Guide to the Carnegie Mellon University Multimodal Activity (CMU-MMAC) Database

Fernando De la Torre Jessica Hodgins Adam Bargteil
Xavier Martin Justin Macey Alex Collado Pep Beltran

CMU-RI-TR-08-22

April 2008

Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

This document summarizes the technology, procedures, and database organization of the CMU Multi-Modal Activity Database (CMU-MMAC). The CMU-MMAC database contains multimodal measures of the human activity of subjects performing the tasks involved in cooking and food preparation. The CMU-MMAC database was collected in Carnegie Mellon University's Motion Capture Lab. A kitchen was built and to date five subjects have been recorded cooking five different recipes: brownies, pizza, sandwich, salad and scrambled eggs. The following modalities were recorded:

- *Video: (1) Three high spatial resolution (*1024×768*) color video cameras at low temporal resolution (*30 *Hertz). (2) Two low spatial resolution (*640×480*) color video cameras at high temporal resolution (*60 *Hertz). (3) One wearable low spatial resolution (*640×480*) camera at low temporal resolution (*12 *Hertz).*

- *Audio: (1) Five balanced microphones. (2) Wearable watch.*

- *Motion capture: A Vicon motion capture system with* 12 *infrared MX-40 cameras. Each camera records images of 4 megapixel resolution at* 120 *Hertz.*

- *Five 3-axis accelerometers and gyroscopes.*

Several computers were used for recording the various modalities. The computers were synchronized using the Network Time Protocol (NTP).

Contents

1 Motivation

Over the past decade, researchers in computer graphics, computer vision and robotics have begun to work with very large collections of data to model human motion (e.g. [1, 2]). These databases have been used to construct models of human movement for which researchers have found many applications in sports science, medicine, biomechanics, animation of avatars in games or movies, surveillance, better strategies for humanoid robots, and human activity recognition among others. These databases have facilitated research and provided standardized test datasets for algorithms. However, many of these databases are limited by the constrained settings within which they are collected. For instance, current human motion capture databases typically capture the motion of professional actors, athletes, and artists who were brought into the studio for their specific talents, rather than a wide range of individuals performing everyday tasks. As a result, the motions in current motion databases are often performances—examples of finely honed skills, or clear caricatures of ordinary events. Furthermore, most databases capture only one or two sensing modalities (e.g. motion capture/video, audio/video).

The CMU-MMAC database aims to overcome some of the previous limitations by collecting multimodal (audio, video, accelerations, motion capture) samples of human behavior. To capture human behavior in settings that are as natural as possible, we have installed an almost fully operable kitchen and captured the cooking of several meals from start to finish. The kitchen is a very important test bed because food preparation and eating are core elements of daily life and hence essential for a data collection that purports to represent the space of natural human activities. Moreover, the kitchen is key to a number of socially significant applications. For instance, an inability to reliably prepare a balanced diet is often a decisive factor in a move to assisted living for the elderly or cognitively impaired. Finally, the kitchen is a common location for accidents (e.g. fires, cuts, broken dishes). Figure 1 illustrates the location of the sensors and several views of the kitchen constructed with working appliances.

2 Modalities

This section explains the hardware and software components for each modality: video, audio, accelerometers/gyroscopes and optical motion capture.

2.1 Video

The visual information is captured from static and wearable cameras. This section describes the technical details of both imaging systems.

2.1.1 Static cameras

We used five FireWire cameras manufactured by Point Grey Research Inc (see Figure 4.a). Three of these cameras (FL2-08S2C) capture high resolution images (1024×768 pixels) at 30 fps. The other two cameras (FL2-03S2C) have lower resolution (640×480 pixels) but higher frame rate (60 fps) to capture faster motion. Both cameras were full

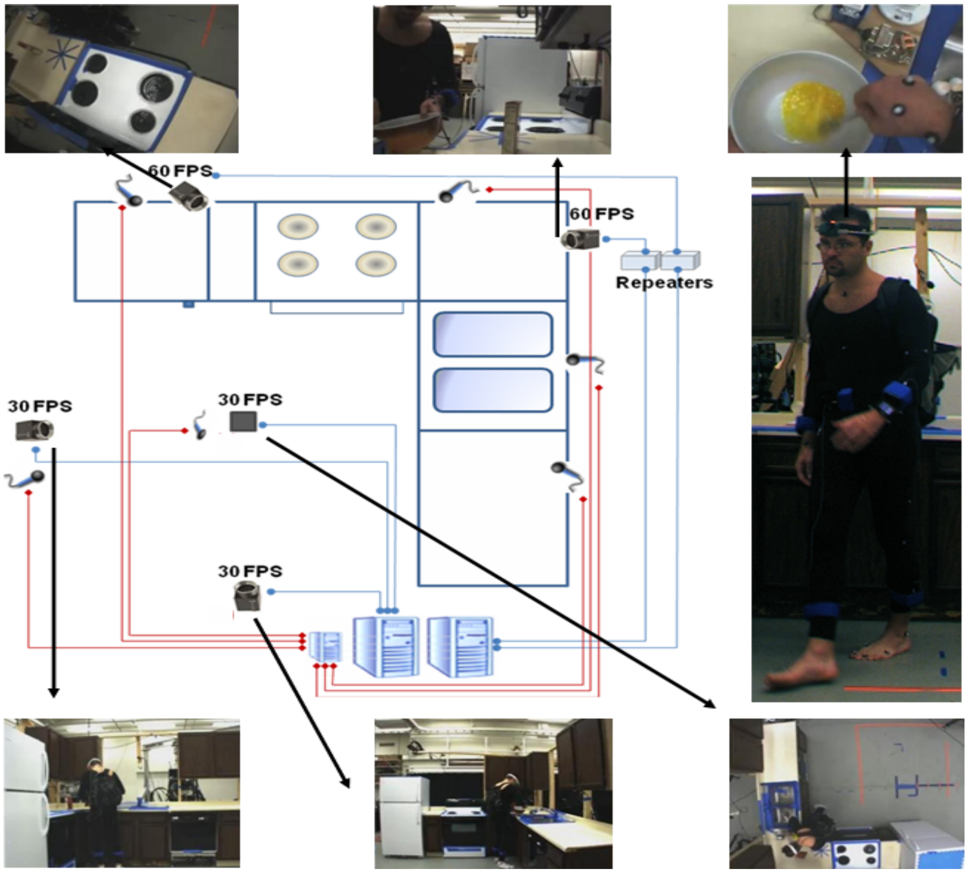


Figure 1: Sensors placement in the kitchen.

native IEEE-1394b (FireWire b) standard compatible, allowing transmission speeds of 800 Mbits/s. The firewire interface enables full frame rate RGB image transmission at the maximum camera resolution. Moreover, the IEEE-1394b protocol describes a timestamp field within the packets header. This timestamp field allows (jointly with the hardware) a precise synchronization among the five cameras (a drift less than 125 micro-seconds).

The main IEEE-1394b limitation is the cable length supported by the standard. Due to the high transmission velocities, severe attenuations of the signal strength are created by the system wires. These attenuations reduce the signal to noise ratio allowing maximal cable lengths of 15 ft to 20 ft. In order to address this problem, we used FireWire b (Unibrain FireRepeater 800) repeaters to regenerate the signal strength.

Another major challenge of the imaging system is to store the uncompressed video information. The space required for the five cameras is

- FL2-03S2C: $640 \times 480 \times 60 \times 2 = 36.84$ Mbytes/sec

- FL2-08S2C: $1024 \times 768 \times 30 \times 3 = 77.78$ Mbytes/sec

The bandwidth required to store the information from the five cameras is about 114.62 Mbytes/sec. Currently, no consumer hard drive exists that can store this information in real-time. In order to minimize the cost of the system, we recorded the camera information with two different computers. To synchronize all the cameras it is necessary to build a FireWire network between the two computers. This network can be set up either following the FireWire a or b specification. The goal of this network is to propagate the timestamp field between the two computers, but not to transmit video data. The network was set up and administrated by means of the MultiSync Point Grey software. Furthermore, we used hard drive configurations in RAID 0 (Redundant Array of Independent Drives mode 0). RAID technology increases bandwidth of the hard drive system using two or more independent units, which work like a single unit in parallel, but multiply the group data rate.

The first computer is a 2xDual Xeon Processor with 4GB of RAM, four 500 GBytes hard drives in RAID 0 configuration and Windows XP 32bit O.S. This computer records three cameras (two cameras 640×480 at 60 fps and one 1024×768 at 30 fps) with 4SIIG FireWire b PCI-Xpress cards. Each card supports one camera, and the fourth card enables synchronization with the other computer. The second computer is 1xCore2Duo Processor with 2 GB RAM, 1x150 Gbytes RAPTOR x 10000rpm + 1x500 Gbytes hard drives, and Windows XP 32bits. This computer records two cameras (1024×768 at 30 fps). Both computers are synchronized with a 1xSIIG FireWire card and the Point Grey MultiSync 1.13 software using Point Grey FlyCapture v1.7 Alpha 2 Drivers.

Figure 2 shows several images from the five static cameras and the wearable camera (lower right image) during the process of making scrambled eggs.

2.1.2 Wearable camera

A Firewire camera, FL2-08S2C (640×480 pixels), is attached to a head lamp whose bulb has been removed and that is placed around the head of the subject (see Figure 1). We recorded the uncompressed video coming from the wearable camera with a Dell-620 laptop using the APIs provided by the software Labview. Additionally, the time stamp of each individual frame is recorded. This laptop also recorded the data coming from the accelerometers/gyroscopes. Labview allows synchronization of the video with the accelerometers/gyroscopes and also provides libraries and support for the accelerometers. A user Labview interface has been designed (see Figure 3) to control the different configuration parameters of the sensors, including the number of sensors, their sample rate, or the resolution of the wearable camera.

2.2 Multi channel audio

The audio modality was captured using five Behringer condenser B-5 Microphones (see Figure 4.b). These high quality Microphones offer different pickup patterns (cardioid or omnidirectional) in order to capture a specific source of sound or a general sample of the room. The microphones use balanced 3-pin XLR connectors, which provide better

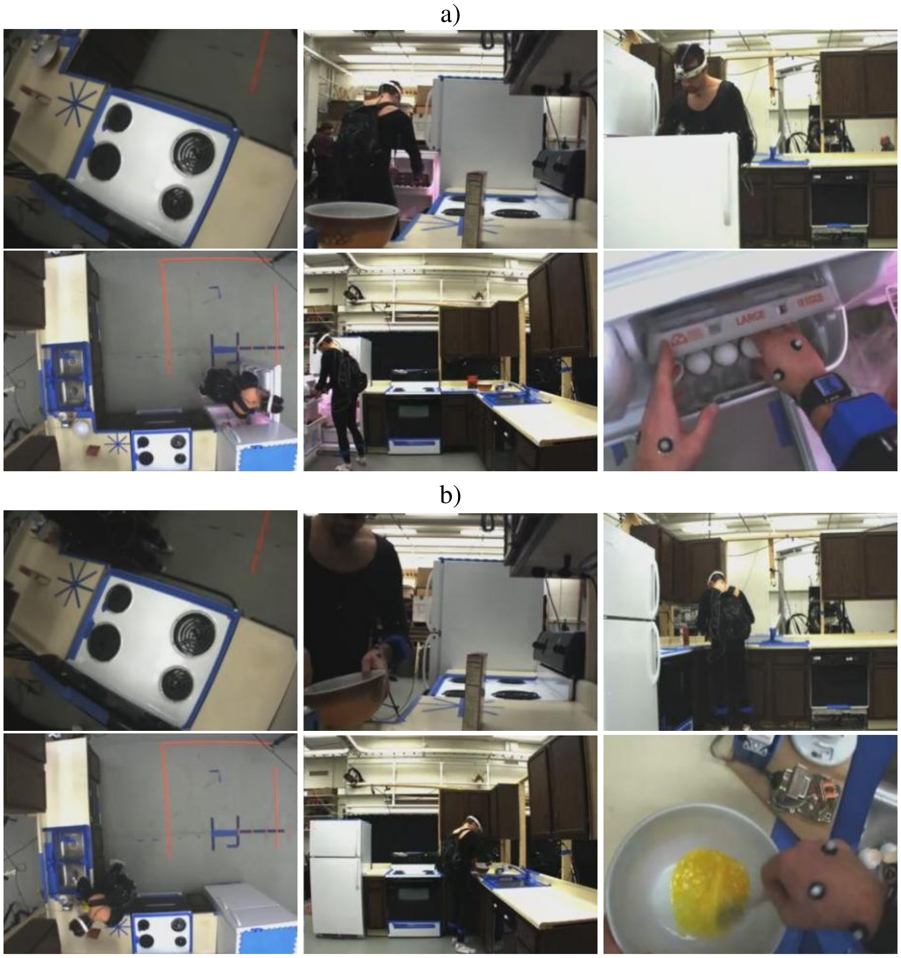
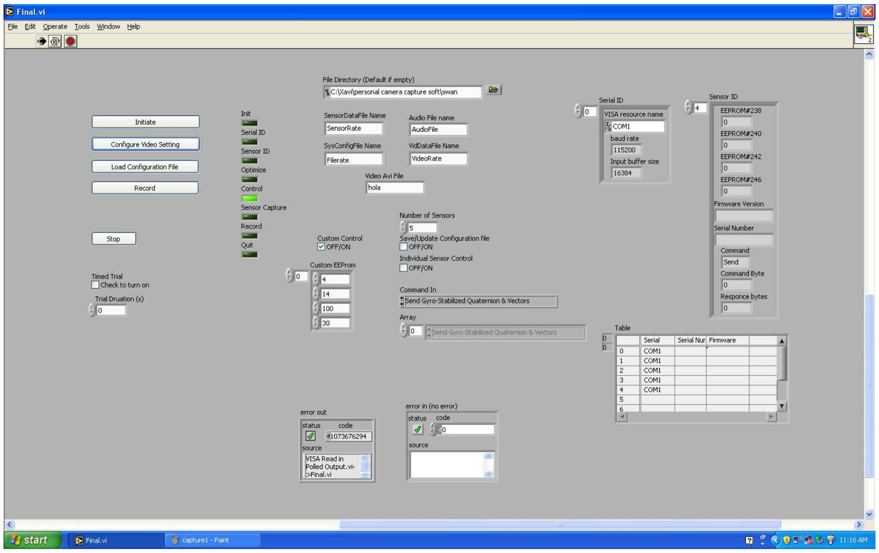


Figure 2: a) Taking eggs from the fridge. b)Scrambling eggs.

sound quality and improve the noise immunity. Each Microphone was connected to an audio Pre-amplifier, which was then connected to a Professional M-Delta Audio PCI card that captured the audio in a regular computer. Audio was recorded and processed under a Unix system with a modified version of the audio editing software Audacity.

To capture audio we placed a total of six microphones around the kitchen. One was placed above the kitchen; the others were attached to cabinets, above the sink and by the refrigerator. All microphones were professional studio condenser microphones made by Behringer (model B-5) (Figure 4.b) and capable of cardiod and omnidirectional polar patterns. The microphone above the kitchen was omnidirectional while the others used the cardiod pattern. The microphones used standard balanced XLR connectors and were connected to an SM Pro Audio PR8 preamp that provided the necessary phantom power. The preamp was then connected to an M-Audio Delta 1010 sound



File Directory (Default if empty): C:\Xavi\personal camera capture soft\swan

SensorDataFile Name: SensorRate

Audio File name: AudioFile

SysConfigFile Name: Filerate

VidDataFile Name: VideoRate

Video Avi File: hola

Number of Sensors: 5

Custom Control: ☑ OFF/ON

Save/Update Configuration file: □ OFF/ON

Individual Sensor Control: □ OFF/ON

Custom EEProm: 0 | 4, 14, 100, 30

Command In: Send Gyro-Stabilized Quaternion & Vectors

Array: 0 | Send Gyro-Stabilized Quaternion & Vectors

Timed Trial: □ Check to turn on

Trial Druation (s): 0

Serial ID: 0

VISA resource name: COM1

baud rate: 115200

Input buffer size: 16384

Sensor ID: 4

EEPROM#238: 0

EEPROM#240: 0

EEPROM#242: 0

EEPROM#246: 0

Firmware Version:

Serial Number:

Command: Send

Command Byte: 0

Responce bytes: 0

error out: status ✓, code 1073676294, source: VISA Read in Polled Output.vi->Final.vi

error in (no error): status ✓, code 0, source:

Table

	Serial	Serial Nur	Firmware	
0	COM1			
1	COM1			
2	COM1			
3	COM1			
4	COM1			
5				
6				

Figure 3: Screenshot of LabView software to gather video and accelerometers/gyroscopes.

card in an ancient Dell computer running Linux. The Delta 1010 has a breakout box to allow for the connections—two 1/4-inch TRS connections per channel for a balanced signal.

Audio was recorded using a modified version of Audacity (audacity.sourceforge.net), an open-source audio editor and recorder. Our modifications enabled logging the recording start and stop events so that we could synchronize the audio with the other recording modalities.

2.3 Accelerometers and gyroscopes

Our third modality is captured with MicroStrain's 3DM-GX1 inertial measurement units. These units contain an accelerometer, gyroscope, and magnetometer. They combine these signals to measure absolute orientation, as well as angular velocity and instantaneous acceleration. All signals are gyro-stabilized and recorded at a regular rate (roughly 60 Hz). The data is captured using LabView software running on the laptop computer carried by the subject. Five sensors are placed on the subjects back, legs, and arms. After synchronous activation of all devices, the program cycles through each of the five sensors to capture a portion of their data stream. Each sensor is set to transmit a continuous data stream. The stream then is broken down into packets and immediately converted to calibrated value. The calibrated values are then combined with the sensor serial and the time stamp to be recorded in a text file. Wires provide battery power to the units and transmit the signal to a serial connector, which is subsequently connected to a serial-to-usb converter that is connected to the usb port of the laptop.

Each sample from the accelerometer includes several data fields. The "Serial num"

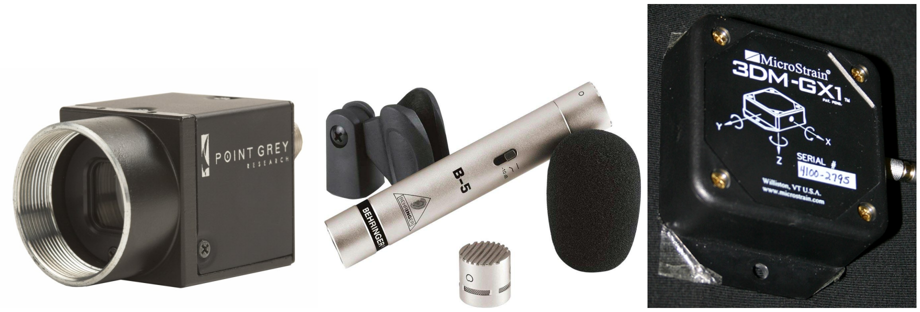


Figure 4: a) Point Grey camera b) Microphones. c) Accelerometers/gyrocopes

field contains the serial number of the accelerometer. The “StabMagField” fields contain the gyro-stabilized output of the magnetometer. The “StabAccel” fields contain the gyro-stabilized output of the accelerometer. The “StabAngRate” contain the gyro-stabilized angular velocity of the unit. The “StabQ” fields contain a gyro-stabilized estimate of absolute orientation given as a quaternion. “Ticks” contains the number of internal clock ticks since the unit was turned on.“TimeStamp” contains an estimated time of the sample from the Windows operating system. Figure 4.c shows an example of the accelerometers/gyroscopes used.

2.4 Motion Capture

Our subjects’ whole body and hand motions were recorded in most of the captures. Current state-of-the-art for whole body capture uses a set of 40-60 markers to approximate the rigid body motion of 15-22 segments. To the extent possible, the markers are placed on joint axes and bony landmarks so that they can more easily be used to find the motion of an idealized skeleton. The hands are often modeled as a single rigid link. We used biomechanical invariants to reduce the number of markers to less than the number required to fully specify the orientation of each rigid body segment. The system used was a Vicon motion capture setup with twelve MX-40 cameras for the captures. The standard motion capture setup was refined to have additional markers as shown in Figure 5. This marker set captured the motion of the shoulders and back more accurately. A whole body motion capture session consists of placing the markers on the subject in the configuration shown in Figure 5. Then, the subject is asked to briefly hold a T-pose (arms straight out to the sides), a motorcycle pose (all joints slightly bent as if riding a motorcycle), and a range of motion where the subject moves each joint through its full range of motion. This information is used to automatically compute a skeleton with limb lengths appropriate for the individual subject in the Vicon software, IQ. Because of the quantity of the data that we intend to capture, the vast majority of the data clean-up will need to be performed automatically. The IQ software provided by Vicon is capable of automatic clean-up of whole body motion when markers are visible in a number of cameras. For more information on the mocap system see $http://mocap.cs.cmu.edu/faqs.php$.

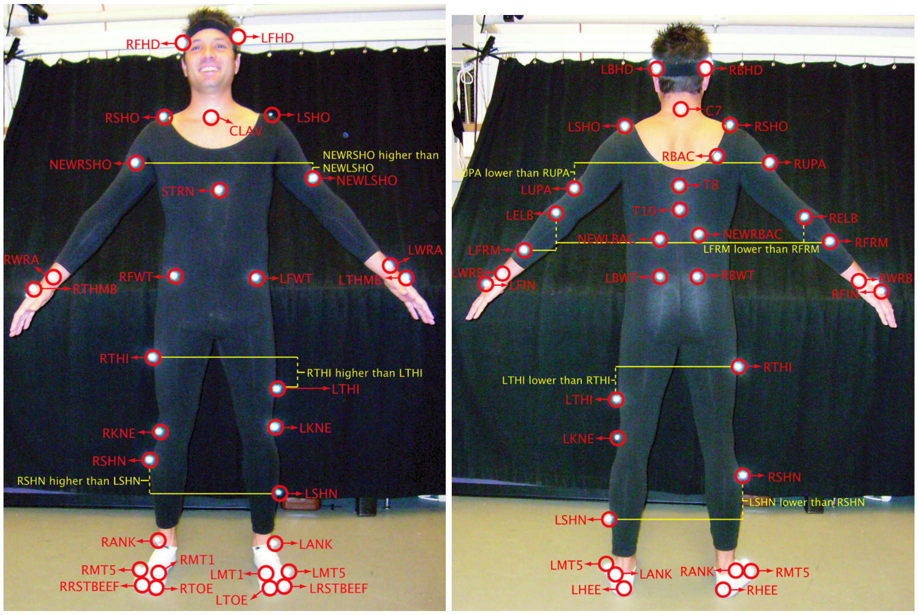


Figure 5: Placement of the markers

3 Data collection procedures

This section describes the calibration procedures for several modalities and how to achieve synchronization across those modalities.

3.1 Camera calibration

Geometric camera calibration is a key aspect for the use of the multi-view video in many potential applications. For geometric calibration, we used the method described by Svodoba et al. [3]. This is a convenient method for calibrating multiple cameras because it is fully automatic, and a freely moving bright spot is the only required calibration object. We gathered calibration data by waving a red LED through the working volume in the dark. The camera projection matrices, one for each camera, are obtained by feeding the calibration data into the Matlab toolbox provided by the authors from http://cmp.felk.cvut.cz/˜svoboda/SelfCal/. To make the algorithm completely automatic a few code changes that remove noise in the image and detect the LED were developed. Additionally, we computed the radial and tangential distortion for each of the cameras.

3.2 Motion capture calibration

There are two steps to calibrating the motion capture system that should be done before capturing begins. For the first step, we use the Vicon 120 mm calibration wand. This wand has three reflective markers on it. We wave the calibration wand around the volume of the capture space. When this is completed, the Vicon software runs an algorithm which figures out where each camera is in relation to the other cameras from the calibration data. The next step is to set the origin of the capture space by placing the Vicon L frame in the space we want to designate as the origin within the capture space.

3.3 Inter-modality synchronization

Synchronization among all the modalities is achieved by combining two different protocols, MultiSync software and Network Time Protocol (NTP). MultiSync software is designed to synchronize the image acquisition of multiple compatible Point Grey cameras across different IEEE-1394b buses on the same computer and across separate buses on multiple computers. Moreover, MultiSinc records the system timestamp in order to be able to synchronize the cameras with other devices. NTP is a protocol for synchronizing the clocks of computer systems over packet-switched, variable-latency data networks. NTP uses UDP port 123 as its transport layer. It is designed particularly to resist the effects of variable latency (jitter). NTP is used to synchronize the time of a computer client or server to another server or reference time source. This protocol is needed in order to synchronize the static cameras, audio, mocap, as well as the wearable computer and sensors (on-board camera and accelerometers/gyroscope). This technique achieves accuracy typically within a millisecond range. NTP is a multiplatform protocol that works with either Windows and Unix/Linux systems, allowing the Audio machine (Linux) to be synchronized with the other machines (Windows). NTP fixes a common time in the system using the LAN connection to refine the drift between machines. After synchronizing the same time reference in all recording machines, the different modalities are captured at the different sample rate typical of each modality. Although the sample rates differ among modalities (30 or 60 samples in the cameras, 100 hertz in the accelerometers or 44100 hertz in the audio), and the sample moment is different in each modality, utilizing the synchronized reference time assures a maximum synchronization drift among modalities of half of the slower sample rate modality. In other words, the maximum drift that will occur among modalities is no more than 1/30*2 (half 30fps frame) or 0.017 seconds.

4 Database organization

We have collected five subjects cooking five recipes (average time 15 minutes/recipe). The file organization of the database is illustrated in Figure 6. There are two main files in the root directory:

- SubjectXX/: Contains the data for subjectXX.
- Util/: Contains software applications to visualize the data (e.g. video codecs, Matlab files to play videos).

Each folder SubjectXX contains subdirectories with the multimodal data:

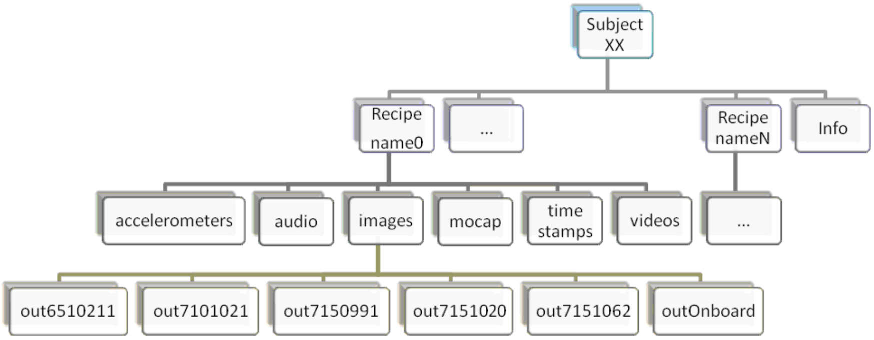


Figure 6: Organization of the database.

- Accelerometers/Gyroscopes:

  All accelerometers/Gyroscope data is contained in the SubjectXX/ RecipeNameX/ accelerometers directory, in a file named SensorRate.txt . See description of section 3.3.

- Audio:

  There are six 44.1KHz, 32-bit float wav files for the static microphones and one mp3 for the watch microphone. We also include the original audacity project files which contain the raw data captured by Audacity. Finally, the file "recording.log" specifies the starting and ending timestamps logged by our modified version of Audacity when the user pushes "record" and "stop." These timestamps can be used to synchronize the audio with the other modalities.

- Image data:

  All image data is contained in the SubjectXX/RecipeNameX/images directory, organized into separate subdirectories for each camera. The aforementioned subdirectories are named with the word 'out' plus the serial number of each camera, except for the onboard camera subdirectory, which is named 'outOnboard'.

- Motion capture

  Motion Capture files are in the SubjectXX/RecipeNameX/mocap directory. There are three files per recipe. C3D files contain the markers position for every frame. The Vfile is a file that describes the applicable motion to a skeleton in a variety of 3D animation software. The Trial file contains the entire capture and has to be opened with the Vicon iQ software.

- Video

  Video files are in the SubjectXX/RecipeNameX/video directory. There is one avi file for each camera named with the same title as the folder that contains the images forming the frames of that video. Moreover, there is a video with the five main cameras playing at the same time, and a six view video playing these five cameras plus the onboard one.

All time stamp information is contained in the SubjectXX/ RecipeNameX/ timestamps directory. Files STime6510211.txt, STime7101021.txt, STime7150991.txt, STime7151020.txt, STime7151062.txt describe the relationship between the name of each image and its real time stamp. Otherwise, STimeOnboard.txt only contains the sequence of time stamps since file name of the images contains the time stamp of each image.

- Info.xls: Contains a list of all subject files and their size.

Acknowledgements

This work was partially supported by the National Science Foundation under Grant No. EEEC-0540865. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Thanks to Minh Hoai and Joan Perez for helping with the camera calibration software. Thanks to Tomas Maldonado for providing the LabView software to record wearable sensors and camera. Thanks to Joan Perez, Ricardo Cervera and Francisco Perez for volunteering to be captured and making such good brownies!

References

[1] Carnegie Mellon Motion Capture Database. http://mocap.cs.cmu.edu.

[2] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *International Conference on Pattern Recognition (ICPR)*, 2004.

[3] T. Svoboda, D. Martinec, and T. Pajdla. A convenient multi-camera self-calibration for virtual environments. *PRESENCE: Teleoperators and Virtual Environments*, 14(4):407–422, 2005.