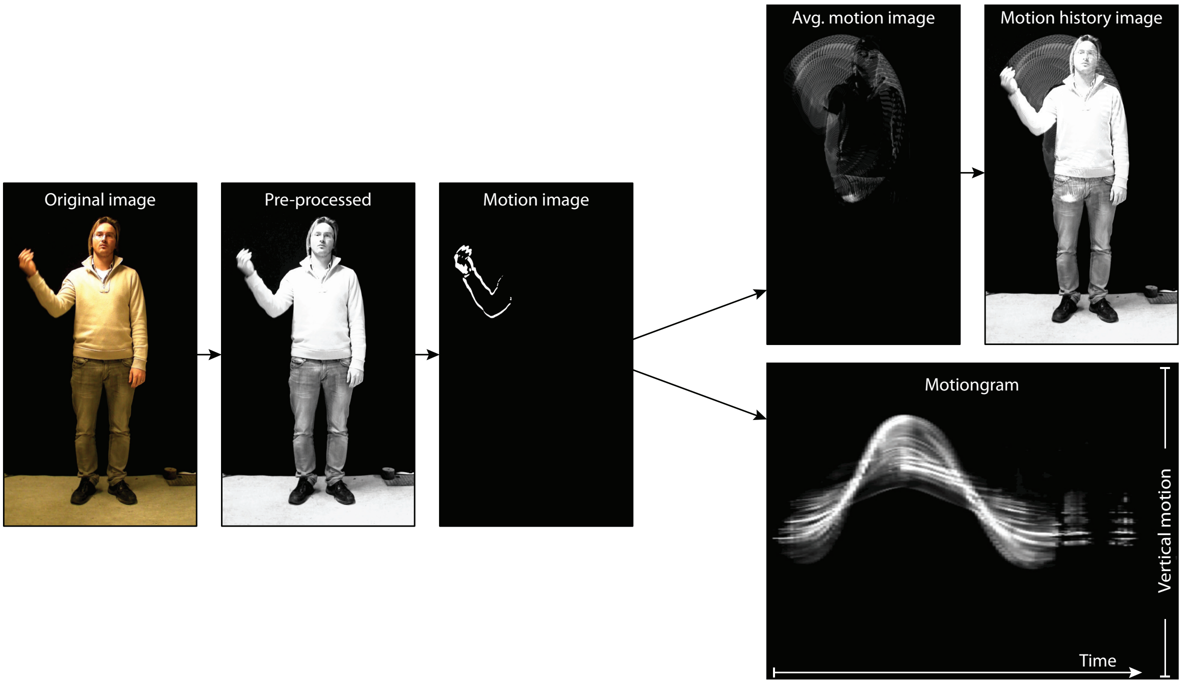


Figure 4.4: Jensenius' techniques for visualising motion. (Adapted from [Jensenius, 2012b])

In the images shown in Figure 4.4 the motion image shows which part of the body is moving in this instant. The image shows precisely which pixels are different between two successive frames of video data, and is a sub-chunk visualisation of motion. The motion history image shows a slightly longer timespan, providing a quite intuitive description of the spatial trajectory of the hand. However, the image does not show precisely which pixels have changed in each timeframe. Finally, a motiongram can be made for longer segments of movement, and motion can be displayed with as high temporal precision as the framerate of video file. However, the spatial information has been reduced, since the motiongram can only display one dimension at a time. Furthermore, the motiongram is less intuitive than the motion history image, because most people are not used to looking at one-dimensional images unfolding over time.

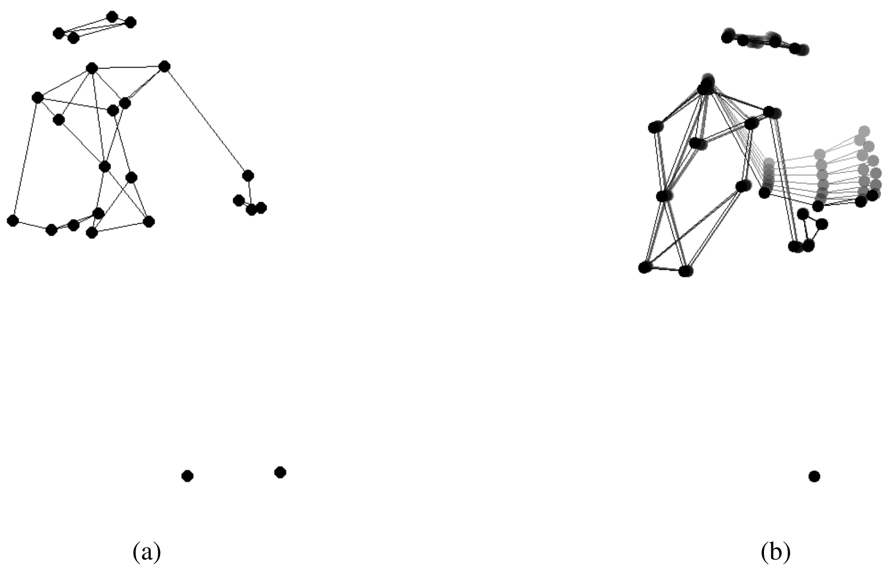


Figure 4.5: The figure shows the 24 markers in the piano recording plotted in Figure 4.2 and Figure 4.3, displaying the head, torso and arms with interconnected lines as well as the feet. (a) illustrates a pose without time-information and can be seen as a cross-section of Figure 4.3 at time = 4 seconds. (b) shows how multiple sequential poses can be superimposed to display trajectories over a time-period.

4.1.3 3D Motion Data

In the case of 3D motion capture data the various suppliers of motion tracking equipment provide proprietary environments for visualising their data.[1] This may involve a 3D view of markers with interconnected lines. It is also normal to be able to show marker trajectories for the past and coming frames in the 3D view. Furthermore, the programs typically contain timeline views of the individual markers with position, velocity and acceleration. These visualisations are useful in getting an initial overview of the motion data; however, the solutions are inadequate if we want to apply various processing techniques to the data that are not implemented in the proprietary motion capture software.

Toiviainen's MoCap Toolbox provides a variety of scripts for plotting motion data [Toiviainen and Burger, 2011]. Individual marker positions and processed data can be plotted on timelines, and marker positions in any timeframe can be plotted in *point-light displays*, as shown in Figure 4.5(a). Such point-light displays have been shown to retain salient perceptual information about the motion, allowing people to recognise the gender of a person, or the affect of bodily gestures [Kozlowski and Cutting, 1977, Pollick et al., 2001]. The toolbox also includes a feature for collecting a sequence of such poses in a video file. By using image processing software the point-light displays can be put together into an intuitive visualisation of motion trajectories at the chunk-level. Figure 4.5(b) shows an example of this where multiple sequential poses have been superimposed.

The supra-chunk level can be illustrated through basic Matlab functions by plotting the position in each timeframe in a scatterplot. However, the plot quickly becomes too complex when more than a single marker is included. Figure 4.6 shows how the position of the same

[1] e.g. Naturalpoint Arena for OptiTrack, Xsens MVN Studio, and Qualisys Track Manager

marker as in Figure 4.2 can be plotted in a more intuitive manner than with the time-series plot. Again, there is a trade-off between precise data and intuition — position and temporal information are present in the plot but cannot be read as precisely as in the time-series plot of Figure 4.2. Supra-chunk trajectory plots are useful for observing how a single marker moves over a longer time period and I have made these for one of our lab publications so far [Jensenius et al., 2012].

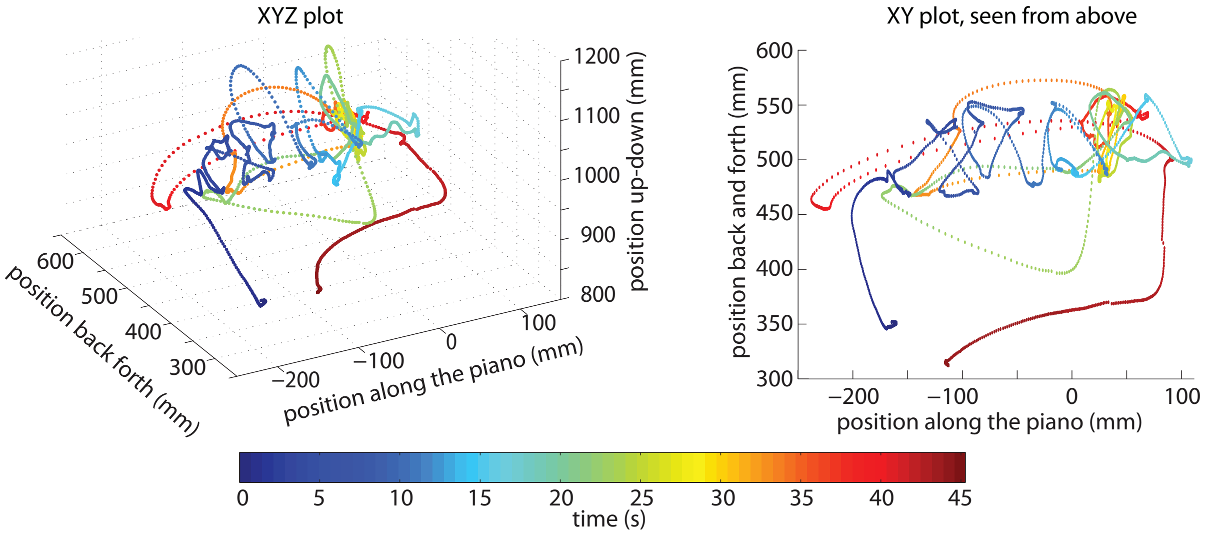


Figure 4.6: The trajectory of a single marker can be shown in 2D or 3D plots. Time can be shown by colour-coding the trajectory. The marker shown in the plots is the same right wrist marker as in Figure 4.2.

4.1.4 High-Dimensional Feature Vectors and Multiple Data Series

When it is desirable to visualise an entire full-body mocap recording or a set of time-varying features describing the data, colour information can be used to indicate the position of each marker, or the magnitude of the features.

Meinard Müller [2007] used colour information to visualise 39 features in his *motion templates*. In this technique each feature is assigned a separate row in a matrix and the time-frames are shown in the columns. This allows studying a high number of dimensions on a timeline, and provides an overview of patterns in the mocap data. An example is shown in Figure 4.7.

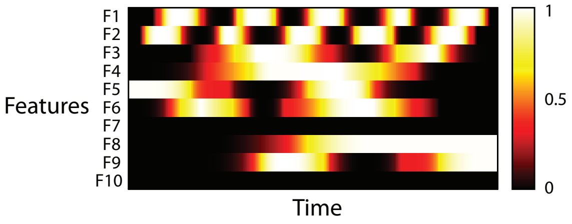


Figure 4.7: Example of Müller's visualisation technique for motion templates, showing the ten first features (not based on actual data). The top two rows show values that alternate between 0 and 1, something that could represent some feature of the left and right foot respectively in a walking pattern.

A similar technique can also be used to show positions of a larger number of markers by assigning the markers to individual rows and projecting the spatial coordinates onto a colourspace [Jensenius et al., 2009]. Figure 4.8 shows a so-called *mocapgram* of the 24 markers in the same piano performance as used in the plots above. Marker names following Vicon's plugin gait[2] convention are shown on the left. The XYZ coordinates have been projected onto red, green and blue, respectively and the values in each row are normalised. Although we can not tell the precise position of the markers from the plot, certain clear patterns can be seen — for instance the large trajectories in the right arm (RELB,RWEI,RHAO,RHAI) at 22, 33 and 40 seconds. Note also the almost binary pattern in the right toe (RTOE) when the sustain pedal is pressed.

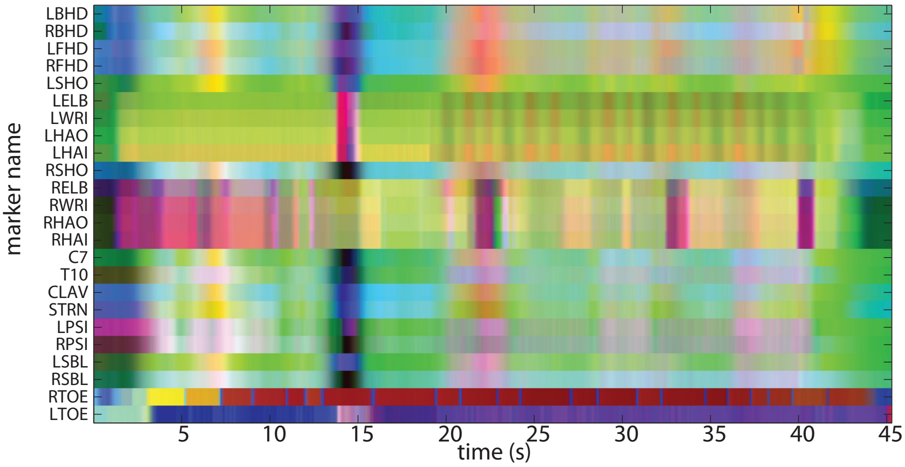


Figure 4.8: Mocapgram showing 3D position coordinates mapped onto a colourspace.

In my own research I needed to display the results of a large number of motion capture sequences in order to show general tendencies in the data. I developed mocapgrams further, in a script in Matlab for visualising data [Kozak et al., 2012, Nymoen et al., 2012]. Figure 4.9 is adopted from Paper VII, and shows how multiple motion capture recordings can be compared (in this case only 5 recordings). The use of these plots in the paper involved comparing motion capture data with a sound stimulus. The stimulus started 0.5 seconds after the start of the motion capture recording and ended 0.5 seconds before the recording ended. As shown in the figure the value of each data series is given as a shade of grey, here normalised between 0 and 1. The mean value of the 5 data series at each time-frame is shown as a dashed line, with two dotted lines showing the standard deviation. The units for the mean value plot are on the left axis.

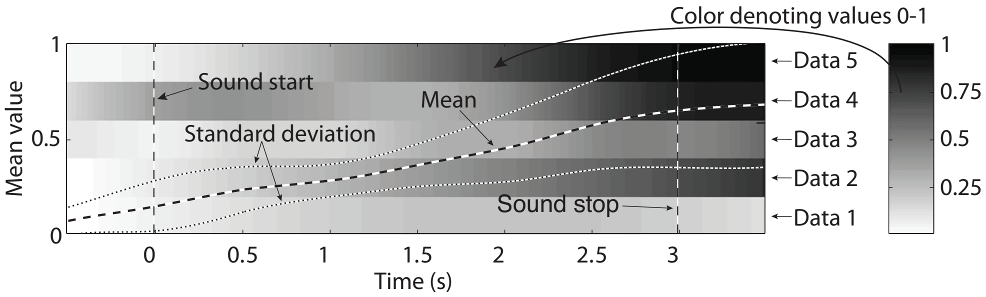


Figure 4.9: Mocapgram example, adopted from [Nymoen et al., 2012].

[2] http://fourms.wiki.ifi.uio.no/MoCap_marker_names

The mocapgrams do not give precise information on the value in each time-series since the different shades of grey may be difficult to distinguish. However, the temporal information is as precise as in any time-series plot, and the plots facilitate illustration of the distribution of a large number of time-series. Figure 4.10 shows an example of how this technique can display a larger number of mocap recordings. The figure shows the absolute acceleration of a rigid object in 122 recordings, all of which are sound-tracings of sound objects with impulsive onsets.

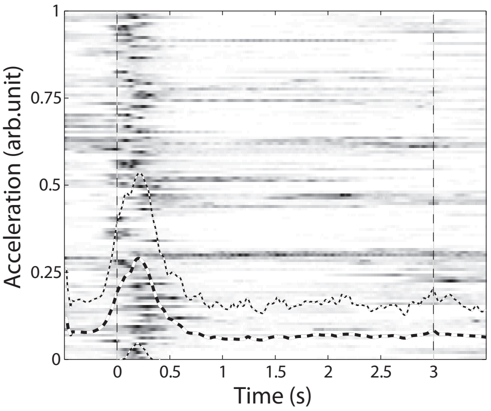


Figure 4.10: Mocapgram showing 122 data series of acceleration data. The data stem from sound-tracings of sounds with impulsive onsets.

4.1.5 Realtime Visualisation

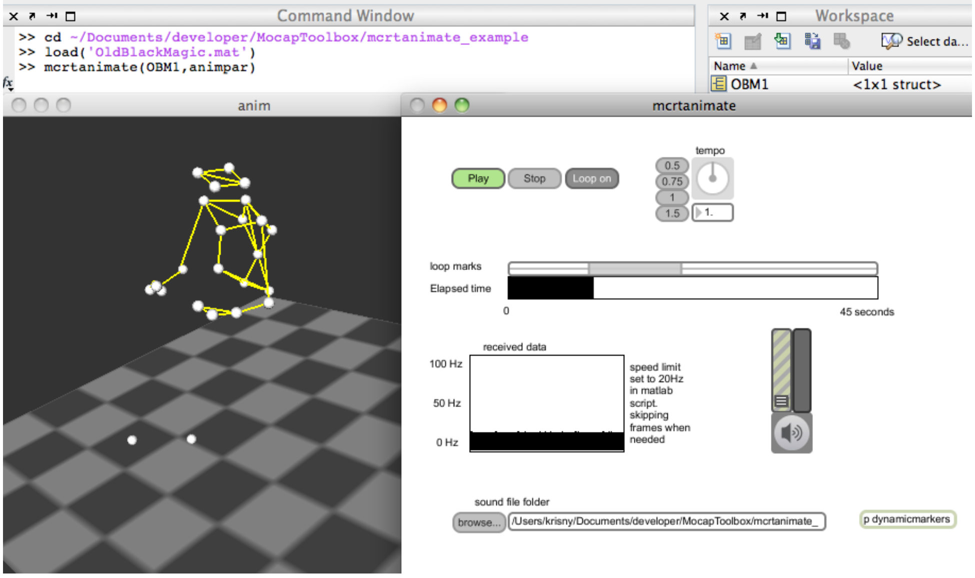
The MoCap Toolbox is an excellent tool for working with recorded motion capture data. However, I missed an interactive 3D visualisation functionality, which could allow playing back motion capture data at different tempi, synchronised with sound files, with support for scrubbing back and forth in the recording and looping short segments of the motion capture data. I therefore implemented an addon to Toivianen's MoCap toolbox, which allows 3D display of motion capture data with scrubbing, looping, zooming, rotating and tempo adjustments, synchronised with audio. The implementation with an example recording is available for download at the fourMs website, along with a video that demonstrates the functionality.[3] Figure 4.11 shows a screenshot of this tool in action, and more details on the tool are provided in Section 5.3.

While visualisations of sound and motion features are useful, they are rarely a sufficient means of analysis. The sections below cover various quantitative methods that can be applied in experiments on correspondences between sound and motion features.

4.2 Statistical Tests

We can use visualisation techniques or simple statistical measures such as mean and standard deviation to get an indication of differences between various groups of data. However, the indications obtained from inspecting visualisations alone should preferably be tested quantitatively. Take as an example a comparison of the body mass of male Danish and Swedish citizens. Just by walking around the streets of Denmark and Sweden we could get a visual impression of the difference (or similarity) between the two populations, but to check the accuracy of our impression we would need to measure the body mass of the people. Since we cannot possibly measure

[3] http://fourms.uio.no/downloads/software/mcrtanimate



```
>> cd ~/Documents/developer/MocapToolbox/mcrtanimate_example
>> load('OldBlackMagic.mat')
>> mcrtanimate(OBM1,animpar)
```

Name ▲	Value
OBM1	<1x1 struct>

speed limit set to 20Hz in matlab script. skipping frames when needed

sound file folder: /Users/krisny/Documents/developer/MocapToolbox/mcrtanimate_

Figure 4.11: My implementation of interactive 3D animation for Toiviainen's MoCap Toolbox.

this for every male person in these countries, we select a subset from each country, called a *sample*. If the samples consist of a hundred Danes and a hundred Swedes chosen at random, the mean mass of the Danes and Swedes will probably be different by a small amount, and there will be some variation within the groups of Swedes and Danes. If the difference between the means is large and the variation within each group is small, we can be quite certain that there is a difference between the populations. However, if the difference is small and the variation within each group is large, we cannot generalise the result to count for the entire Danish and Swedish populations.

Similar problems are commonly faced in many research areas. Various statistical tests can be used to assess the *statistical significance* of the difference between two samples. In other words these tests estimate the probability that there is a difference between two populations based on a sample drawn from the populations. In some of my papers results from *t-test*[4] and *analysis of variance* (ANOVA) are reported. The tests have been applied to compare global motion features for various groups of motion capture recordings; for instance, to assess the statistical significance of the difference between *onset acceleration* for sound-tracings related to sounds with a soft onset and sounds with an impulsive onset.

The statistical tests discussed here assume that the data samples in each set are normally distributed. The results from the tests are thus exactly correct only for normal populations, something which is never the case in real life [Moore and McCabe, 2006]. If we use a larger sample size, the standard deviations of the set will approach the true standard deviation of the population. Thus the robustness of statistical tests increases with the sizes of the samples that are tested. Moore and McCabe [2006] state that even clearly skewed (i.e. not normally distributed) populations can be tested with *t*-tests when the sample size is larger than 40.

[4] Also called Student's *t*-test, after the inventor W. Gosset who was prevented by his employer from publishing under his own name. He published this technique under the pseudonym "Student" [Moore and McCabe, 2006].

4.2.1 *t*-test

A t-test can be used to test the statistical significance of the difference between random samples from two populations. The process involves defining a *null hypothesis*, stating that the means of the populations are equal, and this null-hypothesis is verified or falsified upon the t-test. For the sake of comparing results between experiments three measures are provided when reporting the results of t-tests: (1) The *degrees of freedom* (*df*)[5] is calculated from the sample size, and describes the number of values that are free to vary. (2) The *t-statistic* is calculated from the sample sizes as well as the standard deviations and mean values of the samples. (3) The *p-value* is the probability that the null-hypothesis is true, and is derived from the t-statistic.

The p-value denotes the probability that the two samples stem from populations with equal mean values. The sizes, means and standard deviations of the samples are used to estimate this probability. The p-value is used to infer whether the difference between the two distributions is statistically significant. A *significance level* (α) is defined and if p is less than this value, the result is said to be statistically significant at level α. Typical levels for α are between 0.001 and 0.05 [Moore and McCabe, 2006].

4.2.2 Analysis of Variance

In many cases Analysis of Variance (ANOVA) rather than the t-test is applicable. Like the t-test ANOVA tests for statistically significant differences between groups, but can take multiple groups into account. In other words while a t-test can be used to assess the statistical significance of the difference in *two* sample means, an ANOVA can be applied to test whether the observed difference in mean values of *several* groups is statistically significant.

Furthermore, ANOVA allows measurement of the significance of several factors, or features, at once. For instance, in the example with Danes and Swedes presented above, the age of those measured could be added in the analysis. This would allow us to infer whether there is a difference in body mass between Danes and Sweden and, further, whether age is related to mass.

ANOVAs do not use the t-statistic but rather an *F-statistic*. This statistic is based on the variations *within* each group and *between* the groups [Moore and McCabe, 2006]. In addition to the F-statistic the degrees of freedom and the p-value are specified when reporting ANOVA results.

4.3 Correlation

In real life we daily encounter variables that are related. The value of the volume knob on a hi-fi system is related to the sound level of the output, and the number of floors in a building is related to the number of steps in the stairs. If we want to determine whether or not there is a relation between two variables in a data set and how strong the relation is, we need a measure to describe how the variables *correlate*.

Correlation is a measure of the direction and strength of the relationship between two quantitative variables [Moore and McCabe, 2006]. The value of a correlation is between -1 and 1,

[5] Not to be confused with the term *degrees of freedom* (DOF) in motion tracking.

where a correlation of 1 denotes a full dependence between the two variables, and -1 denotes a full negative dependence between the variables.

Several methods are available for determining correlation coefficients. Firstly, the *Pearson correlation coefficient* measures the linear dependence between variables. When the two variables are plotted on separate axes in a scatterplot a Pearson correlation coefficient of 1 means that all the samples in the two variables follow a straight ascending line, and similarly a correlation coefficient of -1 shows as a straight descending line, as shown on the left in Figure 4.12 [Zou et al., 2003]. Non-linear correlations may also exist, for instance if one of the input variables stems from a skewed distribution. This is particularly true in music-related research, where several sound features scale logarithmically (e.g. loudness and pitch). For non-linear relations, the *Spearman* ρ measure is more applicable than the Pearson correlation. Non-linearity is achieved by ranking (ordering) the input variables and calculating the Pearson correlation from the rank, rather than the variable value [Spearman, 1904]. The result is that a continuously rising or falling tendency in a scatter plot will have correlation coefficients of 1 and -1 respectively, as shown on the right in Figure 4.12.

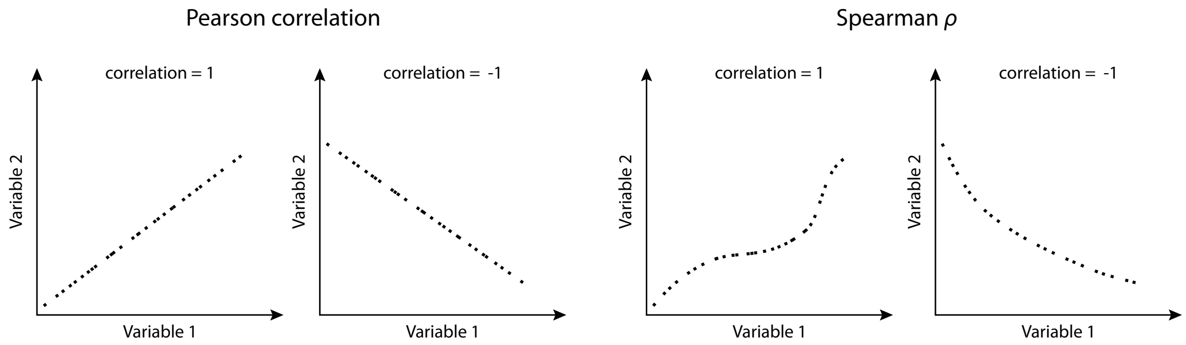


Figure 4.12: The difference between Pearson correlation and Spearman ρ. Pearson correlation measures the linear relation between the variables, and Spearman ρ uses a ranking of the variables to measure the monotonic relation between them.

4.3.1 Correlation and Music-Related Time-Series

Emery Schubert [2002] and later also several other researchers [e.g., Vines et al., 2006, Upham, 2012, Kussner, 2012] have presented critical views on the common practice in music cognition research of uncritically applying the Pearson correlation measure to time-series of music-related data without taking into account the serial nature of the data. Specifically, the correlation coefficients cannot be tested for statistical significance because the value of each sample is not drawn randomly from a normal distribution. This is because the value at each time step will be dependent on the value in the immediately preceding time steps. Take as an example a 200 Hz motion capture recording — it is impossible to have ones arms fully stretched in one time step and then fully contracted in the next time step (5 milliseconds later). Consequently the sample value in each time-frame is likely to be close to the previous sample value, and unlikely to be far away from that value. This effect is known as *serial correlation*.

Some approaches have been suggested to make correlation measures more applicable when analysing time-series in music research. For instance, the serial correlation may be lowered by downsampling the data series, or by applying the correlation analysis to the first-order difference (derivative) of the data series [Schubert, 2002]. Furthermore, Spearman ρ has been suggested as a more appropriate measure than Pearson correlation, since the ranking of sample values in Spearman ρ prevents the inflation of the correlation coefficient that occurs with Pearson correlation [Schubert, 2002].

Upham [2012] argues that the correlation coefficients themselves can be useful measures, but that one cannot uncritically report on the statistical significance of correlations between data-series, for instance by running statistical tests on the correlation coefficients. Schubert [2002] also argues that inspecting the correlation coefficients can be useful as an assessment of the distribution of correlations within a single data set. However, because of the problems with serial correlation the coefficients should not be used for comparison of data sets that have been gathered in different circumstances.

4.3.2 Cross-Correlation

The correlation between two variables is a measure of the relation between them. We may not be interested in this relation *per se*, but rather how it is affected by some other factor. For instance, we can examine how the correlation coefficient between two time-series changes if we shift one of the time-series back or forth in time. In this manner the correlation between the time-series becomes a function of a time-shift (lag) applied to one of them. This process, called *cross-correlation*, is shown in Figure 4.13.

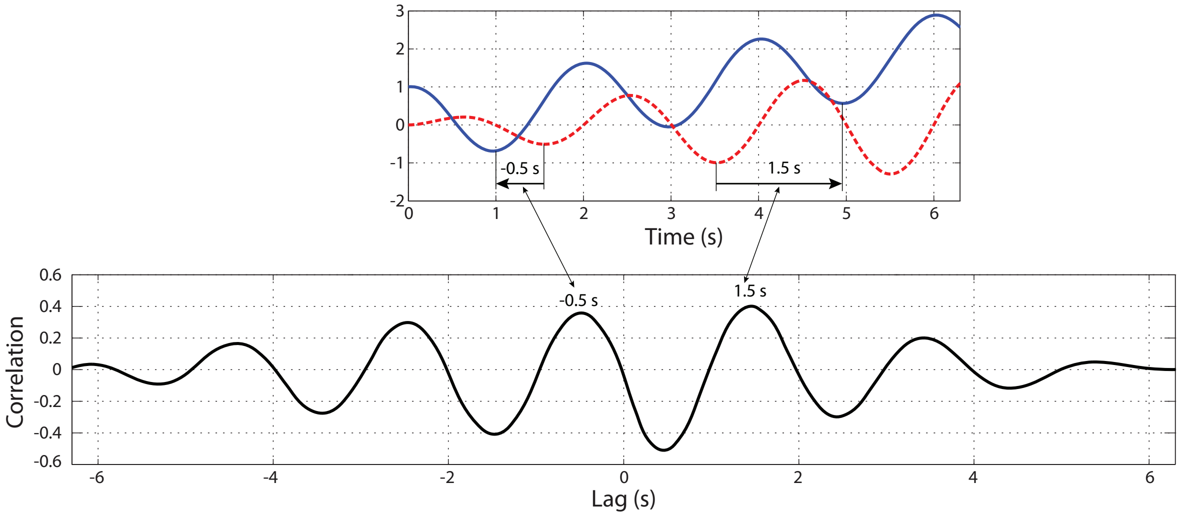


Figure 4.13: Illustration of cross-correlation. Both of the functions in the top plot have a periodic tendency at 0.5 Hz, with a phase difference of the quarter of a wavelength (0.5 s). The correlation is highest when the red dashed line is shifted back 0.5 s or forward 1.5 s.

Cross-correlation applied to two related time-series can give an indication of any time lag between them. In my research I have applied this technique to the orientation data[6] from two tracking systems running in parallel in order to analyse the latency of one system as compared with the other. Cross-correlation can also be applied to find periodicities within a single time-series. In other words we can find repeating patterns in the time-series by calculating its correla-

[6] Actually the first order difference of orientation data. This is presented in more detail in Paper III.

tion with itself as a function of a time lag, a process known as *autocorrelation*. If the time-series is periodic, the resulting cross-correlation function will have peaks at every wavelength.

4.3.3 Canonical Correlation

The correlation approaches discussed above measure the relation between two variables. Canonical correlation analysis (CCA) is slightly different in that it measures the relation between two *sets* of variables [Hotelling, 1936]. As shown by Caramiaux et al. [2010] CCA can be applied to a set of sound features and a set of motion features to analyse how several sound features and several motion features relate to each other. In this case CCA finds two sets of basis vectors, one for the sound features and the other for the motion features, such that the correlations between the projections of the features onto these basis vectors are mutually maximized [Borga, 2001].[7]

CCA is illustrated in Figure 4.14. The first projection of sound and motion features onto their respective basis vectors is that in which the correlation between the projected features is maximised. These projections are known as the *first canonical variates*.[8] The *second canonical variates* follow by projecting the features onto basis vectors that are orthogonal to the first basis vectors, i.e. the second canonical variates are uncorrelated to the first variates. This is repeated until all the dimensions in the sound features or motion features are covered (e.g. if there are 4 sound features and 3 motion features, 3 sets of canonical variates are calculated).

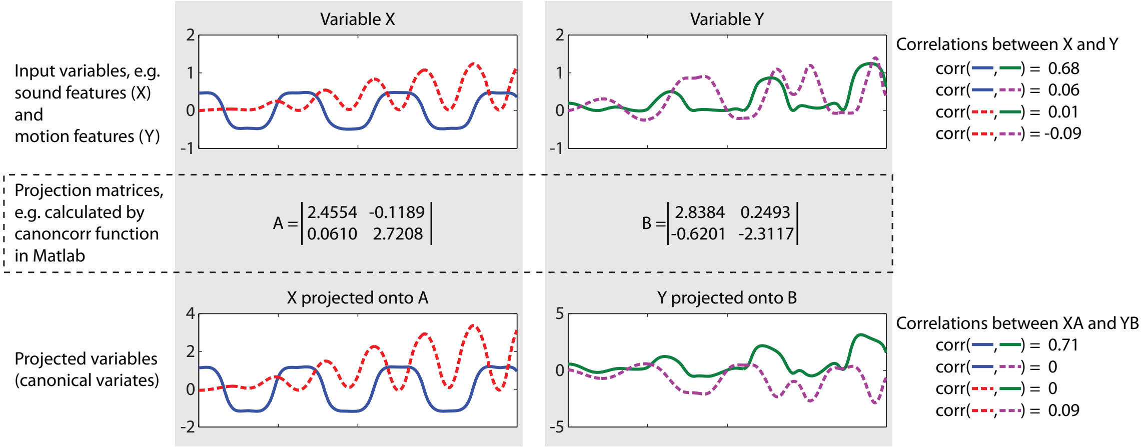


Figure 4.14: Illustration of canonical correlation. The correlations between the variables at the top are between -0.09 and 0.68. By projecting the variables onto new spaces two projected variables are found. The maximum correlation between the two sets is explained between the first canonical variates (0.71), and the correlation between the first and second variate is 0. A similar example applied to sound and motion features is shown in Paper VIII.

In my papers I have followed the approach of Caramiaux et al. [2010] and inspected the *canonical loadings* when interpreting the results of a canonical correlation analysis. This in-

[7] To readers familiar with *Principal Component Analysis* (PCA), CCA may be understood as a similar phenomenon. PCA operates on a set of variables within a single data set, explaining as much as possible of the variance in the first principal component. Then second principal component then explains as much of the remaining variance as possible, and so forth. Rather than explaining variance within a single set of variables, CCA tries to explain the maximum correlation *between two sets* of variables in the first canonical variates, and then as much as possible of the “remaining” correlation in the second canonical variates.

[8] In my papers I have referred to these as *canonical components*, but the term *canonical variates* seems to be more commonly used.

volves calculating the correlation between the input features and their corresponding canonical variate. A high canonical loading between an input variable and a canonical variate indicates that the input variable is pertinent to the correlation described by the particular canonical variate.

One weakness of canonical correlation analysis, especially if a large number of features are used, is the possibility of “overfitting” the CCA to the data. This means that the CCA might give very good solutions that are not due to actual correlations, but rather to small levels of noise in the data that are exploited by the CCA [Melzer et al., 2003]. For this reason limited numbers of sound and motion features have been used in my analyses.

4.4 Pattern Recognition-Based Classification

An analysis approach distinctly different from the correlation methods presented above considers an entire data set and implements a classifier algorithm to search for patterns within the set. Fortunately, a wide variety of ready-made implementations of computer classifiers is available, so these methods can be applied without detailed knowledge of the algorithms involved. In my work I have analysed the motion recordings with a *Support Vector Machine* (SVM) classifier. This technique was chosen because it typically matches or outperforms other classification techniques in terms of error rate [Burges, 1998]. I have used the software *Rapidminer* to implement the classifiers in my research [Mierswa et al., 2006]. This software includes a wide range of classifiers and a user interface which greatly facilitates the classification task. SVM is implemented in Rapidminer by the *LIBSVM* library [Chang and Lin, 2011], which also contains useful scripts for optimising certain parameters of the classifier. Basic concepts of computer classifiers and support vector machines are outlined below, as well as details of how classification results can be analysed.

In computer-based classification each instance in a data set is usually represented by a *class ID* and a *feature vector*. The class ID is equal among all instances in a class, and the feature vector is specific to each instance. If we want to classify fruit, and look at the class ‘apple’, all apples will have ‘apple’ as their class ID, but features such as ‘colour’, ‘size’ and ‘shape’ will vary. The data set is typically split into two subsets: a *training set* and a *validation set*. The classifier uses the data in the training set to develop rules for what is common between the instances in a class, and what distinguishes these instances from other classes. Continuing the fruit example above, a classifier may primarily use ‘shape’ to distinguish bananas from apples, but other features like ‘size’ or ‘color’ may be necessary to differentiate apples from peaches or oranges.

4.4.1 Support Vector Machines

A Support Vector Machine (SVM) classifier is trained to find a hyperplane in the feature space between the classes of training data [Duda et al., 2000]. Figure 4.15 shows the location of the optimal hyperplane between two classes, where three instances make up the so-called *support vectors*, which are equally close to the hyperplane.

It is often the case that the training data are not linearly separable. When this is so, the support vector machine increases the dimensionality of the feature space by a *kernel function*. This process is illustrated in Figure 4.16.

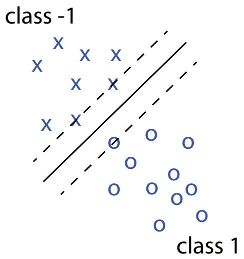
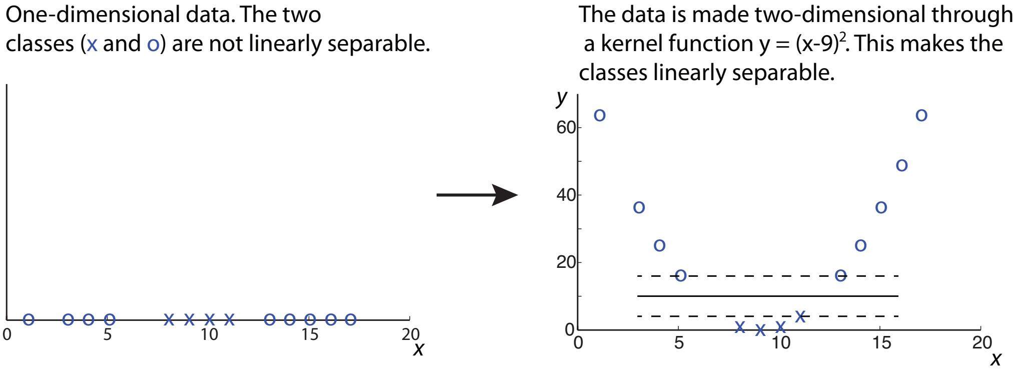


Figure 4.15: The optimal hyperplane (which in this 2-dimensional case means a line) is located between the *support vectors*. The classes are named -1 and 1, corresponding to the way in which this hyperplane is derived, where the two margins (dashed lines) are found -1 and 1 times a certain vector from the hyperplane [Duda et al., 2000].


One-dimensional data. The two classes (x and o) are not linearly separable.

The data is made two-dimensional through a kernel function y = (x-9)². This makes the classes linearly separable.

Figure 4.16: The two classes in the one-dimensional data in the left plot are not linearly separable. By adding another dimension $y = (x-9)^2$ it is possible to identify support vectors.

4.4.2 Validating the Classifier

After the training process the performance of the classifier is evaluated by classifying the instances in the validation set. The evaluation can be measured using terms from the field of document retrieval, namely *precision* and *recall* [Salton and Lesk, 1968]. Continuing with the fruit classification example above, let us say that we want to retrieve all the apples from a fruit basket. We pick fruit from the basket; mostly apples but also a few oranges. We fail to notice some of the apples in the basket. *Precision* then denotes the ratio between the number of apples picked and the total number of fruits we picked (including oranges). *Recall* denotes the ratio between the number of apples picked and the total number of apples that were present in the basket in the first place.

Applied to computer classification, this measure shows correctly classified instances rather than correctly retrieved documents (or fruit), and we get precision and recall measures for each class. We define *class precision* (CP) and *class recall* (CR) for class i as:

$$\mathrm{CP}_i = \frac{||R_i \cap A_i||}{||A_i||} \qquad \text{and} \qquad \mathrm{CR}_i = \frac{||R_i \cap A_i||}{||R_i||},$$

where $||A_i||$ denotes the number of examples classified as i, and $||R_i||$ denotes the total numbers of examples in class i. In other words CP denotes the ratio between correctly classified examples and all the examples the classifier *predicted* to be in the specific class. CR denotes the ratio between correctly classified examples and the *true* number of examples in class i. Figure 4.17 shows how both measures are necessary to get a good assessment of the performance of the classifier: 100 % class precision could mean that the class has been drawn too narrowly, and a 100 % class recall could mean that the class has been defined too broadly.

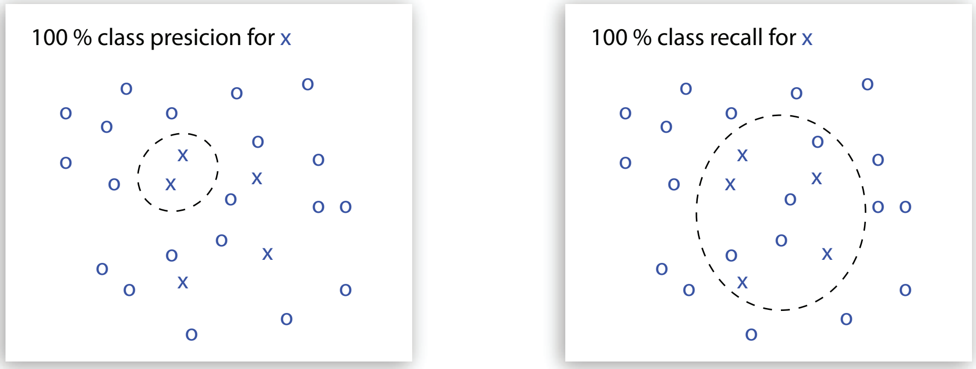


Figure 4.17: In the figure to the left 100 % class precision is obtained. However, several examples that should have been included are left out. To the right all the examples have been included in the classification. However, a number of incorrect examples are also included.

When the data set is of limited size a technique called *cross-validation* can be used to obtain a larger number of examples in the validation set [Duda et al., 2000]. That is, multiple classifications and validations are performed and the examples present in the validation set are different each time. In my experiments I applied the *leave-one-out* principle which entails using the entire data set but one example for training the classifier, and subsequently performing validation with the remaining example. The process is repeated as many times as there are examples in the data set, such that each example is used once for validation.

More detailed results than the precision and recall are obtained by inspecting the classifier results in a *confusion matrix*. This matrix shows the distribution of the examples in the validation set and how they were classified. An example of what the confusion matrix looks like is given in Table 4.1. Systematic classification errors may be revealed by the confusion matrix and such errors may suggest that there are similarities between classes. Examples of how this can be applied to sound and motion analysis will be given in the included Papers V and VIII.

Table 4.1: Confusion matrix showing a classification result. Each row shows the classifications (predictions) made by the classifier and each column shows the actual classes of the examples. The correctly classified examples are found along the diagonal marked in grey. This particular table suggests that classes 1 and 3 have some similarities.

	True 1	True 2	True 3	Class Precision
Predicted 1	6	0	5	55 %
Predicted 2	1	10	1	83 %
Predicted 3	3	0	4	57 %
Class Recall	60 %	100 %	40 %	

4.5 Summary

This chapter has introduced various methods of analysing correspondences between sound and motion. Sound and action objects can be described with time-varying features, meaning features that describe how the sound or motion evolves at regular time-intervals. They can also be described by global features, meaning a single value or typological description that describes the entire object.

The chapter presented techniques for visualising motion data and how the visualisations can be applied to obtain an overview of general tendencies within a single motion recording or a set of recordings. The visualisations may be useful in combination with statistical tests, such as *t*-tests and ANOVAs, which can be applied to test the significance of tendencies in the data. Furthermore, the chapter examined how various correlation measures can be applied to evaluate the correlation between sound and motion features. While correlation coefficients can usually be tested for statistical significance, this is not recommended for continuous sound and motion features given the serial nature of the data. Finally, the use of a computer-based classifier was introduced, with an example of how a confusion matrix can be analysed to get an indication of similar classes.

Bibliography

C. Alain and S. Arnott. Selectively attending to auditory objects. *Frontiers in Bioscience*, 5: 202–212, 2000.

E. Altenmüller, M. Wiesendanger, and J. Kesselring, editors. *Music, Motor Control and the Brain*. Oxford University Press, Oxford, New York, 2006.

R. Ashley. Musical pitch space across modalities: Spatial and other mappings through language and culture. In *Proceedings of the 8th International Conference on Music Perception and Cognition*, pages 64–71, Evenston, IL., USA, 2004.

A. Baltazar, C. Guedes, B. Pennycook, and F. Gouyon. A real-time human body skeletonization algorithm for max/msp/jitter. In *Proceedings of the International Computer Music Conference*, pages 96–99, New York, 2010.

E. Bekkedal. Music Kinection: Sound and Motion in Interactive Systems. Master's thesis, University of Oslo (to appear), 2012.

F. Bevilacqua, J. Ridenour, and D. J. Cuccia. 3D motion capture data: motion analysis and mapping to music. In *Proceedings of the Workshop/Symposium on Sensing and Input for Media-centric Systems*, University of California, Santa Barbara, 2002.

G. Bishop, G. Welch, and B. Allen. Tracking: Beyond 15 minutes of thought. *SIGGRAPH 2001 Courses*, 2001. URL http://www.cs.unc.edu/~tracker/ref/s2001/tracker/ (Accessed June 24, 2012).

P. Boersma and D. Weenink. Praat: doing phonetics by computer (software). version 5.3.19, 2012. URL http://www.praat.org/ (Accessed June 29, 2012).

M. Borga. Canonical correlation: a tutorial. Technical report, Linköping University, 2001.

A. Bouënard, M. M. Wanderley, and S. Gibet. Analysis of Percussion Grip for Physically Based Character Animation. In *Proceedings of the International Conference on Enactive Interfaces*, pages 22–27, Pisa, 2008.

A. S. Bregman. *Auditory Scene Analysis: The Perceptual Organization of Sound*. MIT Press, Cambridge, MA, 1990.

A. S. Bregman and P. Ahad. Demonstrations of auditory scene analysis: The perceptual organization of sound. Audio CD and booklet, Distributed by MIT Press, 1996.

J. Bresson and M. Schumacher. Representation and interchange of sound spatialization data for compositional applications. In *Proceedings of the International Computer Music Conference*, pages 83–87, Huddersfield, 2011.

I. Bukvic, T. Martin, E. Standley, and M. Matthews. Introducing L2Ork: Linux Laptop Orchestra. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 170–173, Sydney, 2010.

B. Burger, M. R. Thompson, G. Luck, S. Saarikallio, and P. Toiviainen. Music moves us: Beat-related musical features influence regularity of music-induced movement. In *Proceedings of the 12th International Conference on Music Perception and Cognition (ICMPC) and 8th Triennial Conference of the European Society for the Cognitive Sciences of Music (ESCOM)*, pages 183–187, Thessaloniki, 2012.

C. Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

C. Cadoz and M. M. Wanderley. Gesture — Music. In M. M. Wanderley and M. Battier, editors, *Trends in Gestural Control of Music*, pages 71–94. Ircam—Centre Pompidou, Paris, France, 2000.

A. Camurri and T. B. Moeslund. Visual gesture recognition. from motion tracking to expressive gesture. In R. I. Godøy and M. Leman, editors, *Musical Gestures: Sound, Movement, and Meaning*, pages 238–263. Routledge, 2010.

A. Camurri, S. Hashimoto, M. Ricchetti, A. Ricci, K. Suzuki, R. Trocca, and G. Volpe. EyesWeb: Toward gesture and affect recognition in interactive dance and music systems. *Computer Music Journal*, 24(1):57–69, 2000.

A. Camurri, I. Lagerlöf, and G. Volpe. Recognizing emotion from dance movement: comparison of spectator recognition and automated techniques. *International Journal of Human-Computer Studies*, 59(1–2):213–225, 2003.

A. Camurri, B. Mazzarino, and G. Volpe. Analysis of expressive gesture: The EyesWeb expressive gesture processing library. In A. Camurri and G. Volpe, editors, *Gesture-based Communication in Human-Computer Interaction*, volume 2915 of *LNAI*, pages 460–467. Springer, Berlin Heidelberg, 2004.

A. Camurri, G. Volpe, G. De Poli, and M. Leman. Communicating expressiveness and affect in multimodal interactive systems. *Multimedia, IEEE*, 12(1):43 – 53, 2005.

C. Cannam, C. Landone, and M. Sandler. Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files. In *Proceedings of ACM Multimedia*, pages 1467–1468, Firenze, Italy, October 2010.

B. Caramiaux, F. Bevilacqua, and N. Schnell. Towards a gesture-sound cross-modal analysis. In S. Kopp and I. Wachsmuth, editors, *Gesture in Embodied Communication and Human-Computer Interaction*, volume 5934 of *Lecture Notes in Computer Science*, pages 158–170. Springer, Berlin Heidelberg, 2010.

B. Caramiaux, F. Bevilacqua, and N. Schnell. Sound selection by gestures. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 329–330, Oslo, 2011.

A. Chandra, K. Nymoen, A. Voldsund, A. R. Jensenius, K. Glette, and J. Torresen. Enabling participants to play rhythmic solos within a group via auctions. In *Proceedings of the 9th International Symposium on Computer Music Modelling and Retrieval*, pages 674–689, London, 2012.

C. Chang and C. Lin. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27, 2011.

J. Chowning. Perceptual fusion and auditory perspective. In P. R. Cook, editor, *Music, Cognition, and Computerized Sound*, pages 261–275. MIT Press, Cambridge, MA, USA, 1999.

M. Ciglar. An Ultrasound Based Instrument Generating Audible and Tactile Sound. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 19–22, Sydney, 2010.

E. F. Clarke. *Ways of listening: An ecological approach to the perception of musical meaning*. Oxford University Press, New York, 2005.

A. W. Cox. *The Metaphoric Logic of Musical Motion and Space*. PhD thesis, University of Oregon, 1999.

A. W. Cox. Hearing, feeling, grasping gestures. In A. Gritten and E. King, editors, *Music and gesture*, pages 45–60. Ashgate, Aldershot, UK, 2006.

S. Dahl. The playing of an accent – preliminary observations from temporal and kinematic analysis of percussionists. *Journal of New Music Research*, 29(3):225–233, 2000.

S. Dahl. Playing the accent-comparing striking velocity and timing in an ostinato rhythm performed by four drummers. *Acta Acustica united with Acustica*, 90(4):762–776, 2004.

R. B. Dannenberg, S. Cavaco, E. Ang, I. Avramovic, B. Aygun, J. Baek, E. Barndollar, D. Duterte, J. Grafton, R. Hunter, C. Jackson, U. Kurokawa, D. Makuck, T. Mierzejewski, M. Rivera, D. Torres, and A. Y. and. The carnegie mellon laptop orchestra. In *Proceedings of the International Computer Music Conference*, pages 340–343, Copenhagen, 2007.

B. De Gelder and P. Bertelson. Multisensory integration, perception and ecological validity. *Trends in Cognitive Sciences*, 7(10):460–467, 2003.

S. de Laubier. The meta-instrument. *Computer Music Journal*, 22(1):25–29, 1998.

S. de Laubier and V. Goudard. Meta-instrument 3: a look over 17 years of practice. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 288–291, Paris, France, 2006.

Y. de Quay, S. Skogstad, and A. Jensenius. Dance jockey: Performing electronic music by dancing. *Leonardo Music Journal*, pages 11–12, 2011.

C. Dobrian and F. Bevilacqua. Gestural control of music: using the vicon 8 motion capture system. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 161–163, Montreal, 2003.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2000.

G. Eckel and D. Pirro. On artistic research in the context of the project embodied generative music. In *Proceedings of the International Computer Music Conference*, pages 541–544, Montreal, 2009.

Z. Eitan and R. Granot. Musical parameters and images of motion. In *Proceedings of the Conference on Interdisciplinary Musicology*, pages 15–18, Graz, 2004.

Z. Eitan and R. Y. Granot. How music moves: Musical parameters and listeners' images of motion. *Music Perception*, 23(3):pp. 221–248, 2006.

Z. Eitan and R. Timmers. Beethoven's last piano sonata and those who follow crocodiles: Cross-domain mappings of auditory pitch in a musical context. *Cognition*, 114(3):405 – 422, 2010.

M. R. Every. *Separation of musical sources and structure from single-channel polyphonic recordings*. PhD thesis, University of York, 2006.

G. Fant. Speech analysis and synthesis. Technical report, Royal Institute of Technology, Stockholm, 1961.

S. Fels and G. Hinton. Glove-talk: A neural network interface between a data-glove and a speech synthesizer. *IEEE Trans. Neural Networks*, 4(1):2–8, 1993.

R. Fischman. Back to the parlour. *Sonic Ideas*, 3(2):53–66, 2011.

A. Freed, D. McCutchen, A. Schmeder, A. Skriver, D. Hansen, W. Burleson, C. Nørgaard, and A. Mesker. Musical applications and design techniques for the gametrak tethered spatial position controller. In *Proceedings of the 6th Sound and Music Computing Conference*, pages 189–194, Porto, 2009.

B. Freedman, A. Shpunt, M. Machline, and Y. Arieli. Depth mapping using projected patterns. Patent Application, 2010. US 12522171.

V. Gallese, L. Fadiga, L. Fogassi, and G. Rizzolatti. Action recognition in the premotor cortex. *Brain*, 119(2):593–609, 1996.

J. J. Gibson. *The Ecological Approach to Visual Perception*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1979.

N. Gillian, P. Coletta, B. Mazzarino, and M. Ortiz. Techniques for data acquisition and multimodal analysis of emap signals. EU FP7 ICT FET SIEMPRE, Project No. 250026, Deliverable report 3.1, May 2011.

R. I. Godøy. Motor-mimetic music cognition. *Leonardo Music Journal*, 36(4):317–319, 2003.

R. I. Godøy. Gestural imagery in the service og musical imagery. In A. Camurri and G. Volpe, editors, *Gesture-Based Communication in Human-Computer Interaction: 5th International Gesture Workshop, GW 2003, Genova, Italy, April 15-17, 2003, Selected Revised Papers, LNAI 2915*, pages 55–62. Springer, Berlin Heidelberg, 2004.

R. I. Godøy. Gestural-sonorous objects: embodied extensions of Schaeffer's conceptual apparatus. *Organised Sound*, 11(02):149–157, 2006.

R. I. Godøy. Gestural affordances of musical sound. In R. I. Godøy and M. Leman, editors, *Musical Gestures: Sound, Movement, and Meaning*, chapter 5, pages 103–125. Routledge, New York, 2010.

R. I. Godøy. Sonic feature timescales and music-related actions. In *Proceedings of Forum Acusticum*, pages 609–613, Aalborg, 2011. European Acoustics Association.

R. I. Godøy and A. R. Jensenius. Body movement in music information retrieval. In *Proceedings of the 10th International Society for Music Information Retrieval Conference*, pages 45–50, Kobe, Japan, 2009.

R. I. Godøy and M. Leman, editors. *Musical Gestures: Sound, Movement, and Meaning*. Routledge, New York, 2010.

R. I. Godøy, E. Haga, and A. R. Jensenius. Playing "air instruments": Mimicry of sound-producing gestures by novices and experts. In S. Gibet, N. Courty, and J.-F. Kamp, editors, *International Gesture Workshop. Revised Selected Papers*, volume 3881/2006, pages 256–267. Springer, Berlin Heidelberg, 2006a.

R. I. Godøy, E. Haga, and A. R. Jensenius. Exploring music-related gestures by sound-tracing. a preliminary study. In *2nd ConGAS International Symposium on Gesture Interfaces for Multimedia Systems*, Leeds, UK, 2006b.

R. I. Godøy, A. R. Jensenius, and K. Nymoen. Chunking in music by coarticulation. *Acta Acustica united with Acustica*, 96(4):690–700, 2010.

R. I. Godøy, A. R. Jensenius, A. Voldsund, K. Glette, M. E. Høvin, K. Nymoen, S. A. Skogstad, and J. Torresen. Classifying music-related actions. In *Proceedings of 12th International Conference on Music Perception and Cognition*, pages 352–357, Thessaloniki, 2012.

J. M. Grey. Multidimensional perceptual scaling of musical timbres. *The Journal of the Acoustical Society of America*, 61(5):1270–1277, 1977.

A. Gritten and E. King, editors. *Music and gesture*. Ashgate, Aldershot, UK, 2006.

A. Gritten and E. King, editors. *New perspectives on music and gesture*. Ashgate, Aldershot, UK, 2011.

F. Grond, T. Hermann, V. Verfaille, and M. Wanderley. Methods for effective sonification of clarinetists' ancillary gestures. In S. Kopp and I. Wachsmuth, editors, *Gesture in Embodied Communication and Human-Computer Interaction*, volume 5934 of *Lecture Notes in Computer Science*, pages 171–181. Springer, Berlin Heidelberg, 2010.

C. Guedes. Extracting musically-relevant rhythmic information from dance movement by applying pitch-tracking techniques to a video signal. In *Proceedings of the Sound and Music Computing Conference SMC06*, pages 25–33, Marseille, 2006.

J. Hagedorn, S. Satterfield, J. Kelso, W. Austin, J. Terrill, and A. Peskin. Correction of location and orientation errors in electromagnetic motion tracking. *Presence: Teleoperators and Virtual Environments*, 16(4):352–366, 2007.

M. Halle and K. Stevens. Speech recognition: A model and a program for research. *IRE Transactions on Information Theory*, 8(2):155 –159, 1962.

B. Haslinger, P. Erhard, E. Altenmüller, U. Schroeder, H. Boecker, and A. Ceballos-Baumann. Transmodal sensorimotor networks during action observation in professional pianists. *Journal of cognitive neuroscience*, 17(2):282–293, 2005.

J. Haueisen and T. R. Knösche. Involuntary motor activity in pianists evoked by music perception. *Journal of cognitive neuroscience*, 13(6):786–792, 2001.

K. Hayafuchi and K. Suzuki. Musicglove: A wearable musical controller for massive media library. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 241–244, Genova, 2008.

H. Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936.

A. Hunt, M. M. Wanderley, and M. Paradis. The importance of parameter mapping in electronic instrument design. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 1–6, Singapore, 2002.

D. Huron. The new empiricism: Systematic musicology in a postmodern age. *The 1999 Ernest Bloch Lectures*, 1999. URL http://www.musicog.ohio-state.edu/Music220/Bloch.lectures/3.Methodology.html (Accessed June 6, 2012).

D. Huron, S. Dahl, and R. Johnson. Facial expression and vocal pitch height: Evidence of an intermodal association. *Empirical Musicology Review*, 4(3):93–100, 2009.

E. Husserl. *The Phenomenology of Internal Time Consciousness*. (trans. J.S. Churchill.) Indiana University Press, Bloomington, 1964.

G. Iddan and G. Yahav. 3d imaging in the studio (and elsewhere). In B. D. Corner, J. H. Nurre, and R. P. Pargas, editors, *Three-Dimensional Image Capture and Applications IV*, volume 4298 of *Proceedings of SPIE*, pages 48–55, 2001.

J. Impett. A meta-trumpet(er). In *Proceedings of the International Computer Music Conference*, pages 147–150, Aarhus, 1994.

H. Ip, K. Law, and B. Kwong. Cyber composer: Hand gesture-driven intelligent music composition and generation. In *Proceedings of the IEEE 11th International Multimedia Modelling Conference*, pages 46–52. Melbourne, 2005.

A. R. Jensenius. *Action–Sound : Developing Methods and Tools for Studying Music-Related Bodily Movement*. PhD thesis, University of Oslo, 2007a.

A. R. Jensenius. GDIF Development at McGill. Short Term Scientific Mission Report, COST Action 287 ConGAS. McGill University, Montreal 2007b. URL http://urn.nb.no/URN:NBN:no-21768 (Accessed October 10, 2012).

A. R. Jensenius. Motion capture studies of action-sound couplings in sonic interaction. Short Term Scientific Mission Report, COST Action IC0601 SID. Royal Institute of Technology, Stockholm, 2009. URL http://urn.nb.no/URN:NBN:no-26163 (Accessed October 10, 2012).

A. R. Jensenius. Some video abstraction techniques for displaying body movement in analysis and performance. *Leonardo Music Journal* (to appear), 2012a.

A. R. Jensenius. Evaluating how different video features influence the quality of resultant motiongrams. In *Proceedings of the Sound and Music Computing Conference*, pages 467–472, Copenhagen, 2012b.

A. R. Jensenius. Motion-sound interaction using sonification based on motiongrams. In *Proceedings of The Fifth International Conference on Advances in Computer-Human Interactions*, pages 170–175, Valencia, 2012c.

A. R. Jensenius and K. A. V. Bjerkestrand. Exploring micromovements with motion capture and sonification. In A. L. Brooks et al., editors, *Arts and Technology*, volume 101 of *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, pages 100–107. Springer, Berlin Heidelberg, 2012.

A. R. Jensenius and A. Voldsund. The music ball project: Concept, design, development, performance. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 300–303, Ann Arbor, 2012.

A. R. Jensenius, R. I. Godøy, and M. M. Wanderley. Developing tools for studying musical gestures within the max/msp/jitter environment. In *Proceedings of the International Computer Music Conference*, pages 282–285. Barcelona, 2005.

A. R. Jensenius, R. Koehly, and M. Wanderley. Building low-cost music controllers. In R. Kronland-Martinet, T. Voinier, and S. Ystad, editors, *Computer Music Modeling and Retrieval*, volume 3902 of *Lecture Notes in Computer Science*, pages 123–129. Springer, Berlin Heidelberg, 2006a.

A. R. Jensenius, T. Kvifte, and R. I. Godøy. Towards a gesture description interchange format. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 176–179, Paris, 2006b.

A. R. Jensenius, A. Camurri, N. Castagne, E. Maestre, J. Malloch, D. McGilvray, D. Schwarz, and M. Wright. Panel: the need of formats for streaming and storing music-related movement and gesture data. In *Proceedings of the International Computer Music Conference*, pages 13–16, Copenhagen, 2007.

A. R. Jensenius, K. Nymoen, and R. I. Godøy. A multilayered GDIF-based setup for studying coarticulation in the movements of musicians. In *Proceedings of the International Computer Music Conference*, pages 743–746, Belfast, 2008.

A. R. Jensenius, S. A. Skogstad, K. Nymoen, J. Torresen, and M. E. Høvin. Reduced displays of multidimensional motion capture data sets of musical performance. In *Proceedings of ESCOM 2009: 7th Triennial Conference of the European Society for the Cognitive Sciences of Music*, Jyväskylä, Finland, 2009.

A. R. Jensenius, M. M. Wanderley, R. I. Godøy, and M. Leman. Musical gestures: Concepts and methods in research. In R. I. Godøy and M. Leman, editors, *Musical Gestures: Sound, Movement, and Meaning*. Routledge, New York, 2010.

A. R. Jensenius, K. Nymoen, S. A. Skogstad, and A. Voldsund. How still is still? a study of the noise-level in two infrared marker-based motion capture systems. In *Proceedings of the Sound and Music Computing Conference*, pages 258–263, Copenhagen, 2012.

M. R. Jones. Music perception: Current research and future directions. In M. Riess Jones, R. R. Fay, and A. N. Popper, editors, *Music Perception*, volume 36 of *Springer Handbook of Auditory Research*, pages 1–12. Springer, New York, 2010.

S. Jordà. Afasia: the Ultimate Homeric One-man-multimedia-band. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 102–107, Dublin, 2002.

A. Kapur, G. Tzanetakis, N. Virji-Babul, G. Wang, and P. R. Cook. A framework for sonification of vicon motion capture data. In *Proceedings of the International Conference on Digital Audio Effects*, pages 47–52, Madrid, 2005.

S. T. Klapp and R. J. Jagacinski. Gestalt principles in the control of motor action. *Psychological Bulletin*, 137(3):443–462, 2011.

M. Klingbeil. Software for spectral analysis, editing, and synthesis. In *Proceedings of the International Computer Music Conference*, pages 107–110, Barcelona, 2005.

T. Koerselman, O. Larkin, and K. Ng. The mav framework: Working with 3d motion data in max msp / jitter. In *Proceedings of the 3rd International Conference on Automated Production of Cross Media Content for Multi-channel Distribution (AXMEDIS 2007). Volume for Workshops, Tutorials, Applications and Industrial, i-Maestro 3rd Workshop*, Barcelona, 2007.

E. Kohler, C. Keysers, M. A. Umiltà, L. Fogassi, V. Gallese, and G. Rizzolatti. Hearing sounds, understanding actions: Action representation in mirror neurons. *Science*, 297(5582):846–848, 2002.

D. Kohn and Z. Eitan. Seeing sound moving: Congruence of pitch and loudness with human movement and visual shape. In *Proceedings of the 12th International Conference on Music Perception and Cognition (ICMPC) and 8th Triennial Conference of the European Society for the Cognitive Sciences of Music (ESCOM)*, pages 541–546, Thessaloniki, 2012.

M. Kozak, K. Nymoen, and R. I. Godøy. The effects of spectral features of sound on gesture type and timing. In E. Efthimiou, G. Kouroupetroglou, and S.-E. Fotinea, editors, *Gesture and Sign Language in Human-Computer Interaction and Embodied Communication. 9th International Gesture Workshop - GW 2011, May 2011, Athens, Greece. Revised selected papers.*, volume 7206 of *Lecture Notes in Computer Science/LNAI*. Springer (to appear), Berlin Heidelberg, 2012.

L. Kozlowski and J. Cutting. Recognizing the sex of a walker from a dynamic point-light display. *Attention, Perception, & Psychophysics*, 21(6):575–580, 1977.

M. Kussner. Creating shapes: musicians' and non-musicians' visual representations of sound. In U. Seifert and J. Wewers, editors, *Proceedings of SysMus11: Fourth International Conference of students of Systematic Musicology*, Osnabrück, epOs-Music (to appear), 2012.

T. Kvifte. *Instruments and the Electronic Age*. Solum, Oslo, 1989.

A. Lahav, E. Saltzman, and G. Schlaug. Action representation of sound: audiomotor recognition network while listening to newly acquired actions. *The journal of neuroscience*, 27(2):308–314, 2007.

G. Lakoff and M. Johnson. *Metaphors We Live By*. University of Chicago Press, Chicago, IL., 1980.

F. Langheim, J. Callicott, V. Mattay, J. Duyn, and D. Weinberger. Cortical systems associated with covert music rehearsal. *NeuroImage*, 16(4):901–908, 2002.

O. Lartillot, P. Toiviainen, and T. Eerola. A matlab toolbox for music information retrieval. In C. Preisach, H. Burkhardt, L. Schmidt-Thieme, and R. Decker, editors, *Data Analysis, Machine Learning and Applications*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 261–268. Springer, Berlin Heidelberg, 2008.

M. Leman. *Embodied Music Cognition and Mediation Technology*. The MIT Press, 2008.

M. Leman and L. A. Naveda. Basic gestures as spatiotemporal reference frames for repetitive dance/music patterns in samba and charleston. *Music Perception*, 28(1):71–91, 2010.

G. Leslie, D. Schwarz, O. Warusfel, F. Bevilacqua, B. Zamborlin, P. Jodlowski, and N. Schnell. Grainstick: A collaborative, interactive sound installation. In *Proceedings of the International Computer Music Conference*, pages 123–126, New York, 2010.

A. M. Liberman and I. G. Mattingly. The motor theory of speech perception revised. *Cognition*, 21(1):1 – 36, 1985.

E. Lin and P. Wu. Jam master, a music composing interface. In *Proceedings of Human Interface Technologies*, pages 21–28, Vancouver, BC, 2000.

G. Luck, S. Saarikallio, B. Burger, M. Thompson, and P. Toiviainen. Effects of the big five and musical genre on music-induced movement. *Journal of Research in Personality*, 44(6):714 – 720, 2010a.

G. Luck, P. Toiviainen, and M. R. Thompson. Perception of expression in conductors' gestures: A continuous response study. *Music Perception*, 28(1):47–57, 2010b.

P.-J. Maes, M. Leman, M. Lesaffre, M. Demey, and D. Moelants. From expressive gesture to sound. *Journal on Multimodal User Interfaces*, 3:67–78, 2010.

E. Maestre, J. Janer, M. Blaauw, A. Pérez, and E. Guaus. Acquisition of violin instrumental gestures using a commercial EMF tracking device. In *Proceedings of the International Computer Music Conference*, pages 386–393, Copenhagen, 2007.

J. Malloch, S. Sinclair, and M. M. Wanderley. From controller to sound: Tools for collaborative development of digital musical instruments. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 66–69, New York, 2007.

T. Marrin and R. Picard. The "conductor's jacket": A device for reocording expressive musical gestures. In *Proceedings of the International Computer Music Conference*, pages 215–219, Ann Arbor, 1998.

M. Marshall, M. Rath, and B. Moynihan. The virtual bodhran: the vodhran. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 1–2, Dublin, 2002.

M. Marshall, N. Peters, A. R. Jensenius, J. Boissinot, M. M. Wanderley, and J. Braasch. On the development of a system for gesture control of spatialization. In *Proceedings of the International Computer Music Conference*, pages 360–366, New Orleans, 2006.

M. Mathews. What is loudness? In P. R. Cook, editor, *Music, Cognition, and Computerized Sound*, pages 71–78. MIT Press, Cambridge, MA, USA, 1999a.

M. Mathews. Introdction to timbre. In P. R. Cook, editor, *Music, Cognition, and Computerized Sound*, pages 79–87. MIT Press, Cambridge, MA, USA, 1999b.

S. McAdams. *Spectral Fusion, Spectral Parsing, and the Formation of Auditory Images*. PhD thesis, Stanford University, 1984.

S. McAdams, S. Winsberg, S. Donnadieu, G. Soete, and J. Krimphoff. Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes. *Psychological Research*, 58:177–192, 1995.

H. McGurk and J. MacDonald. Hearing lips and seeing voices. *Nature*, 264(5588):746–748, 12 1976.

I. Meister, T. Krings, H. Foltys, B. Boroojerdi, M. Müller, R. Töpper, and A. Thron. Playing piano in the mind—an fmri study on music imagery and performance in pianists. *Cognitive Brain Research*, 19(3):219 – 228, 2004.

T. Melzer, M. Reiter, and H. Bischof. Appearance models based on kernel canonical correlation analysis. *Pattern Recognition*, 36(9):1961–1971, 2003.

A. Merer, S. Ystad, R. Kronland-Martinet, and M. Aramaki. Semiotics of sounds evoking motions: Categorization and acoustic features. In R. Kronland-Martinet, S. Ystad, and K. Jensen, editors, *Computer Music Modeling and Retrieval. Sense of Sounds*, number 4969 in Lecture Notes in Computer Science, pages 139–158. Springer, Berlin Heidelberg, 2008.

I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler. Yale: Rapid prototyping for complex data mining tasks. In L. Ungar, M. Craven, D. Gunopulos, and T. Eliassi-Rad, editors, *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 935–940, New York, 2006.

G. A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63:81–97, 1956.

E. R. Miranda and M. Wanderley. *New Digital Musical Instruments: Control And Interaction Beyond the Keyboard*. A-R Editions, Inc., Middleton, WI, 2006.

T. Mitchell and I. Heap. Soundgrasp: A gestural interface for the performance of live music. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 465–468, Oslo, 2011.

D. S. Moore and G. P. McCabe. *Introduction to the practice of statistics*. W.H. Freeman and Company, New York, 5th edition, 2006.

M. Müller. *Information retrieval for music and motion*. Springer, Berlin Heidelberg, 2007.

M. Müller and T. Röder. Motion templates for automatic classification and retrieval of motion capture data. In *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 137–146, Aire-la-Ville, Switzerland, 2006.

K. Ng, O. Larkin, T. Koerselman, B. Ong, D. Schwarz, and F. Bevilacqua. The 3D augmented mirror: motion analysis for string practice training. In *Proceedings of the International Computer Music Conference*, pages 53–56, Copenhagen, 2007.

L. Nigay and J. Coutaz. A design space for multimodal systems: concurrent processing and data fusion. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*, pages 172–178, Amsterdam, 1993.

M. Nusseck and M. Wanderley. Music and motion-how music-related ancillary body movements contribute to the experience of music. *Music Perception*, 26(4):335–353, 2009.

K. Nymoen. The Nymophone2 – a study of a new multidimensionally controllable musical instrument. Master's thesis, University of Oslo, 2008a.

K. Nymoen. A setup for synchronizing GDIF data using SDIF-files and FTM for Max. Short Term Scientific Mission Report, COST Action IC0601 SID. McGill University, Montreal, 2008b. URL `http://urn.nb.no/URN:NBN:no-20580` (Accessed October 10, 2012).

K. Nymoen, J. Torresen, R. Godøy, and A. R. Jensenius. A statistical approach to analyzing sound tracings. In S. Ystad, M. Aramaki, R. Kronland-Martinet, K. Jensen, and S. Mohanty, editors, *Speech, Sound and Music Processing: Embracing Research in India*, volume 7172 of *Lecture Notes in Computer Science*, pages 120–145. Springer, Berlin Heidelberg, 2012.

J. Oh, J. Herrera, N. J. Bryan, L. Dahl, and G. Wang. Evolving the mobile phone orchestra. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 82–87, Sydney, 2010.

Oxford Dictionaries: "Modality". URL http://oxforddictionaries.com/definition/english/modality (Accessed September 21, 2012).

R. Parncutt. Systematic musicology and the history and future of western musical scholaship. *Journal of Interdisciplinary Music Studies*, 1(1):1–32, 2007.

G. Peeters. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Tech. rep., IRCAM, Paris, 2004.

G. Peeters, B. Giordano, P. Susini, N. Misdariis, and S. McAdams. The timbre toolbox: Extracting audio descriptors from musical signals. *The Journal of the Acoustical Society of America*, 130(5):2902–2916, 2011.

G. Pellegrino, L. Fadiga, L. Fogassi, V. Gallese, and G. Rizzolatti. Understanding motor events: a neurophysiological study. *Experimental Brain Research*, 91:176–180, 1992.

J.-M. Pelletier. cv.jit — Computer Vision for Jitter (software).
URL http://jmpelletier.com/cvjit/ (Accessed June 27, 2012).

N. Peters, T. Lossius, J. Schacher, P. Baltazar, C. Bascou, and T. Place. A stratified approach for sound spatialization. In *Proceedings of 6th Sound and Music Computing Conference*, pages 219–224, Porto, 2009.

J. Pierce. Introduction to pitch perception. In P. R. Cook, editor, *Music, Cognition, and Computerized Sound*, chapter 5. MIT Press, Cambridge, MA, USA, 1999.

T. Place and T. Lossius. Jamoma: A Modular Standard for Structuring Patches in Max. In *Proceedings of the International Computer Music Conference*, pages 143–146, New Orleans, 2006.

T. Place, T. Lossius, A. R. Jensenius, and N. Peters. Flexible control of composite parameters in max/msp. In *Proceedings of the International Computer Music Conference*, pages 233–236, Belfast, 2008.

Polhemus Inc. Liberty brochure. URL http://www.polhemus.com/polhemus_editor/assets/LIBERTY.pdf (Accessed June 25, 2012).

F. Pollick, H. Paterson, A. Bruderlin, and A. Sanford. Perceiving affect from arm movement. *Cognition*, 82(2):B51–B61, 2001.

E. Pöppel. A hierarchical model of temporal perception. *Trends in cognitive sciences*, 1(2): 56–61, 1997.

F. Raab, E. Blood, T. Steiner, and H. Jones. Magnetic position and orientation tracking system. *IEEE Transactions on Aerospace and Electronic Systems*, 15(5):709–718, 1979.

N. Rasamimanana, D. Bernardin, M. Wanderley, and F. Bevilacqua. String bowing gestures at varying bow stroke frequencies: A case study. In M. Sales Dias, S. Gibet, M. Wanderley, and R. Bastos, editors, *Gesture-Based Human-Computer Interaction and Simulation*, volume 5085 of *Lecture Notes in Computer Science*, pages 216–226. Springer, Berlin Heidelberg, 2009.

T. Ringbeck. A 3d time of flight camera for object detection. In *Proceedings of the 8th Conference on Optical 3D Measurement Techniques*, pages 1–10, ETH Zürich, 2007.

J.-C. Risset. Timbre analysis by synthesis: representations, imitations, and variants for musical composition. In G. De Poli, A. Piccialli, and C. Roads, editors, *Representations of musical signals*, pages 7–43. MIT Press, Cambridge, MA, 1991.

D. G. E. Robertson, G. E. Caldwell, J. Hamill, G. Kamen, and S. N. Whittlesey. *Research Methods in Biomechanics*. Human Kinetics, 2004.

D. Roetenberg, H. Luinge, and P. Slycke. Xsens MVN: Full 6DOF Human Motion Tracking Using Miniature Inertial Sensors. Technical report, Xsens Technologies B.V., 2009. URL http://www.xsens.com/images/stories/PDF/MVN_white_paper.pdf (Accessed March 27, 2011).

D. Rosenbaum. *Human motor control*. Academic Press, San Diego, 2001.

F. Rumsey. Spatial quality evaluation for reproduced sound: Terminology, meaning, and a scene-based paradigm. *J. Audio Eng. Soc*, 50(9):651–666, 2002.

G. Salton and M. Lesk. Computer evaluation of indexing and text processing. *Journal of the ACM*, 15(1):8–36, 1968.

P. Schaeffer. *Traité des objets musicaux*. Éditions du Seuil, 1966.

P. Schaeffer and G. Reibel. *Solfège de l'objet sonore*. ORTF, Paris, France, INA-GRM 1998 edition, 1967.

M. Schleidt and J. Kien. Segmentation in behavior and what it can tell us about brain function. *Human nature*, 8(1):77–111, 1997.

N. Schnell, R. Borghesi, D. Schwarz, F. Bevilacqua, and R. Müller. FTM – complex data structures for Max. In *Proceedings of the International Computer Music Conference*, pages 9–12, Barcelona, 2005.

L. Schomaker, J. Nijtmans, A. Camurri, F. Lavagetto, P. Morasso, C. Benoit, T., J. Robert-Ribes, A. Adjoudani, I. Defée, S. Münch, K. Hartung, and J. Blauert. A taxonomy of multimodal interaction in the human information processing system. Report of the esprit project 8579 MIAMI, Nijmegen University, The Netherlands, 1995.

E. Schoonderwaldt and M. Demoucron. Extraction of bowing parameters from violin performance combining motion capture and sensors. *The Journal of the Acoustical Society of America*, 126(5):2695–2708, 2009.

E. Schoonderwaldt, N. Rasamimanana, and F. Bevilacqua. Combining accelerometer and video camera: reconstruction of bow velocity profiles. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 200–203, Paris, 2006.

E. Schubert. Correlation analysis of continuous emotional response to music. *Musicae Scientiae*, Special issue 2001–2002:213–236, 2002.

S. Sentürk, S. W. Lee, A. Sastry, A. Daruwalla, and G. Weinberg. Crossole: A gestural interface for composition, improvisation and performance using kinect. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Ann Arbor, 2012.

B. G. Shinn-Cunningham. Object-based auditory and visual attention. *Trends in Cognitive Sciences*, 12(5):182 – 186, 2008.

R. Siegwart and I. Nourbakhsh. *Introduction to autonomous mobile robots*. MIT Press, 2004.

S. A. Skogstad, A. R. Jensenius, and K. Nymoen. Using IR optical marker based motion capture for exploring musical interaction. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 407–410, Sydney, 2010.

S. A. Skogstad, K. Nymoen, Y. de Quay, and A. R. Jensenius. OSC implementation and evaluation of the Xsens MVN suit. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 300–303, Oslo, 2011.

S. A. Skogstad, S. Holm, and M. Høvin. Designing Digital IIR Low-Pass Differentiators With Multi-Objective Optimization. In *Proceedings of IEEE International Conference on Signal Processing*, Beijing Jiaotong University (to appear), 2012a.

S. A. Skogstad, S. Holm, and M. Høvin. Designing Low Group Delay IIR Filters for Real-Time Applications. In *Proceedings of the International Conference on Engineering and Technology*, Cairo (to appear), 2012b.

S. A. Skogstad, K. Nymoen, Y. de Quay, and A. R. Jensenius. Developing the dance jockey system for musical interaction with the Xsens MVN suit. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 226–229, Ann Arbor, 2012c.

S. W. Smith. *The scientist and engineer's guide to digital signal processing*. California Technical Publishing, San Diego, 1997.

B. Snyder. *Music and Memory. An Introduction*. MIT Press, Cambridge, MA, 2000.

C. Spearman. The proof and measurement of association between two things. *The American journal of psychology*, 15(1):72–101, 1904.

M. Spong, S. Hutchinson, and M. Vidyasagar. *Robot modeling and control*. John Wiley & Sons, New York, 2006.

B. E. Stein and M. A. Meredith. *The merging of the senses*. The MIT Press, Cambridge, MA, 1993.

S. S. Stevens. A metric for the social consensus. *Science*, 151(3710):530–541, 1966.

M. Thompson and G. Luck. Exploring relationships between pianists' body movements, their expressive intentions, and structural elements of the music. *Musicae Scientiae*, 16(1):19–40, 2012.

P. Toiviainen and B. Burger. *MoCap toolbox manual*. University of Jyväskylä, 2011. URL https://www.jyu.fi/music/coe/materials/mocaptoolbox/MCTmanual (Accessed June 29, 2012).

P. Toiviainen, G. Luck, and M. R. Thompson. Embodied meter: Hierarchical eigenmodes in music-induced movement. *Music Perception*, 28(1):59–70, 2010.

S. Trail, M. Dean, G. Odowichuk, T. F. Tavares, P. Driessen, W. A. Schloss, and G. Tzanetakis. Non-invasive sensing and gesture control for pitched percussion hyper-instruments using the kinect. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Ann Arbor, 2012.

D. Trueman, P. Cook, S. Smallwood, and G. Wang. Plork: Princeton laptop orchestra, year 1. In *Proceedings of the International Computer Music Conference*, pages 443–450, New Orleans, 2006.

F. Upham. Limits on the application of statistical correlations to continuous response data. In *Proceedings of the 12th International Conference on Music Perception and Cognition (ICMPC) and 8th Triennial Conference of the European Society for the Cognitive Sciences of Music (ESCOM)*, pages 1037–1041, Thessaloniki, 2012.

L. van Noorden. *Temporal Coherence in the Perception of Tone Sequences*. PhD thesis, Technical University Eindhoven, 1975.

L. van Noorden. The functional role and bio-kinetics of basic and expressive gestures in activation and sonification. In R. I. Godøy and M. Leman, editors, *Musical Gestures: Sound, Movement, and Meaning*, pages 154–179. Routledge, New York, 2010.

L. van Noorden and D. Moelants. Resonance in the perception of musical pulse. *Journal of New Music Research*, 28(1):43–66, 1999.

V. Verfaille, O. Quek, and M. Wanderley. Sonification of musicians' ancillary gestures. In *Proceedings of the International Conference on Auditory Display*, pages 194–197, London, 2006.

G. Vigliensoni and M. M. Wanderley. A quantitative comparison of position trackers for the development of a touch-less musical interface. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 103–108, Ann Arbor, 2012.

B. W. Vines, C. L. Krumhansl, M. M. Wanderley, and D. J. Levitin. Cross-modal interactions in the perception of musical performance. *Cognition*, 101(1):80–113, 2006.

F. Vogt, G. Mccaig, M. A. Ali, and S. S. Fels. Tongue 'n' Groove: An Ultrasound based Music Controller. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 181–185, Dublin, 2002.

J. Vroomen and B. de Gedler. Sound enhances visual perception: Cross-modal effects on auditory organization on vision. *Journal of Experimental Psychology: Human Perception and Performance*, 26(5):1583–1590, 2000.

B. Walker. *Magnitude estimation of conceptual data dimensions for use in sonification*. PhD thesis, Rice University, Houston, TX, 2000.

M. M. Wanderley. Quantitative analysis of non-obvious performer gestures. In I. Wachsmuth and T. Sowa, editors, *Gesture and Sign Language in Human-Computer Interaction*, volume 2298 of *Lecture Notes in Computer Science*, pages 241–253. Springer, Berlin Heidelberg, 2002.

M. M. Wanderley and M. Battier, editors. *Trends in Gestural Control of Music*. IRCAM — Centre Pompidou, Paris, 2000.

M. M. Wanderley and P. Depalle. Gestural control of sound synthesis. In *Proceedings of the IEEE*, volume 92, pages 632–644, 2004.

M. M. Wanderley, B. W. Vines, N. Middleton, C. McKay, and W. Hatch. The musical significance of clarinetists' ancillary gestures: An exploration of the field. *Journal of New Music Research*, 43(1):97–113, 2005.

G. Wang, G. Essl, and H. Penttinen. Do mobile phones dream of electric orchestras. In *Proceedings of the International Computer Music Conference*, Belfast, 2008.

G. Welch and E. Foxlin. Motion tracking: No silver bullet, but a respectable arsenal. *IEEE Computer Graphics and Applications*, 22(6):24–38, 2002.

D. Wessel and M. Wright. Problems and prospects for intimate musical control of computers. *Computer Music Journal*, 26:11–22, 2002.

M. Wright and A. Freed. Open sound control: A new protocol for communicating with sound synthesizers. In *Proceedings of the International Computer Music Conference*, pages 101–104, Thessaloniki, 1997.

M. Wright, A. Chaudhary, A. Freed, D. Wessel, X. Rodet, D. Virolle, R. Woehrmann, and X. Serra. New applications of the sound description interchange format. In *Proceedings of the International Computer Music Conference*, pages 276–279, Ann Arbor, 1998.

M. Yoo, J. Beak, and I. Lee. Creating musical expression using kinect. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 324–325, Oslo, 2011.

M. Zadel, S. Sinclair, and M. Wanderley. Haptic feedback for different strokes using dimple. In *Proceedings of the International Computer Music Conference*, pages 291–294, Montreal, 2009.

R. Zatorre. Music, the food of neuroscience? *Nature*, 434:312–315, 2005.

R. Zatorre and A. Halpern. Mental concerts: musical imagery and auditory cortex. *Neuron*, 47 (1):9–12, 2005.

K. Zou, K. Tuncali, and S. Silverman. Correlation and simple linear regression. *Radiology*, 227 (3):617–628, 2003.