MUSICAL GESTURES
TOOLBOX MANUAL

Bo Zhou

Department of Informatics
University of Oslo
2.2016

Contents

1. BASICS

Matlab Environment

Musical Gestures Toolbox(MGT) requires Matlab 2015a environment or more recent. Matlab software is available on the website(www.mathworks.com). To install MGT is quite simple.Just add MGT in the Matlab path:

1. Open the Matlab **Set Path** window:

- In Matlab 2015a or more recent,on the **Home** tab, in the **Environment** section,click **Set Path**.

- In the new **Set Path** window,click on the second button **Add with Subfolders**.

- In the file browser,select the folder you downloaded and unzipped and click **Open**.Then click **Save** button. The addresses of the folder are added.

Introduction

Musical Gestures Toolbox is a Matlab toolbox which contains functions for the analysis and visualization of video, mocap, and audio data. Musical Gestures Toolbox integrates Mocap tool box and Mirtoolbox which aim for the analysis the motion capture data and audio or music input data respectively. So before using Musical Gestures Toolbox, make sure Mocap tool box and MIRtoolbox are correctly installed. Mocap tool box is available on the website(www.jyu.fi/music/coe/materials/mocaptoolbox). The URL of MIRtoolbox website is (www.jyu.fi/music/coe/materials/mirtoolbox). The steps of installation are the same as MGT above. To use all the functions in the MGT, Signal Processing Toolbox and Image Processing Toolbox are needed as well. The Musical Gestures Toolbox comes with no warranty. It is free software.

2. GENERAL

Functions

The Musical Gestures Toolbox contains a set of functions for the analysis and visualization of video, audio, and mocap data. There are four categories of functions:

- Data input and edit functions

- Data preprocessing functions

- Visualization functions

- Middle and higher level feature extraction functions

Data Structures

The Musical Gestures Toolbox uses only one struct data structure. This struct contains three fields,video,audio,mocap, which corresponds three types input data. A Musical Gestures data structure is created by the function **mginitstruct**. For instance:

mg=mginitstruct;

Then a struct mg is created with three fields. Here video field contains data and general parameters of video. Since this is a core part of Musical Gestures Toolbox, it will be introduced in detail. As for other two fields, audio and mocap,please refer to reference documentation respectively. Normally, the field video is a struct as well containing several fields. Data field stores each frame of the video. Gram field stores the information of horizontal and vertical motion gram. Motion field contains the motion information between two successive frames. Qom field contains quantity of motion of each frame. Com field contains centroid of motion of each frame. Nframe field contains the number of the frames of the video. The framerate of video is stored in the field framerate. Usually, it is 30 frames per second. Duration field is for length of the video. Musical Gestures Toolbox uses two general methods to estimate the motion, one is motion gram, another is optical flow. The method field is used for the option of these two methods. Other two fields filename and path are for the name of video and path.

3. FUNCTION REFERENCE

mginitstruct

synopsis

Initializes Musical Gestures data structure.

syntax

mg = mginitstruct;

output

mg:musical gestures data structure with three fields, audio, video, and mocap.

examples

mg = mginitstruct;

comments

also see

mgread

synopsis

Reads any one of three types of data file,mocap,video,audio data file and returns a musical gestures data structure.

syntax

mg = mgread('Video');
mg = mgread('Audio');
mg = mgread('Mocap');
mg = mgread('Folder');

input parameters

Video, Audio, and Mocap indicate three types of input data. With any one of them,a file open dialog opens and provides user to choose corresponding input file. With option folder, mgread reads any combinations of three types of data. For instance, it could only

read Mocap and Video,etc.

output

mg:musical gestures data structure with corresponding field filled by parameters and data.

examples

comments

Currently mgread provides .avi,.mp4,.m4v,.mpg,.mov file formats.

also see

mgvideoreader

mgvideoreader

synopsis

Reads specified video file with or without extracting video and returns a data structure containing the video parameters and data. If input is a musical gestures data structure containing the video data, mgvideoreader will extract the section according to the extraction parameters.

syntax

mg = mgvideoreader(filename);

mg = mgvideoreader(mg);

mg = mgvideoreader(filename,'Extract',starttime);

mg = mgvideoreader(filename,'Extract',starttime,endtime);

mg = mgvideoreader(mg,'Extract',starttime,endtime);

input parameters

filename:specifies the name of the video file

mg:musical gestures data structure containing the video data

Extract:indicates that mgvideoreader reads a certain segment from the video

starttime:start of extracted segment

endtime:end of extracted segment

output

mg:a struct data structure containing the video frames from starttime to endtime and other video parameters

examples

mg = mgvideoreader(filename);
mg = mgvideoreader(filename,'Extract',10,15) mg = mgvideoreader(mg,'Extract',10,15);

comments

also see

mgreadsegment

mgreadsegment

synopsis

Extracts a temporal segment from the video, audio, mocap data file.

syntax

mg = mgreadsegment(filename);
mg = mgreadsegment(filename,starttime,endtime);

input parameters

filename:can be any type of three different input data files
starttime:start of extracted segment
endttime:end of extracted segment

output

mg:a struct containing video, or audio, or mocap data from starttime to endtime

examples

mg = mgreadsegment('filename.mp4');
mg = mgreadsegment('filename.wav');
mg = mgreadsegment('filename.tsv');
mg = mgreadsegment('filename.mp4',10,15);
mg = mgreadsegment('filename.wav',10,15);
mg = mgreadsegment(mg,10,15);

mg = mgreadsegment('filename.tsv',10,15);

comments

Filename.tsv is a type of the mocap data file. Other types of mocap data file can be filename.c3d,filename.mat,filename.wii.

also see

mgread,mgvideoreader,mctrim,miraudio

mgvideocrop

synopsis

Crops video recording according to the position given by user. If the position is not given in the function, it will show first frame of video, user can draw any shape, any region of interest, then function will create the mask of this area in every frame. Finally it writes back the croped video into disk and returns a musical gestures of croped video.

syntax

mg = mgvideocrop(mg,pos);

mg = mgvideocrop(mg);

mg = mgvideocrop(filename,pos);

mg = mgvideocrop(filename);

input parameters

mg:musical gestures data structure contains video parameters

pos:2 by N position matrix,first row indicates the index of column of the vertex;second row indicates the index of row of the vertex,N is the number of vertexes

filename:if the input is a video, the name of video should be given

output

mg:a musical gestures data structure contains the croped video parameters

examples

mg = mgvideocrop(mg,[20 30 30 20;30 30 40 40]);

mg = mgvideocrop(filename,[20 30 30 20;30 30 40 40]);

mg = mgvideocrop(filename);

mg = mgvideocrop(mg);

comments

also see

mgvideorotate

synopsis

Rotates the video by angle degrees in a counterclockwise around its center point. To rotate the video clockwise, specify a negative value for angle. It will write the rotated video in the disk and return a musical gestures data structure contains the rotated video.

syntax

mg = mgvideorotate(file,angle);

mg = mgvideorotate(file,angle,method);

mg = mgvideorotate(mg,angle);

mg = mgvideorotate(mg,angle,method);

input parameters

file:the video file which needs to be rotated

mg:musical gestures data structure contains a video information

angle:specify the angle of rotation, positive value rotates in counterclockwise;Negative value rotates in clockwise

method:gives the interpolation method

output

mg:a musical gestures data structure contains rotated video

examples

mg = mgvideorotate(file,5,'bilinear','crop');

mg = mgvideorotate(file,5,'bilinear');

mg = mgvideorotate(mg,5,'bilinear');

mg = mgcrop(mg,-5,'bilinear','crop');

comments

also see

imrotate

mgvideocat

synopsis

Concatenates two videos into one,finally writes the concatenated video in the disk and returns a musical gestures data structure which contains concatenated video.

syntax

mg = mgvideocat(file1,file2);

mg = mgvideocat(mg1,mg2);

input parameters

file1,file2:the name of video file

mg1,mg2:musical gestures data structures contain the videos

output

mg:a musical gestures data structure contains the concatenated video

examples

mg = mgvideocat(file1,file2);

mg = mgvideocat(mg1,mg2);

comments

also see

mgvideoadjust

synopsis

Adjust the contrast of the video,for RGB video, mapping value of three channels must be given. Finally, it writes the adjusted video in the disk and returns a musical gestures data structure which contains the adjusted video.

syntax mg = mgvideoadjust(mg);

mg = mgvideoadjust(file);

mg = mgvideoadjust(file,mapin,mapout);

input parameters

file:the name of video file

mg:a musical gestures data structure contains the video

mapin:a matrix contains range of mapping in, for instance, $[low\_in; high\_in]$ for graylevel video;$[low\_in1, low\_in2, low\_in3; high\_in1, high\_in2, high\_in3]$ for RGB video

mapout:a matrix contains range of mapping out, for instance, $[low\_out; high\_out]$ for graylevel video;$[low\_out1, low\_out2, low\_out3; high\_out1, high\_out2, high\_out3]$ for RGB video

output mg:musical gestures data structure contains the adjusted video

examples

mg = mgvideoadjust(file,[0.1;0.8],[0.2;0.9]);

mg = mgvideoadjust(file,[0.1 0.2 0.05;0.8 0.9 1],[0.3 0.4 0.2;0.9 0.8 0.9]);

mg = mgvideoadjust(mg,[0.1;0.8],[0.2;0.9]);

mg = mgvideoadjust(mg,[0.1 0.2 0.05;0.8 0.9 1],[0.3 0.4 0.2;0.9 0.8 0.9]);

comments

also see

imadjust

mgvideomagnify

synopsis

Magnify the subtle variations of a video which are difficult or impossible to see by naked human eyes, providing two types of temporal spatial filters:IIR,Butter. This function requires matlabPyrtools toolbox.

syntax

mg = mgvideomagnify(videofile,'IIR',f1,f2,alpha,lambda_c ,attenuation);

mg = mgvideomagnify(videofile,'Butter',f1,f2,alpha,lambda_c ,samplingrate,attenuation)

input parameters

videofile: input video file

med: fitler methed with option 'IIR','Butter'

f1,f2: coefficients of second-order filter

alpha: magnification factor

lambda_c : spatial cutoff frequency

samplingrate: spatial resolution

attenuation: attenuation factor

output mg: musical gestures data structure containing magnified video

examples

mg = mgvideomagnify('video.mp4','IIR',0.4,0.05,10,16,0.1);

mg = mgvideomagnify('video.mp4','Butter',3.6,6.2,60,90,30,0.3);

comments

also see

mgvideofilter

synopsis

Filter a video with 'Temporal', 'Spatial', 'Both' options. With 'Temporal' option, perform temporal filter; with 'Spatial' option, perform spatial filter by median, or average filter.

syntax mg = mgvideofilter(videofile,type,med,h);

input parameters

videofile: input video file

type: filter type

med: filter method

h: size

output mg: musical gestures data structure containing filtered video

examples

mg = mgvideofilter(videofile,'Spatial','Mediean',[3,3]);

mg = mgvideofilter(videofile,'Spatial','Average',[3,3]);

comments

also see

mgmotionfilter

synopsis

Filter the motion image or optical flow filed with 'Regular', 'Binary', 'Blob' option.

syntax

mg = mgmotionfilter(f,med,thres);

input parameters

f: input motion image or optical flow field

med: 'Regular', turns all values below thres to 0; 'Binary' truns all values below thres to 0, above thres to 1; 'Blob' removes individual pixels with erosion method.

thres: for 'Regular', 'Binary' option, thres is a value of threshold [0,1]; for 'Blob', thres is element structure created by strel.

output

mg: musical gestures data structure containing the filtered video.

examples

mg = mgmotionfilter(f,'Regular',0.2); mg = mgmotionfilter(f,'Blob',[1 1 1;1 1 1;1 1 1]);

comments

also see

mgmap

synopsis

Maps the same section from audio and mocap data file according to the parameters of video data file.

syntax

mg = mgmap(mg,'Audio');

mg = mgmap(mg,'Audio',filename);

mg = mgmap(mg,'Mocap',filename);

mg = mgmap(mg,'Both',filename);

mg = mgmap(mg,'Both',filename,filename);

input parameters

mg:musical gestures data structure containing parameters of video file. Such parameters contain the specified duration, starttime, endtime, etc

type:Audio,Mocap indicate the type of mapping data

filename:if map the mocap data file, then the filename of mocap file must be given;the filename of audio is not necessary to be given. The first filename is audio name, the second is mocap name

output

mg:returns a musical gestures data structure containing the same segment from video, audio, mocap

examples

mg = mgmap(mg,'Audio');

mg = mgmap(mg,'Audio',audiofilename);

mg = mgmap(mg,'Mocap',mocapfilename);

mg = mgmap(mg,'Both',audiofilename,mocapfilename);

comments

also see

mgmotion

synopsis

Computes the motion, motion gram, quantity of motion, centroid of motion according to the method given by user.

syntax

mg = mgmotion(mg,'Diff');

mg = mgmotion(mg,'OpticalFlow');

mg = mgmotion(filename,'Diff',starttime,endtime);

mg = mgmotion(filename,'OpticalFlow',starttime,endtime);

input parameters

mg:musical gestures data structure containing the parameters and data of video

method:'Diff', 'OpticalFLow' specify the method used to compute the motion and features. 'Diff' method takes the absolute difference value between two successive frames. 'OpticalFlow' method uses optical field to estimate the motion.

filename:the video file should be given,when the musical gestures data structure is given

starttime:compute from the specified start time of video file

endtime:compute motion and features until the specified end time of video

output

mg:return a gestures data structure containing the parameters and data of video, the motion gram,quantity of motion and centroid of motion will be stored in the corresponding fields

examples

mg = mgmotion(mg,'Diff');

mg = mgmotion(filename,'Diff',5,10);

comments

When specify OpticalFlow method,the motion gram field in the data structure is empty.

also see

mgcentroid

synopsis

Computes the centroid of the motion image. If the input image is rgb image, mgcentroid will first change it to a gray image.

syntax

[com,qom] = mgcentroid(im);

input parameters

    im:an image or motion image

output

    com:centroid of the given image

    qom:quantity of motion of the given image

examples

    [com,qom] = mgcentroid(im);

comments

also see

mgautocor

synopsis

    Estimates the period of movement for the quantity of motion.

syntax

    [per,ac,eac,lag] = mgautocor(mg);

    [per,ac,eac,lag] = mgautocor(mg,maxperiod,method);

input parameters

    mg:musical gestures data structure containing the parameters and data of the video

    maxperiod:maximum period is considered

    method:sets if 'first' or 'highest' maximal value of the autocorrelation function is taken as periodicity estimation

output

    per:a scalar containing the period estimation of the quantity of motion

    ac:a vector containing autocorrelation functions of quantity of motion

    eac:a vector containing enhanced autocorrelation functions of qantity of motion

    lag:a vector containing lag values for the autocorrelation functions

examples

[per,ac,eac,lag] = mgautocor(mg);

[per,ac,eac,lag] = mgautocor(mg,2,'first');

comments

also see

mcperiod,mcwindow

mgopticalflow

synopsis

Estimates the motion using Horn-Schunck method.

syntax

flow = mgopticalflow(fr2,fr1);

input parameters

fr2:the second video frame

fr1:the first video frame

output

flow:optical flow field containing the magnitude and orientation of velocity

examples

flow = mgopticalflow(fr2,fr1);

comments

also see

ComputeFlow,Gradient,opticalFlow

ComputeFlow

synopsis

Compute the horizontal and vertical component of the optical flow field using the Horn-Schunck method.

syntax

    [U,V] = ComputeFlow(dx,dy,dt);

input parameters

    dx:the horizontal gradient

    dy:the vertical gradient

    dt:the temporal gradient

output

    U:the horizontal component of the optical flow field

    V:the vertical component of the optical flow field

examples

comments

also see

    Gradient,ComputeFlow

Gradient

synopsis

    Compute the dx,dy,dt graident of two video frames.

syntax

    [dx,dy,dt] = Gradient(imNew,imPrev);

output

    imNew:the second video frame

    imPrev:the first video frame

input parameters

    dx:the horizontal gradient

    dy:the vertical gradient

    dt:the temporal gradient

examples

comments

also see
    ComputeFlow

mgvideoplot

synopsis
    Plots the motion image,motion grams,mocap grams over time.

syntax
    mgvideoplot(mg);

input parameters
    mg:musical gestures data structure containing the paramemters and data of the video,mocap data file

output

examples

comments
    The method field in the data structure should be 'Diff'.

also see

mastatistics

synopsis
    Calculates the various features of the motion.

syntax
    features = mgstatistics(mg,'Video','FirstOrder');
    features = mgstatistics(mg,'Video','SecondOrder');

input parameters

mg:musical gestures data structure containing the motion data of the video

type:'Video' indicates that compute the features of the video data

type:'FirstOrder' or 'SecondOrder' chooses to compute lower or higher level features

output

features:a matrix containing features. When chooses 'FirstOrder',return a N by 4 dimensions matrix. When chooses 'SecondOrder', return a N by 8 dimensons matrix. N represents the number of the samples.

examples

features = mgstatistics(mg,'Video','SecondOrder');

features = mgstatistics(mg,'Video','FirstOrder');

comments

After computing the features, it is necessary to normalize the features such that all features are scaled to range (0,1).

also see

mawaveplot

synopsis

Plots the waveform,root mean square,spectrum of the audio.

syntax

mgwaveplot(mg);

input parameters

mg:musical data structure

output

examples

comments

The input mg should contain the audio part, which can be computed from mgcrop, mgtrim, mgread.

also see

mirrms,mirspectrum

mgpca

synopsis

Computes the principle components and projects the original features space into selected principle components.

syntax

[x,p] = mgpca(x,ind);

input parameters

x:input feature space

ind:selection componaents

output

out:projected features space

p:p.r,eigenvalue, p.v,eigenvectors, p.ratio, ratios of eigenvalues

examples

[out,p] = mgpca(x,1:3);

comments

When the selection components variable is not given, mgpca chooses the principle components whose variances are greater than 0.9.

also see

mgvideosample

synopsis

Downsamples the video frame in the space for the sake of memory.

syntax

mg = mgvideosample(mg,'Sampling',factor);

input parameters

mg:musical gestures data structure containing the video data at least

'Sampling':indicates video sampling

factor:sampling factor of image in space

output

mg:sampled musical gestures data structure, resulting in reduced size of all frames fo
the video

examples

mg = mgvideosample(mg,'Sampling',2);

comments

also see

mgsimilarity

synopsis

Computes the similarity of the input array or vector based on the pairwise of observations,and returns a similarity matrix.The default method is euclidean method.

syntax

sim = mgsimilarity(x,distance);

sim = mgsimilarity(x);

input parameters

x:a vector or a array contains the data input

distance:the type of method, the default is euclidean

output

sim:a matrix contains the measurement of similarity for each pair of observations

examples

sim = mgsimilarity(x,'euclidean');

sim = mgsimilarity(x,'cityblock');

comments

also see
pdist

mgsave

synopsis

Saves the musical gestures data in the current workspace into a folder. Meanwhile, mgsave creates a .txt file in this folder.

syntax
mgsave(fn);

input parameters
fn:musical gestures dataset,variables

output
A folder contains the dataset or variables.

examples
mgsave(mg);

comments

also see
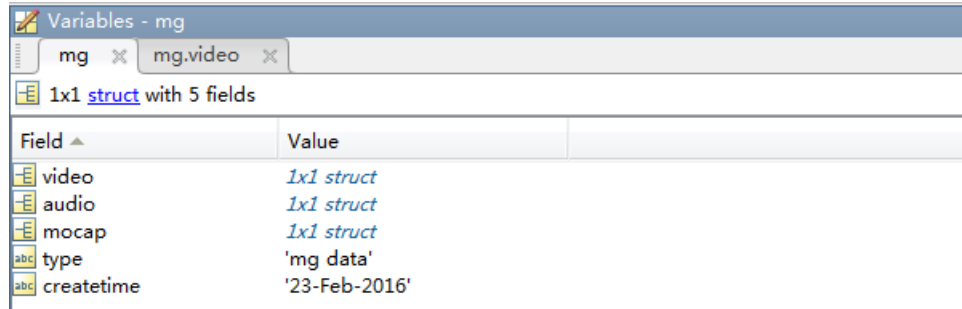
4. EXAMPLES

mgdemo1

This example shows how you can read video, audio, mocap files into Matlab and how you can preprocess the data. The data is stored as musical gestures data structure after importing. The Musical Gestures Toolbox includes mgdemo1 that contains video, audio, mocap recordings. It is used for the example of this manual. Firstly, read video data into matlab workspace, mgread will open a file open dialog and select the videl file Video.mp4, audio1.wav, a.tsv.

>   **mg = mgread('Folder')**

A musical gestures data structure is imported into Matlab workspace. It shows three substructs, video, audio, mocap, respectively. The video recordings are the pianst who is playing the piano. The length of the video is around 97.5seconds. Let's look futher at the
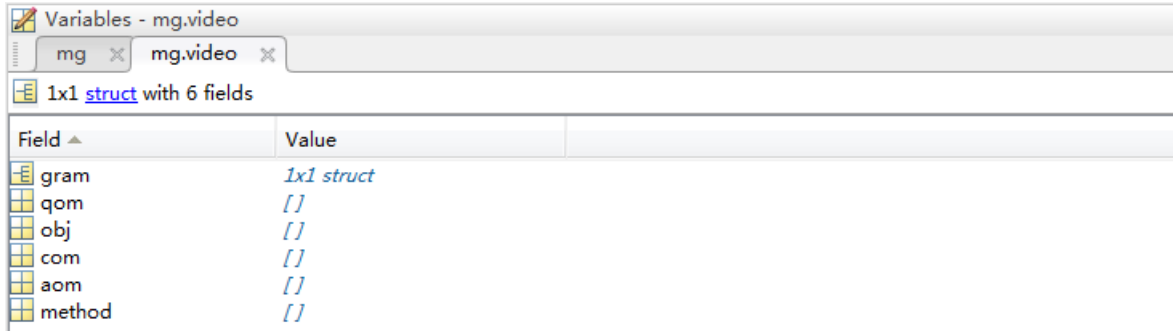
Figure 1: musical gestures structure

video substruct: Next, we can extract the segment from the structure. For instance:

>   **mgseg = mgvideoreader(mg,'Extract',10,20);**

This operation will extract 20seconds long from 10seconds to 30seconds of the video. In order to keep the same segment with video, it needs to do the same extraction on the audio and mocap data. This can be done by mgmap.

>   **mg.video = mgseg.video;**

Figure 2: video substruct

mgseg = mgmap(mg,'Both');

Alternatively, all operations above can be simply done by one step. The function mgreadsegment does it.

mgseg = mgreadsegment(mg,10,30);

Sometimes, you may think the video recordings are too large. For instance, the video frame is with 480 by 680 dimension. Usually, this is large enough to satisfy your needs. Reducing to half may be accepted as well. mgvideosample could help you conveniently.

mg_v_sam = mgvideosample(mgseg,'Sampling',2);

Now, each video frame has been reduced into half. If we want to look only the region of interest of the video locally, here mgvideocrop will crop specified region on each frame. User can define any shape of region first or draw the region on the video frame. The cropped video can be used to compute the motion as well. In a sense, the result should have good estimation off local object, for instance, hands, head, etc. Next, the motion grams, quantity of motion and centroid of motion can be computed.

mgcrop = mgvideocrop(mg);
mgcrop = mgvideocrop(mg,pos);

mg_motion = mgmotion(mg_v_sam,'Diff');
mg_motion = mgmotion(mgseg,'Diff');

26
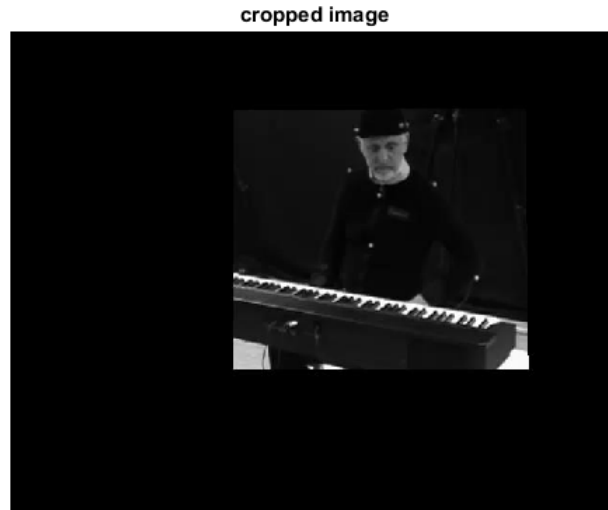
Figure 3: cropped video frame

The method 'Diff' uses the absolute difference between two successive frames. It is interesting to note the player does almost the same action repeatedly from the quantity of motion. From the view of the centroid of motion, most of the blue points cluster in the right side. This is because the player plays the piano using his left hand more often. So the center of movement appears in the right side.

Next,the periodicity of movement might be investigated from the quantity of motion. Let's have a look at it.

[per,ac,eac,lag] = mgautocor(mgseg);

The period is 0.198 seconds. The first maximum at nonzero lag is 0.198 seconds, this is the same as the privious value. Sometimes, it is necessary to show the motion grams and mocap gram. The function mgvideoplot plots the horizontal and vertical motion gram combined with mocap gram over time.

mgvideoplot(mg_motion);
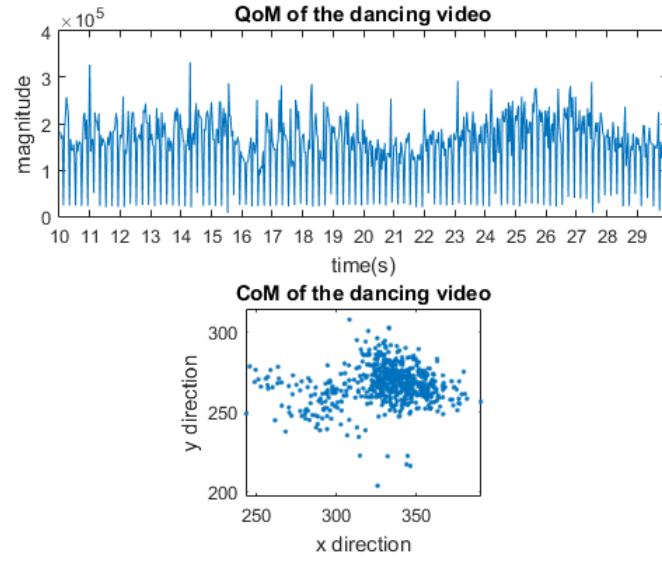
Figure 4: quantity of motion and centroid of motion from 10s to 30s
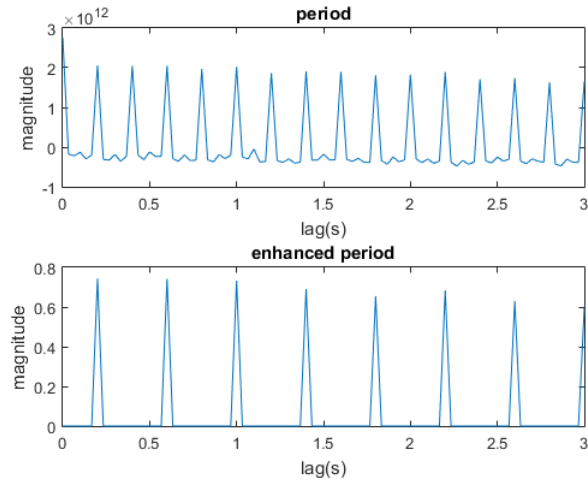
Figure 5: the period estimation of movement

To look at motions more clearly, the vertical motion gram is transposed and the motion grams are ploted by imagesc function after some noise removal. It can be showed that
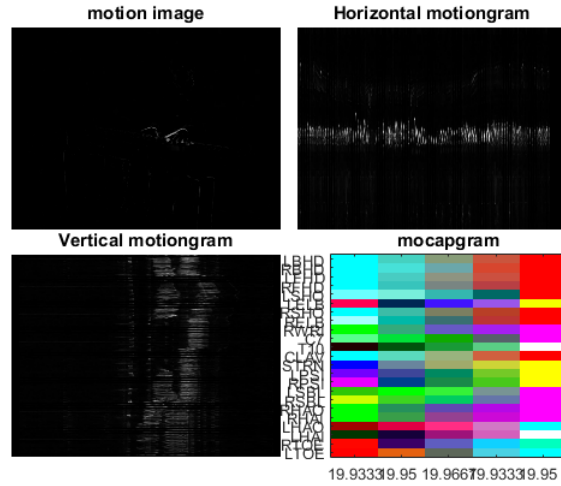
Figure 6: motion gram and mocap gram over time
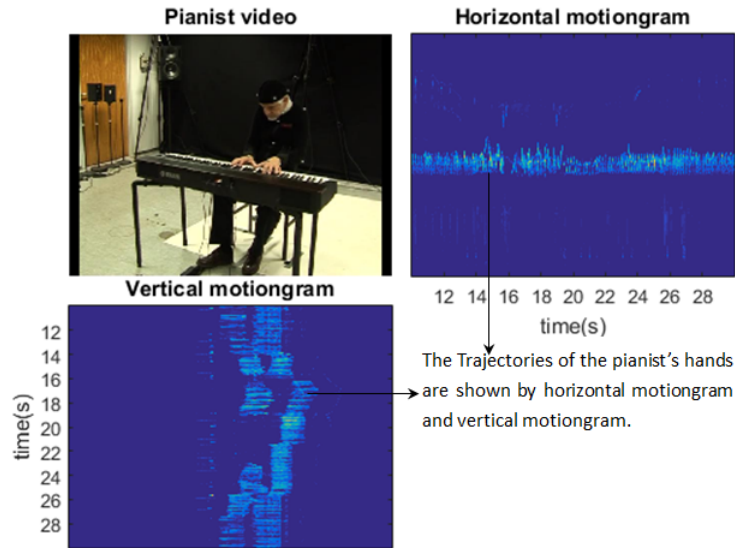
Figure 7: horizontal and vertical motion gram,the vertical motion gram is transposed

the player repeats the similar action from the horizontal motion gram, which corresponds to the quantity of motion. From the vertical motion gram, it is very clear to note there

29

is only hand playing the piano from 9 seconds to 13 seconds. This explians why most centroid of motion locate in the right side.

Next,let's continue to investigate the audio recording. mgwaveplot plots the waveform of audio, root mean square,and their spectrum.

mgwaveplot(mg_motion);
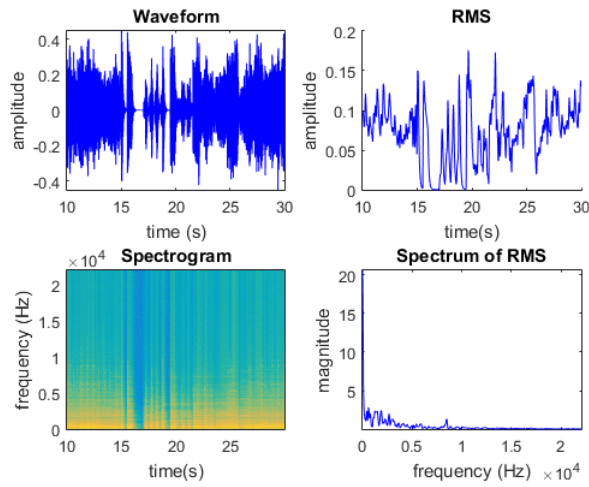
Figure 8: waveform,rms and their spectrum

We may hope that the spectrum of the quantity of motion also gives us some useful information. Here we can compare it with the spectrum of the rms.
Look at the spectrum of qom, the foundamental frequency is given by 5Hz, which is 0.2 seconds. This value is the same as period estimated by mgautocor function.

If we want to look at some features of motion image, function mgstatistics can generate first order and second order features.

features1 = mgstatistics(mg_motion,'Video','FirstOrder');
features2 = mgstatistics(mg_motion,'Video','SecondOrder');
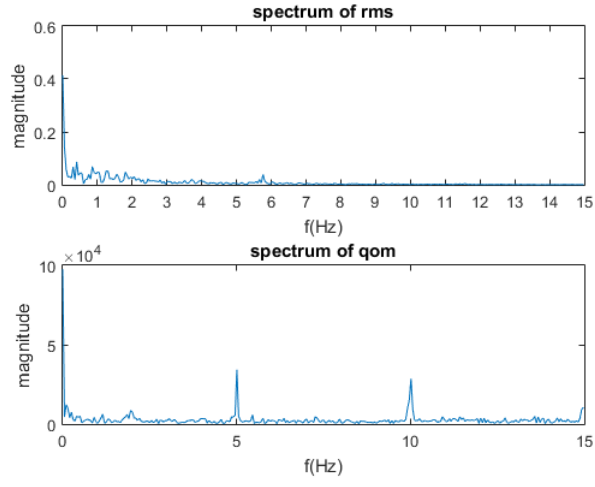
Figure 9: spectrum of qom and rms
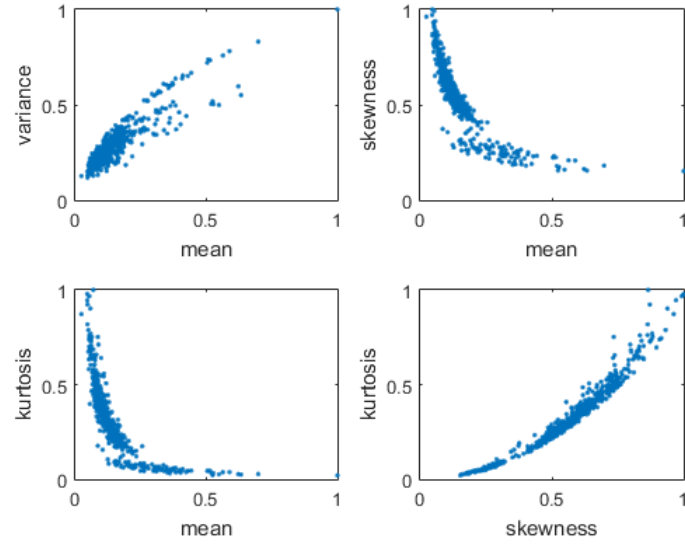
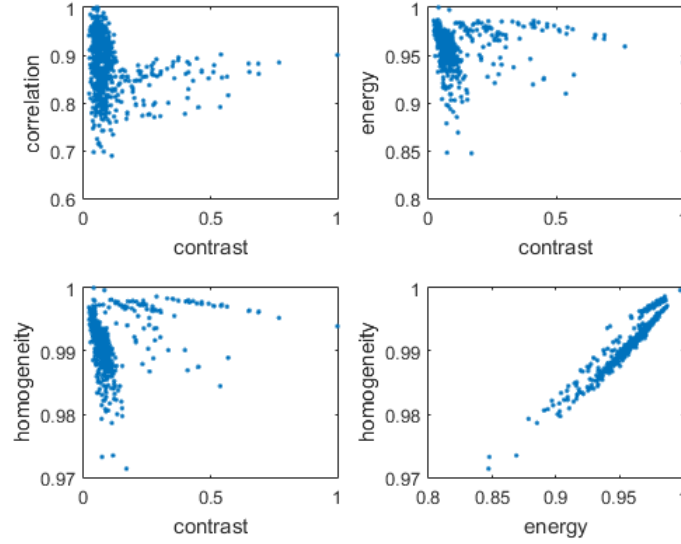Figure 10: the first order feature space

31

Figure 11: the second order feature space

mgdemo2

This example shows how you can read video, audio, mocap files into Matlab and how you can preprocess the data. The data is stored as musical gestures data structure after importing. The Musical Gestures Toolbox includes mgdemo2 that contains video, audio, mocap recordings. It is used for the example of this manual.

```
mg = mgread('Folder');
```

A musical gestures data structure is imported into Matlab workspace. It shows three substructs, video, audio, mocap, respectively. The video recordings are the nine people dancing to music. The length of the video is around 10 minutes. In this demo, we will find that preprocessing the video recording is very important if the video recordings are quite large. Because the frame size is 1080 by 1920, this is very high resolution video recording. In partice, 320 by 480 is enough to compute the motion gram and extract the features. In order to speed up the the calculation, resample the video is necessary.

```
mgsample = mgvideosample(mg,4);
```

The function mgvideosample resamples the video recording. After sampling, the size of each frame is reduced into 270 by 480. Here because of the permission issue, the figures is not included here, we will only show the motion image and motion gram results.

For the 10 minutes long video recording, extracting a certain temporal segment is also necessary.

mgsampleseg = mgvideoreader(mgsample,'Extract',5,125);

It extracts two minutes from 5seconds to 125seconds of the video recording. After this, it should map the same temporal segment to audio and mocap data recording.

mgsampleseg = mgmap(mgsampleseg,'Both','dance.wav','dance.c3d');

Now, the musical gestures data structure mgsampleseg contains the same temporal segment of video, audio, mocap data. Next, you can do analysis based on this data structure.

mg_motion = mgmotion(mgsampleseg,'Diff');

The quantity of motion and centroid of motion are showed in following figures.

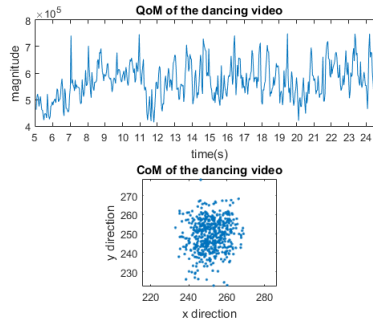Figure 12: Qom and Com of dancing video

The periodicity is estimated by mgautocor based on quantity of motion.

[per,ac,eac,lag] = mgautocor(mg_motion);

In the enhanced period, the first nonzero lag is 1.5seconds. However, in the upper figure, the first nonzero lag is 0.41 seconds. It will be interesting to investigate the tempo
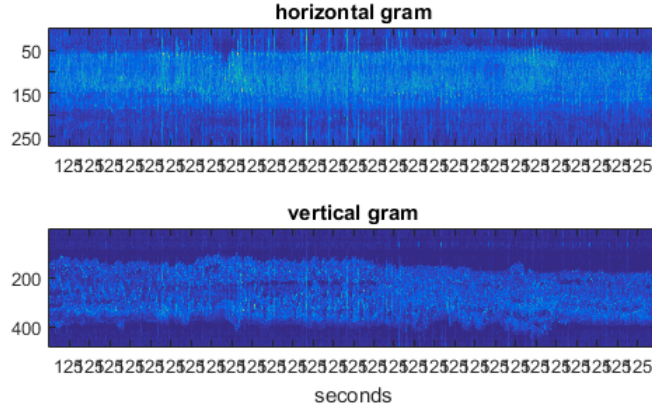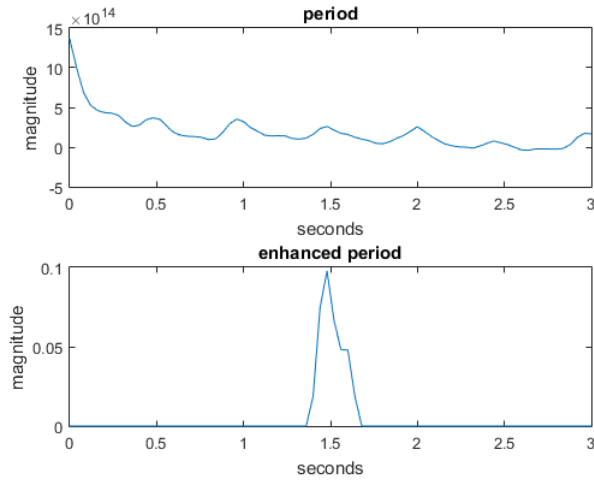
Figure 13: Motion grams of dancing video

Figure 14: Periodicity estimation of dancing

of the sound and spectrum of the quantity of motion. Here the tempo of the sound can be estimated by funtion mirtemp in MIR Toolbox. It is 122bpm, namely around 2 beats per second. Now, let us have a look at the spectrum of the quantity of motion. Figure 15 shows a peak located in around 2.4Hz, this gives us an interesting clue. It seems that people try to trace the tempo of the music. Next, we try to analysis the movements according to mocap data set. The similarity between 9 persons is calculated by function
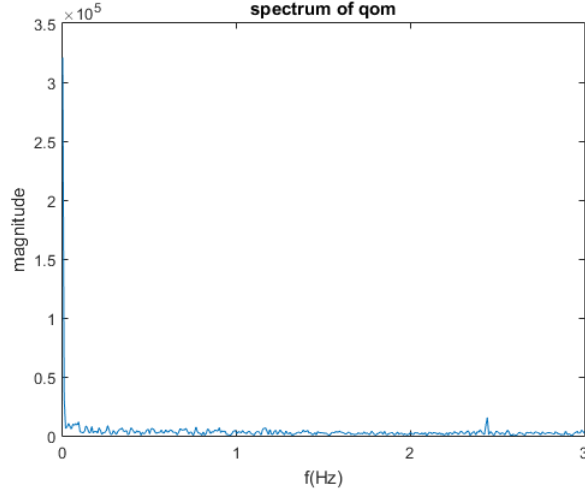
Figure 15: The spectrum of Qom

mgsimilarity in MGT box. However, before computing, the mocap data should be preprocessed. Firstly, take the absolute difference between two successive frames as the way of computing motion image. Secondly, norm the dataset, this is simply done by function mcnorm in Mocap tool box.

sm = mgsimilarity(mg_motion.mocap.data);

To analysis movements of every person, the peroidicity of absolute difference movement of each person can be computed by the same way. Figure 18 shows the each person's periodicity of absolute difference movement. From this view, person 2 move to music very regularly.
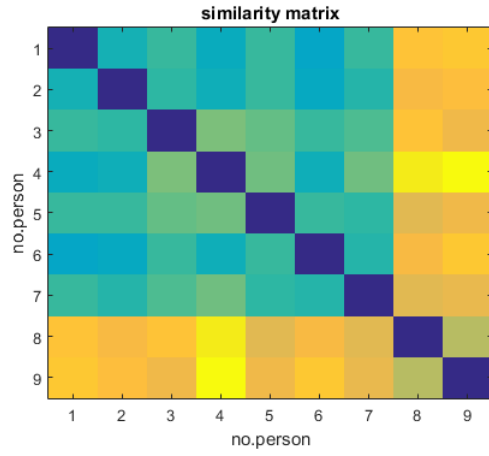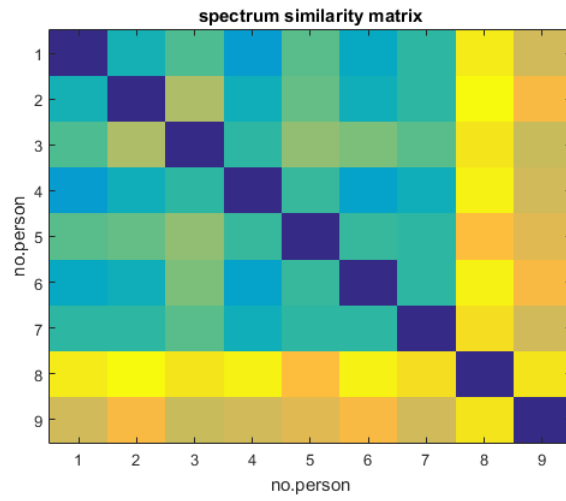
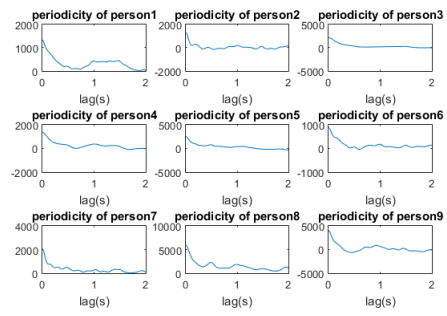Figure 16: Similarity matrix of 9 persons

Figure 17: Similarity matrix of spectrum of 9 persons

Figure 18: Periodicity of each person