University of Twente
Enschede - The Netherlands

Universitat
Autònoma
de Barcelona

etse

Dance Movement Patterns Recognition (Part I)

Computer Science Final Project Report
made by

Israel Solà Cerdán

and directed by

Anton Nijholt and

Dennis Reidsma

Jordi Carrabina

Enschede, 09 of February of 2007

Universitat
Autònoma
de Barcelona

etse
Escola Tècnica Superior d'Enginyeria

El sotasignat, Jordi Carrabina

Professor de l'Escola Tècnica Superior d'Enginyeria de la UAB,

CERTIFICA:

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en Israel Solà Cerdán

I per tal que consti firma la present.

Signat: ....

Bellaterra, 09 de Frebrer de 2007

University of Twente
Enschede - The Netherlands

HMI
Human Media Interaction

Who signs, Anton Nijholt

Electrical Engineering, Mathematics and Computer Science Department at the University of Twente teacher,

CERTIFIES:

That the work, explained in this report, has been done under his management by Israel Solà Cerdán

And to certify it, he signs this document.

Signed: ....

Enschede, 09 of February of 2007

Acknowledgments

After many hours of dedication and big efforts made, has been possible to carry out this final project.

This project would not has been possible without the collaboration of our project directors Anton Nijholt, Dennis Reidsma and Jordi Carrabina which have not doubt to help us in everything that we have needed.

Thanks for the received aid of Herwin van Welbergen and Ronald Poppe that introduced us in The Virtual Dancer wonderful world.

Thanks to my family to support the fact I have come to Netherlands and they only have been able to see me by video call (thanks to the video call technology).

And thanks to my partner Jesús Sánchez to have made this project hand by hand, day by day.

Contents

Chapter 1. Introduction

1.1 Presentation

The report that you have in your hands is the first part of a joint project, divided in two parts. To understand the global idea of the whole project you also have to read the second part of the project done by Jesús Sánchez (Dance Movement Patterns Recognition Part II).

The global project consists to recognize dance movement patterns using the **Hidden Markov Model (HMM)**. In this way a greater interaction can be obtained.

We have formed part of the **Human Media Interaction (HMI)** group at the University of Twente where interactions between people and machines are the main goal.

There, we have worked in a project named "**Interacting with a virtual dancer**". The Virtual Dancer and the user dance together. It aligns dance moves in real time to the beat of the music, adapting its style to the observed user information captured through real time computer vision. Alternating between following the user and taking the lead, the system achieves a mutual dancing interaction.

After studying the application and thinking about how we could improve it we reached the conclusion that we could try to recognize dance movement patterns.

The author tried to make this report as comprehensible as possible, explaining concepts and accompanying them with figures and diagrams to give a global vision of the topic.

The purpose of this report is to provide to the reader the necessary knowledge to recognize dance movement patterns using the HMM, a consolidated model used for speech recognition with successful results.

1.2 Goals

The global project goals are the following ones:

- 1st) to start up The Virtual Dancer application (computer vision part and artificial intelligent part).
- 2nd) to study the application and to think about how we could improve it.
- 3rd) to recognize simple dance movements.
- 4th) to recognize dance movement patterns (a group of simple dance movements).
- 5th) to generate dance movement search patterns automatically.
- 6th) to find patterns without a reference (without a dance movement search pattern).

The first thing that we have to do is to start up the application. It has two parts: the computer vision part, implemented with C++ and the artificial intelligent part, implemented with Java; each one running on different computers.

To know how we could improve the application we have to study it with accuracy. It is the second goal; we want to do a great improvement. For that reason the study has been extensive and hard.

The third goal consists to receive the parameters from the computer vision (x-axis variation, y-axis variation, center of mass, person detector and so on), to process it and to recognize simple dance movements (for example left step, right step, jump, duck, twister and so on).

In order to obtain the fourth goal we need a pattern to look for it (for example left step + right step + jump + twister). Then, we recognize when the pattern appears during the user dance in real time. If the user does the pattern (left step + right step + jump + twister) we recognize it.

To carry out the fifth goal we have to dance in front of the application and it generates the pattern automatically (with the corresponding graph and probabilities that we need to implement the Hidden Markov Model). After it, we can use the generated pattern to do a search. Another way to do the pattern is to make it manually but it is costlier and slower.

The idea of the last goal is to find patterns without a reference. Normally, we know what patterns we want to find. In this goal we have a dance, and we want to find repetitions, but we do not know what repetition.

This part of the report includes all the previous study (when we though about how could improve the application), a general vision about The Virtual Dancer application, the computer vision part and a little introduction about the artificial intelligent part (the first, second and part of the third goal are included in this part of the report).

1.3 Report Organization

The memory is organized in eight chapters, and each chapter is divided in different sections.

The first chapter presents and describes the goals of the project and how the memory is organized, giving a small description of each chapter.

In the second chapter we explain The Virtual Dancer application. How it is made, the different parts of it, what it does and so on.

In the third chapter we comment the computer vision system ParleVision. This system is used by the Virtual Dancer to receive the user's movements.

We show the work environment with the setup application and the used material, in the fourth chapter.

In the fifth chapter we can see the study carried out at the beginning of the project. This study served us to understand the application and to think about how we could improve it.

In the sixth chapter we show the modifications done in the computer vision system ParleVision: the new variables created and sent to the Artificial Intelligent part, the new features added to the application, and so on.

In the seventh chapter we explain the conclusions of this project: the reached and not reached goals, the problems that we have found during the application development, the possible improvements ...

And in the last chapter the references are shown which are used to do this part of the project and this part of the report (Books, articles and URLs).

Chapter 2. The Virtual Dancer Application

The following section consists of a small explanation about the Virtual Dancer Application; what it does, how it does, and so on. In this chapter we use information from a HMI document (reference [1] and [16]).

2.1 Introduction

The Virtual Dancer is an Embodied Conversational Agent (ECA), which dances together with the user following the beat of the music. It adapts its performance to whatever the human user is doing, introducing new movements, who is observed using real time computer vision.

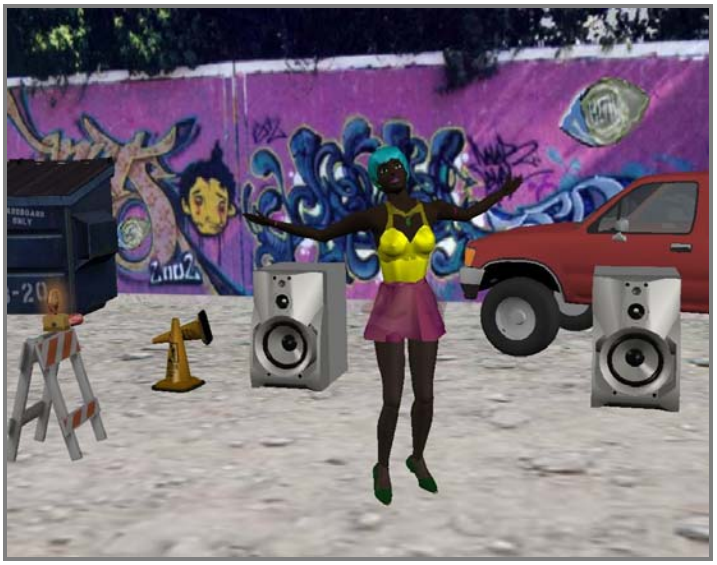


Figure 1. Virtual Dancer Application

The basic idea of the application is to monitor movement global characteristics of the user, and then, use those characteristics to select and adapt movements for the Virtual Dancer. A particularity of this application is that in contrast to other ones, the user is simply invited to dance together with the Virtual Dancer; any interaction patterns and implicit relations between the dance behavior of the human and the Virtual Dancer should be evoked intuitively without explicit appeal.

Next, the architecture of the system is discussed.

2.2 Architecture

The architecture of the system is shown in Figure 2. The Virtual Dancer is projected on a screen and the user is observed by a camera that is placed above the screen, monitoring the area in front of it. A dance pad is placed in front of the screen and a sound system with speakers is used to play the music, which the user and the Virtual Dancer can dance.

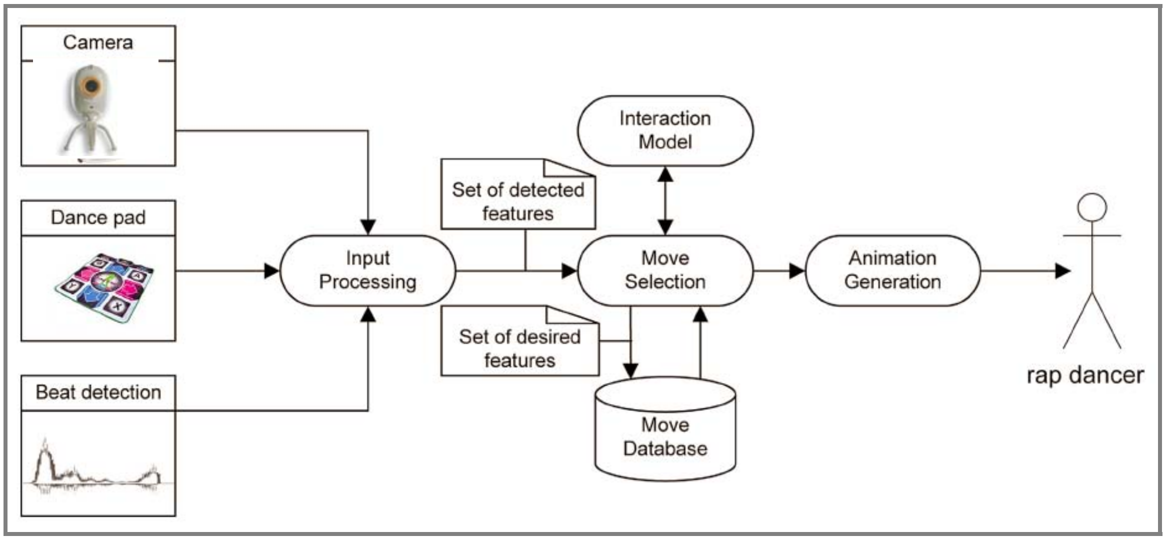


Figure 2. Virtual Dancer Application Architecture

The different components of the architecture are explained in this section.

- Beat Detection
- Video Analysis
- Dance Pad
- Move Database
- Move Selection
- Animation Generation
- Interaction Model

2.2.1 Beat detection

The virtual dancer has the ability to interpret the music and to find the beats to align it with the music. The application implements a beat detection algorithm (Klapuri's algorithm) that detects the tempo and beat in the music played.

2.2.2 Video Analysis

The system observes the movements of the user using the computer vision system ParleVision (see Chapter 3) and a single video camera. For that reason there are advantages and disadvantages, as for example:

- No depth information is available.
- It is possible to have large variations in appearance and body dimensions between users.
- The application needs to extract certain characteristics from the movements of the user and react to them. When poses are described in great detail, it is not trivial how these can be used in the dancer's move selection phase.

The application uses global movement features as for example:

- User's Silhouette
- Center of Mass
- Radial Activity

User's Silhouette

This method requires a known background model, but it is computationally inexpensive. Moreover, silhouettes encode a lot of information about the user's pose. The application employs two image processes to recover the movement features.

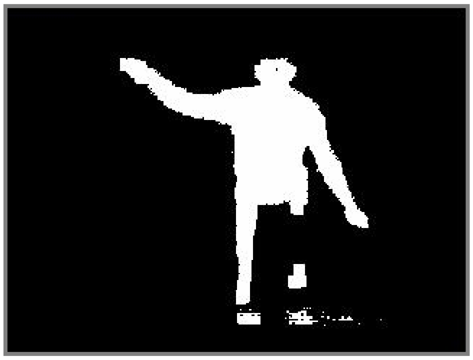


Figure 3. Extracted Silhouette

Center of Mass Detector

The center of mass detector uses central moments to determine the 2D location of the silhouette's center of mass (CoM).

Two thresholds are set on the vertical component of the CoM. These threshold values are determined empirically. Furthermore, the average difference in successive values of the horizontal component is a measure for the horizontal activity value. This value is normalized with respect to the silhouette's width.

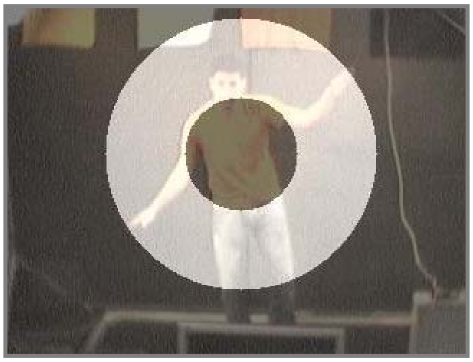


Figure 4. Center of mass with ring

Radial Activity Detector

Taking the distribution of silhouette pixels around the CoM into account, we are especially interested in the extremities of the silhouette (legs and arms). A ring divided into 16 radial bins is used to determine its movements. A threshold on the percentage of active pixels is determined empirically. In addition, the radial activity value is determined by the normalized average change in the bin's values between successive frames.

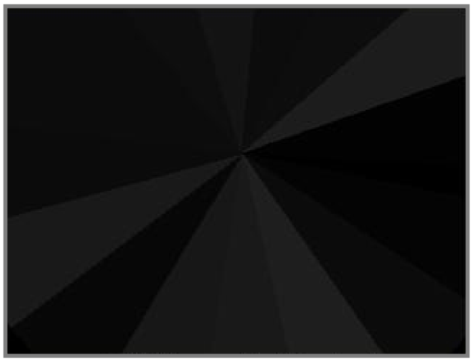


Figure 5. Radial activity bins

2.2.3 Dance Pad

In order to recognize feet movements a Dance Revolution (DDR) pad is used. This pad contains eight 'buttons' that are pressed if a foot is placed on them. The main idea is to determine how many times a button is pressed in a given period of time.

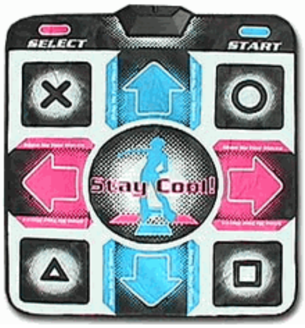


Figure 6. DDR

2.2.4 Move Database

A human pose is described as a variation of joint's rotation values. The animations are defined as a number of keyframes that describe poses (it can be specified manually, by motion capture or using the location of end effectors), and interpolation between them.

Using inverse kinematics (IK), the rotation of joints involved in the animation is determined. In a similar way, formulae that describe joint rotation paths are defined. Then keyframe animation, rotation formulae and path descriptions, for limbs and body center, are combined. The move key positions, which are aligned to the beats in the animation phase, are stored.

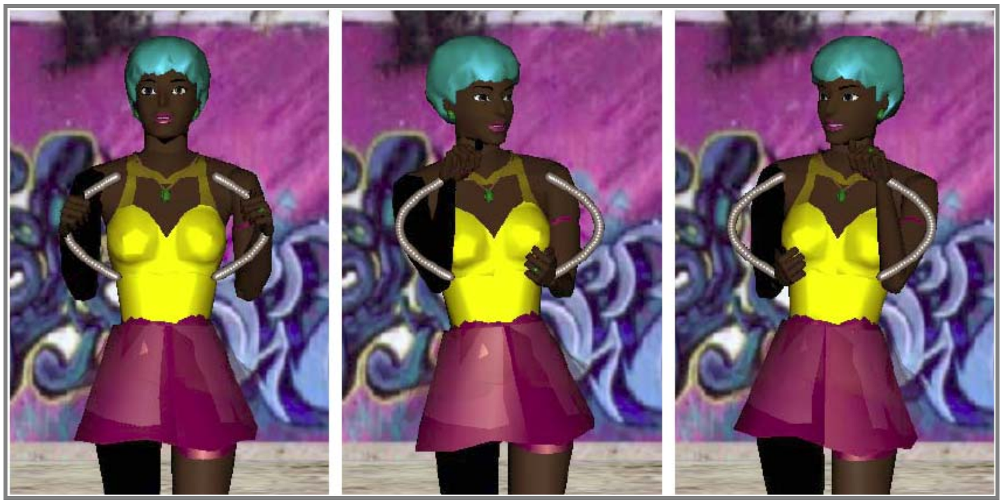


Figure 7. Movement example: The arms are rotated like a driver wheel turn. The path of the hands is shown by the white spheres.

2.2.5 Move Selection

The move selection is built to choose moves based on the current state of the Dancer and the characteristics of the dancing behavior of the human. A mapping between this information and the stored information about each move determines the next move selection of the Dancer. In the database the type of the movement (e.g. dancing, bored and so on) and the default duration are annotated.

To select a move, a set of observed characteristics from the human dancer is calculated. These features are mapped to a set of desired characteristics in the dance move. The movement is selected depending on this map.

2.2.6 Animation Generation

Dancing to the Beat

One important feature in any dance animation is the alignment of the dance movements with the beat of the music. Whenever a new move is being planned, the beat detector module is queried to know the current tempo and beat pattern of the music. This information is used to produce a vector of beats' predictions in the near future. The set of key points, from the selected move, and the beats, from the beat prediction vector, are time-aligned between them.

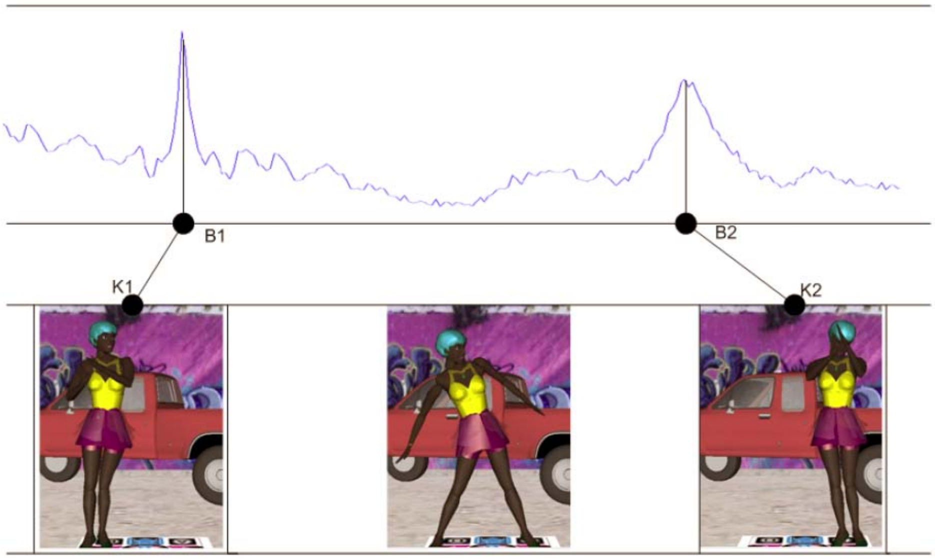


Figure 8. Move alignment to the beat: beat *B*1 is aligned to keyframe *K*1 beat *B*2 is aligned to keyframe *K*2.

Interpolation

To generate a transition from one dance move to the next, a simple interpolation algorithm has been used. The root position is linearly interpolated from the end position of the previous animation to the start position of the next animation. If there is not significant feet displacement, all joint rotations are interpolated. If significant feet displacement is needed, from the previous animation to the next, the dancer makes two intermediary steps.

2.2.7 Interaction Model

The interaction model is implemented as a state machine. Currently it has three states:

- Bored
- Invite
- Dance

During the 'bored' state, the Dancer exhibits bored behavior such as scratching the head or inspecting the fingernails. If the presence of a human is detected by the computer vision, the application tries to invite the user to dance (using nonverbal invitation gestures). Once the user steps on the dance pad, the dance starts. The application can follow or to lead the user (or al least try to do it). 'Following' means to dance using similar movements showed by the user. 'Leading' involves varying the movement properties considerably in one or more dimensions. The implicit intention is that the user reacts adapting himself to the application.

2.3 Summary

The Virtual Dancer is an ECA that invites a user to dance. The system observes the movements of the user using the computer vision system ParleVision. It extracts global characteristics about the movements of the human dancer.

The Virtual Dancer uses a beat detector which lets it to dance following the song's rhythm. Moreover, it uses a database with many different dance moves where it selects the most appropriate. Then, these moves are timed with the beat. The transition from one move to the next is made using an IK-generated stepping motion and interpolation techniques.

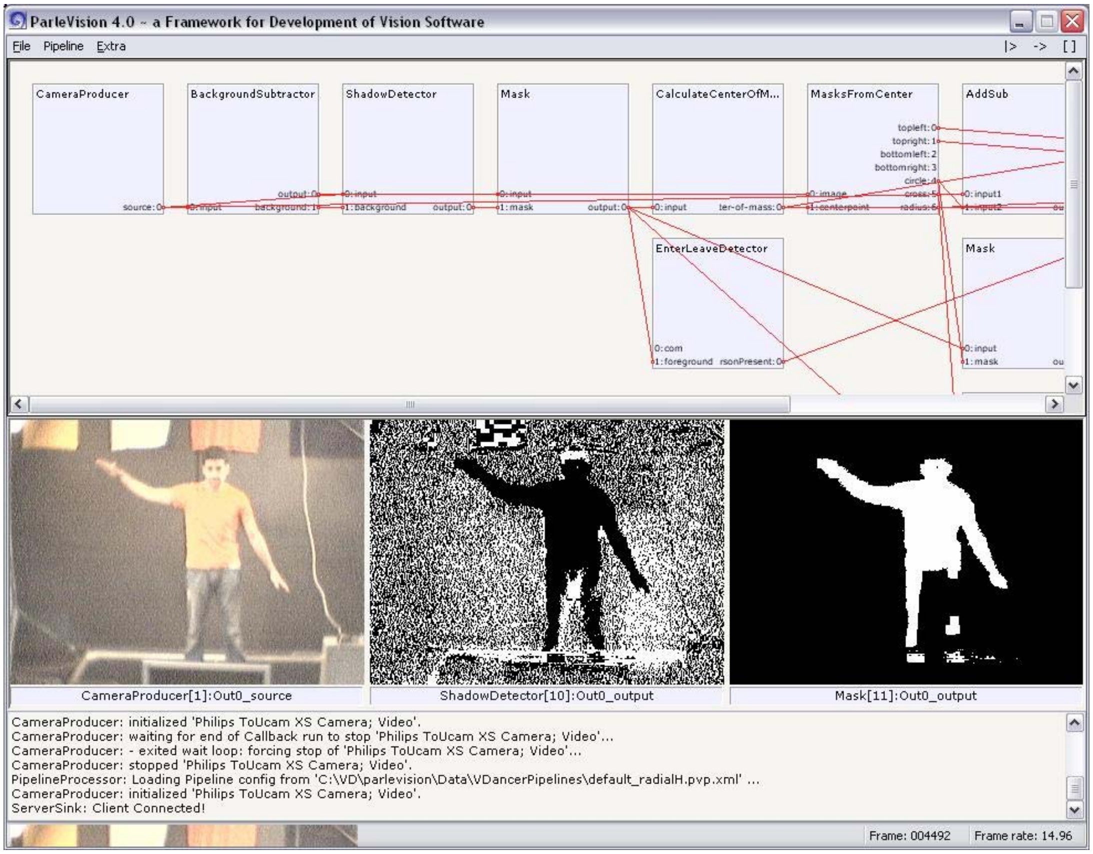
The application alternates between to follow the user and to take the lead. Then, dancing interactions between user and virtual dancer are achieved.

Chapter 3. The computer vision system - ParleVision

In the following section we are going to have a little explanation about the computer vision system Parlevision; what it does, how it does, and so on. In this chapter we use information from a HMI document (reference [2] and [17])

3.1 Introduction

ParleVision is a software that aims to be a generic framework for development of computer vision projects. It provides means to easily experiment and to extend the existing components in the framework. Most tedious tasks like internal data communication, visual representation and configuration files are done by the framework and it does not need to be implemented by users. ParleVision also provides a graphical way to visualize and to work with the Processors. It allows a clear view of the data flows between the Processors.



```
CameraProducer: initialized 'Philips ToUcam XS Camera; Video'.
CameraProducer: waiting for end of Callback run to stop 'Philips ToUcam XS Camera; Video'...
CameraProducer: - exited wait loop: forcing stop of 'Philips ToUcam XS Camera; Video'...
CameraProducer: stopped 'Philips ToUcam XS Camera; Video'.
PipelineProcessor: Loading Pipeline config from 'C:\VD\parlevision\Data\VDancerPipelines\default_radialH.pvp.xml' ...
CameraProducer: initialized 'Philips ToUcam XS Camera; Video'.
ServerSink: Client Connected!
```

Figure 9. Parlevision System

3.2 Features

Parlevision is written in C++ and uses Intel's Open-CV image processing library. This library contains many optimized image processing functions. Parlevision is a graphical interface for this library and it contains many higher-order functions.

A processor is simply a function that transforms inputs to outputs. The data from input to output are split up into filters, each one have a distinct function. The input and output pins can be connected to pins of other processors (e.g. a Processor can receive data from other Processors on its input pins). Then it can process this incoming data to produce new data that can be exported through the output pins.

There are various types of data, ranging from single and triple channel images to coordinates of points and rectangles. Input and output types can be:

- Images
- Bounding boxes
- Numbers
- Any user-defined type.

Only pins with similar types can be connected and each processor can have many parameters, each one can be specified by the user.

A group of processors is called a pipeline. A pipeline typically starts with a producer and it has one or more processors connected. One type of Processor can be used more than once in a Pipeline. A pipeline can be stored, with all the processor's parameters included. This allows a fast settings' change.

Within Parlevision, many processors are included:

- Producers for camera, video and image input.
- Processors that operate on images include edge detection, background subtraction, color detection, cropping, adding and subtracting and masking.
- Searching for the largest blobs, finding shadow and point tracking.

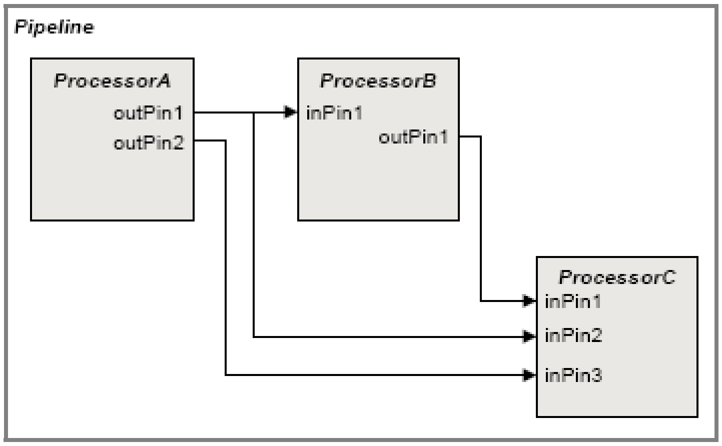


Figure 10. Pipeline diagram with three processors

Parlevision contains a development environment. It is possible to add new processors using the processor development kit (PDK).

The Parlevision system is used in many applications including tracking of hands, estimating human poses, corridor surveillance, analyzing beats and analysis of facial expressions.

Chapter 4. Work Environment

Next a general vision of the work context will be made.

4.1 Material

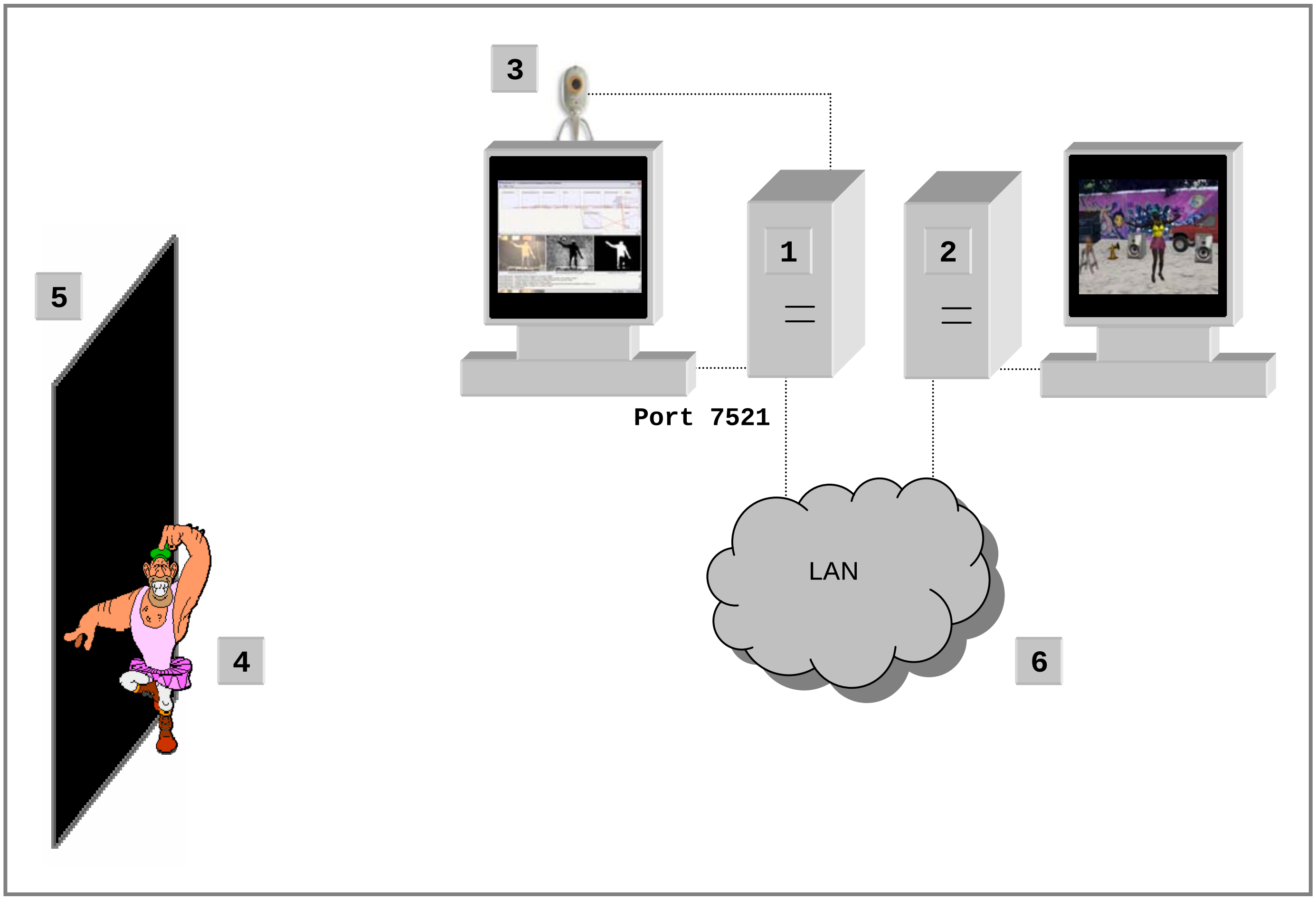
To carry out the project we count with:

- Hardware
    - 2 computers with the following properties
        - Pentium 4 2.8GHz
        - 512 MB of RAM
        - Microsoft Windows XP with the SP 2
    - 1 camera
        - Philips ToUcam XS

- Software
    - ParleVision
    - Virtual Dancer

- Others
    - Human Dancer
    - In our case, a black background
    - Dance floor
    - For us is not necessary the Dance Pad
    - A tube to support the camera
    - t-shirts with different colors

4.2 Setup



1	Computer Vision PC
2	Artificial Intelligent PC
3	Camera
4	User
5	Background
6	LAN

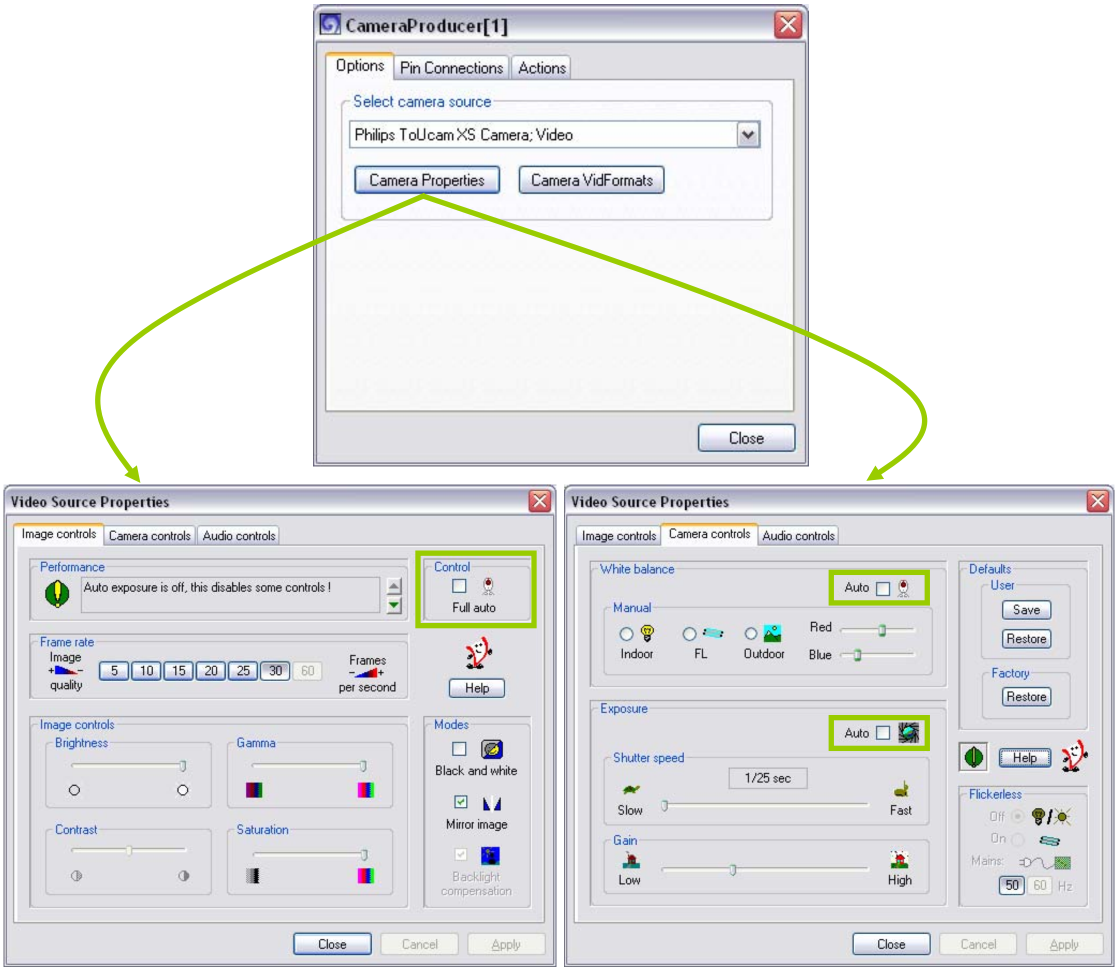
Figure 11. Setup Application

4.3 Start up the Applications

We did not have problems to start up the two applications, but it was too difficult to install the camera. We did not have the drivers of the camera and the camera manufacturer did not give us the necessary software to be able to use it.

After a lot of hours searching on the Internet we could make it to work. Using different parts of others cameras drivers we “made” a driver for our webcam. It was a hard work.

Then, in the cameras options, we only took in count the automatic options. We disabled all the automatic options to avoid the light changes in the captured images. We selected 30 for the frame rate of the camera.



Select camera source: Philips ToUcam XS Camera; Video

Auto exposure is off, this disables some controls !

Shutter speed: 1/25 sec

Figure 12. Camera Properties

Chapter 5. Previous Study - Interacting with a Virtual Dancer

In this chapter, the previous study that we did before the start of the project is explained. We started thinking about how we could improve the interaction between user and application.

5.1 Interaction Dimensions

We have thought that the best way to divide the interaction is to create 5 different dimensions, where each one contains a different characteristic that we are going to explain in the following points:

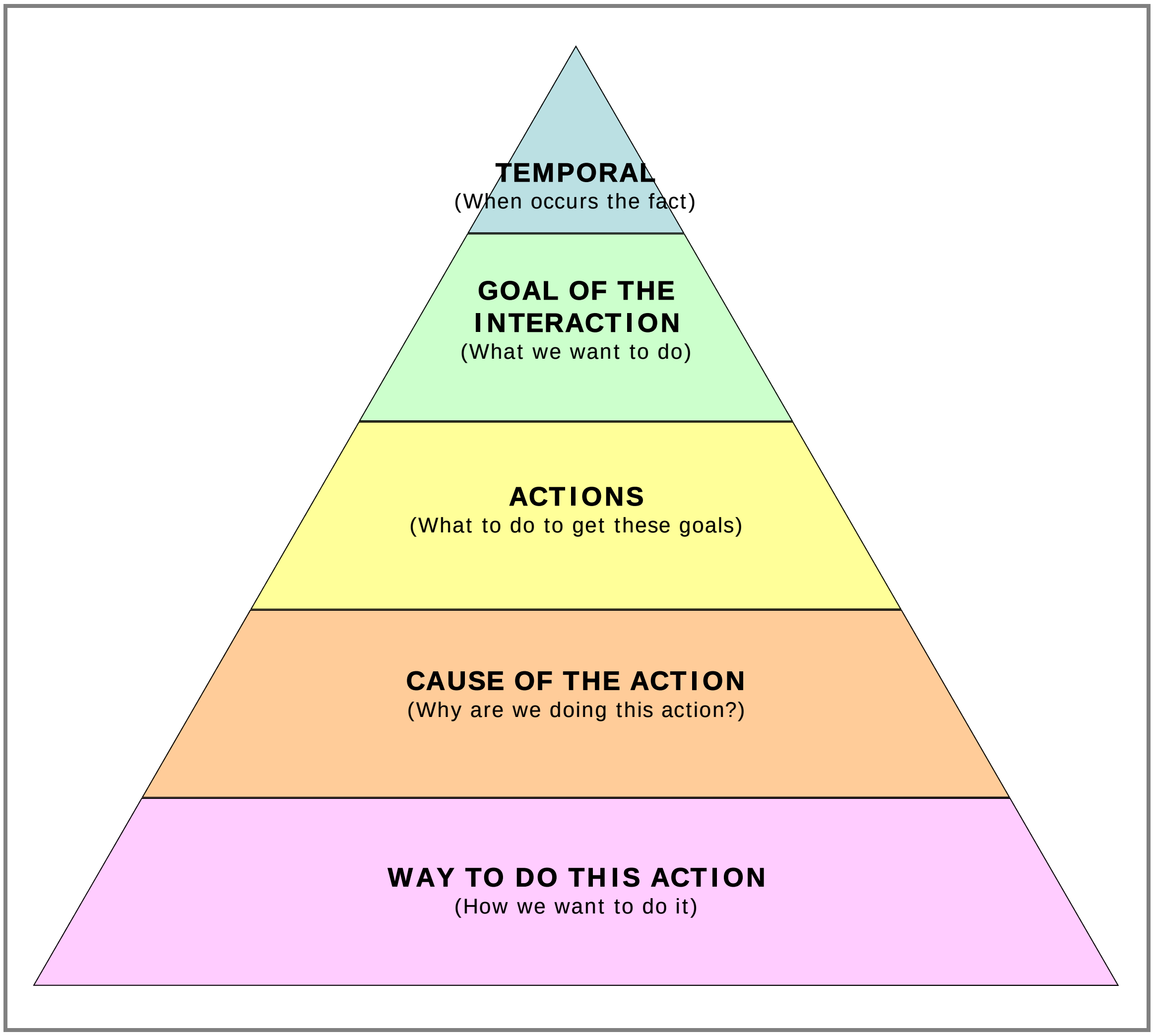


Figure 13. Interaction dimensions

5.1.1 Temporal dimension

The first of our dimensions is used to express when the action occurs, and we have decided to divide this dimension in four different states:

Before the song

This group should contain all the possible situations that occur when the song is not already played.

During the song

Here we have the actions occurred during the dance.

After the song

Here we have the actions that happen at the end of the song.

Between songs

Finally we have a special temporal state that is produced when the user has danced one song and he wants to follow dancing.

5.1.2 Goal of the interaction

The second of our dimensions is used to know what the goal of the action that occurs is, but in a general way, we only have 3 different goals and in these we can include everything that can happen.

To attract the attention

In this goal we have the actions that we use to get that the user look at us, and pay attention to the application.

To entertain

With the actions included in this goal, we want the user to enjoy himself.

To leave a good memory

Finally we want to get that the user has a good memory of the application.

5.1.3 Actions

In each one of these goals we have included some actions, that will help us to reach it, and we are going to explain these actions following:

To attract the attention

- To say hello

  Here we have included the actions to welcome the user. This is produced in the "To attract the attention" goal.

- To invite the user

  We have a group of actions to ask the user if he wants to dance. We have to use these actions in the "To attract the attention" goal.

- To be seem

  These actions only are used if we want that the user (or possible user) look at us. It can be necessary as in "before the song" as in "during the song" time, but booth are included in the "To attract the attention" goal.

To entertain

- To Start the dance

  Here we show the user how to warn. We must to do these actions in the "To entertain" goal.

- To Encourage the user

  This big group of actions includes all the possible actions that we could use to cheer the user and he feels good with our system. We can need this group as in "Before the song" as in "During the song" as in "Between songs" time. Depending on where we are, we use a different kind of animations.

- To do a choreography / chorus

  We are able to do some known choreography if we know what song is and when its chorus is. It only can happen in the "during the song" time.

- To enrich the dance

  These movements are introduced by the agent to enrich the dance, doing some random movements during the dance or responding to some user's movement. It is included in the "during the song" time.

- To explain some step

  We can use the time between two songs to explain to the user some step that we are going to do during the dance. This is included in the "Between songs" time.

To leave a good memory

- To congratulate

  At the end of the dance, usually we congratulate (with more or less enthusiasm depending of how the user has danced) the user. It is included in the "After the song" time.

- To say goodbye

  When the user leaves the application, we can say good bye to the user in some different ways. It is also included in the "After the song" time.

5.1.4 Cause of the action

When we have decided what action we want to do, we have 2 ways to do this action:

Reactive

The user does something that causes our reaction.

Deliberative

We do not expect anything; simply, we decide to do some action.

5.1.5 Way to do this action

Now we only have to decide how to do this action:

Verbal

Here we have a list of actions with the feature that we must speak to give sense to the action.

Non Verbal

In this list we have the actions done by means of gestures (and possibly some sound).

5.2 Interaction Tree

Next, we are going to show the different actions that could take our agent according to our state in the interaction tree.

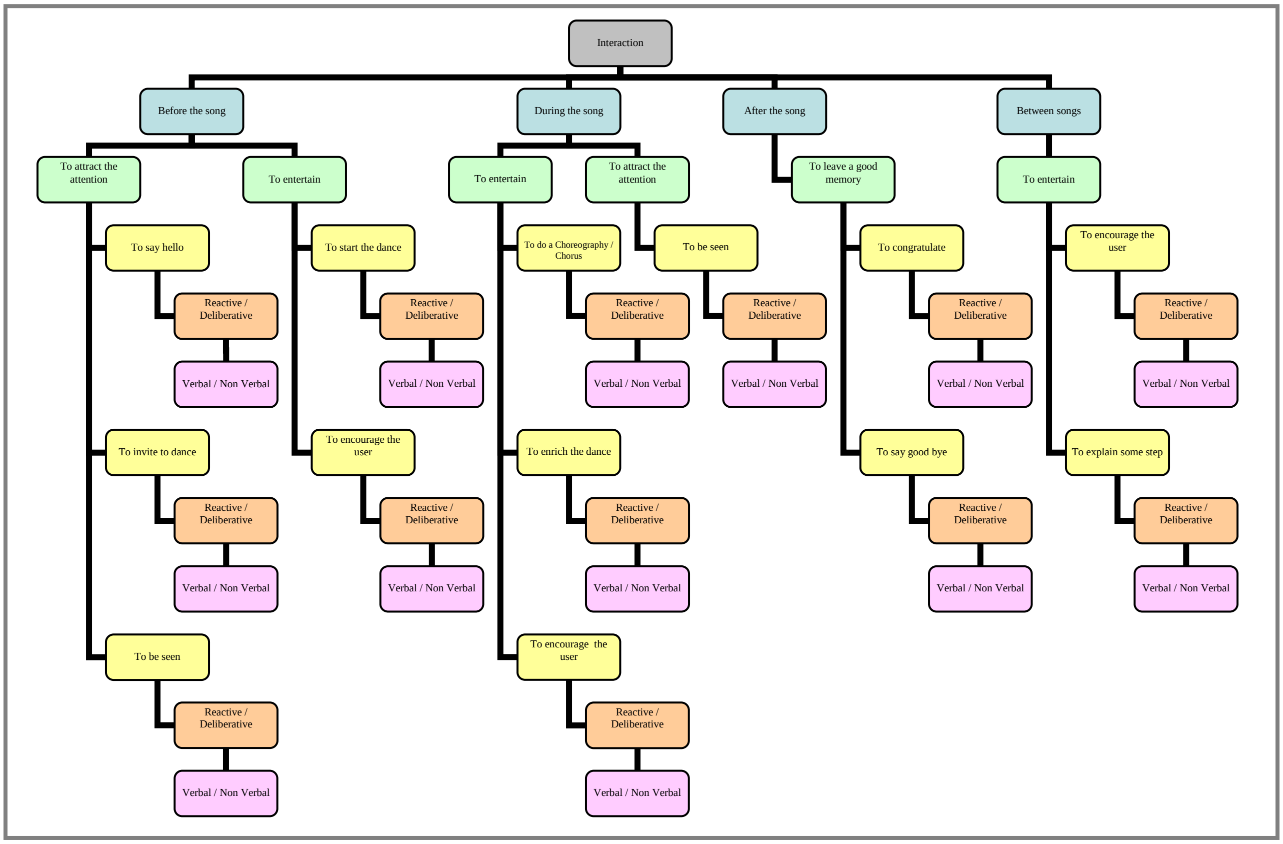


Figure 14a. Interaction tree (see next page, figure 11b)

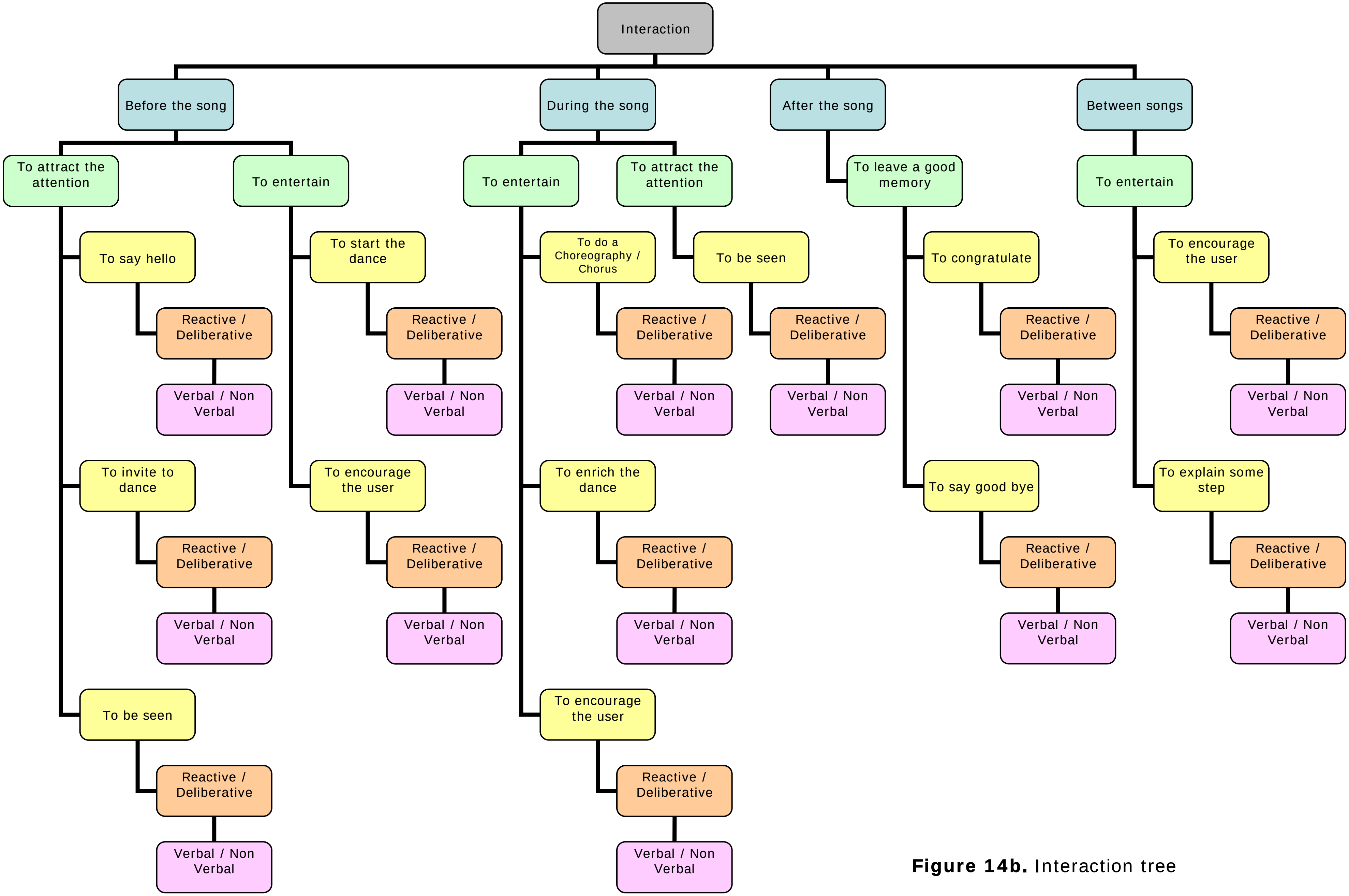


Figure 14b. Interaction tree

5.2.1 Before the song

<u>To attract the attention</u>

- **To say hello**:
    - Reactive
        - Verbal

Human	Agent

- Non verbal

Human	Agent

- Deliberated
    - Verbal

Human	Agent
	“Hi, are you ready to spend a good time?”
	“Hi friend, Do you want to dance with me?”
	“Welcome to the floor dance”
	“Thanks to dance with me!”
	“Hello!!”

- Non verbal

Human	Agent
	To shake the hand
	To move the head affirmatively
	To up the eyebrows
	To smile
	To bow

- **To invite to dance**:
  - Reactive
    - Verbal

Human	Agent

    - Non verbal

Human	Agent

  - Deliberated
    - Verbal

Human	Agent
	"Do you want to dance?"
	"Would you like to dance with me?"
	"Are you warmed? Show me it then"
	"Do you invite me to dance?"
	"Can I dance with you?"
	"I Wish I could dance with you"

    - Non Verbal

Human	Agent
	To walk near the user
	To do some step looking at the user
	To move the hands
	To extend the arm towards the user
	To point at the ear

- **To be seen**:
  - Reactive
    - Verbal

Human	Agent
The CV detects the user	“Eiii brother!!!”
The CV detects the user	To whistle
The CV detects the user	“Where do you go?”
The CV detects the user	“I was waiting for you”
The CV detects the user	“Buuu” (going to the screen)

    - Non verbal

Human	Agent
The CV detects the user	To jump moving the arms
The CV detects the user	To knock to the screen
The CV detects the user	To do a sensual movement
The CV detects the user	To kiss to the user
The CV detects the user	To shake the hands
The CV detects the user	To dance alone

  - Deliberated
    - Verbal

Human	Agent

    - Non verbal

Human	Agent

To entertain

- **To Start the dance**:
    - Reactive
        - Verbal

Human	Agent

- Non Verbal

Human	Agent

- Deliberated
    - Verbal

Human	Agent
	“Let’s go!!”
	To indicate that I like this song
	To indicate that I am going to take the initiative and to start to dance
	To tell to the user that I am going to follow him and to wait for his movement

- Non verbal

Human	Agent
	To indicate the rhythm moving the hand
	To indicate that I like this song

- **To Encourage the User**:
    - Reactive
        - Verbal

Human	Agent

- Non verbal

Human	Agent

- Deliberated
    - Verbal

Human	Agent
	“You seems a good dancer”
	“Come on!!!!”
	“Do you think you could follow me?”
	“I want to see your body in movement”
	To Shout “Weeeeeeeeeee”, “iiiiaaaaaaaaaahhhh”…
	“Show me what you can do”

- Non verbal

Human	Agent
	To clap
	To throw the fist to the air
	To point at the user
	To move the head affirmatively
	To send a kiss
	To wink one eye

5.2.2 During the song

<u>To entertain</u>

- **To do a Choreography / Chorus**:
    - Reactive
        - Verbal

Human	Agent

- Non verbal

Human	Agent

- Deliberated
    - Verbal

Human	Agent
	"Follow me" (and some choreography)
	"Come on, we are going to animate this party!" (and some choreography)
	"Look these steps" (and some choreography)
	"Now begins the choreography!" (and some choreography)
	"Do you know this step?" (and some choreography)
	"Ohhh your steps are very funny"
	"Show me some steps" (and some choreography)
	"Great chorus!"

- Non Verbal

Human	Agent
	To mark the rhythm with the hand
	To clap
	To put booth arms up and move it
	To point at the ear
	To point at the feet

- **To enrich the dance**:
    - Reactive
        - Verbal

Human	Agent
If the user is too far (1)	To say "come here I don't bite"
If after to be dancing for a time, the human leaves to dance (1)	To ask him about if he is tired
If after to be dancing for a time, the human leaves to dance (2)	To tell him that he has danced very well and to admit that I am also tired
He is under the rhythm	To indicate him that he has to move more quickly
He is over the rhythm	To indicate him that he has to move more slowly
If he do the same movement by a long time (2)	To say that he already controls this movement and to wait for a movement change of the human.
If the agent takes the initiative and the human does not follow the agent (1)	To try to encourage the human again, and to follow with the initiative
Sounds Macarena's song	To dance Macarena's dance
If the user is stopped	To ask him if he does not like this song
If the user moves his head negatively	To request another song
If the user moves his head affirmatively	To animate to follow dancing
If he goes out in the middle of the song	To Ask him "do you leave me alone?"
If he goes out at the end of the song	To Tell him "see you!"
If he keeps his arms down	To animate him to move his body more
Between song and song	To do some funny comment ("do you want to drink?")
If there are more than one person	To invite to the other person to try it too
If there are more than one person	To ask who of them want to dance first
Random	To tell him that she does not like this kind of music
If he always keeps steeped the same buttons	To tell him that he should move his legs
If he steeps the dance pad too hard	To tell him that he could break the dance pad
If he steeps the dance pad too soft	To tell him that he dances as a ballet dancer
If he is steeping many buttons at same time	To request him to center his position
If there are more than one person over the pad (1)	To tell them than only can play one
If there are more than one person over the pad (2)	To tell them than they should try to combine themselves better

- Non verbal

Human	Agent
If the user is too far (2)	To follow dancing, but looking at the human
If he does a complex movement	To congratulate to the human
To dance following the beat (2)	To dance without following the beat (between beat and beat)
He is under the rhythm	To indicate him that he has to move more quickly
He is over the rhythm	To indicate him that he has to move more slowly
If he does the same movement for a long time (1)	To take the initiative
If the agent has the initiative and the human does not follow the agent (1)	To try to encourage the human again, and to follow with the initiative
If the agent has the initiative and the human does not follow the agent (2)	bored stated
If the agent have the initiative and the human does not follow the agent (3)	To follow the human
To do a choreography	To detect it and memorize it to be used after
To do a known choreography	To detect when begins this choreography and to do it
Sounds macarena's song	To dance macarena's dance
If the user move his head affirmatively	To animate to follow dancing
If he keeps his arms down	To animate him to move more his body
Between song and song	To do some funny comment ("do you want some drink?" "are you enjoying?")
To follow the agent	To do some choreography (for example front step + back step) and using the dance pad to check it
If there are more than one person	To invite to the other person to try it too
To play with another person	To dance with both people at same time
To chose the kind of dance	To adjust her movements to this kind of music
If he is on the 3 first buttons of the dance pad	To dance near the screen
If he is on the 3 second buttons of the DDP	To dance far of the screen
If he is on the 3 third buttons of the dance pad	To dance near the screen

- Deliberated
    - Verbal

Human	Agent
	“Good movement!!”
	“You are out of rhythm”
	“Who has shown you this step”
	“Well done!”
	“Look at me!”
	“Come on!!! I love this!!!”
	“Que pasa nenggggg”
	“I had never enjoyed so much like now”
	“I wish it never finishes”
	“I’m beginning to be warm”
	“Aaaaaazuuuuuuucarrr”
	“You have been practicing at home!”
	“Are you warm?”

- Non Verbal

Human	Agent
	To jump
	To clap
	To put booth arms up and move it
	To throw the fist to the air
	To clap the fingers
	To run from one side to the other side
	To duck
	To turn around herself

- **To Encourage the User**:
  - Reactive
    - Verbal

Human	Agent

- Non verbal

Human	Agent

- Deliberated
  - Verbal

Human	Agent
	“Come on!!!!”
	“Well done, but now I want more rhythm”
	“Do you only know to do it?”
	“Very good!!!!!!”
	“Could you keep this rhythm?”
	“Don’t tell me that you are tired”
	“That’s all? I’m sure you can do it better”
	“Good song, did you ear it before?”

- Non verbal

Human	Agent
	To clap
	To throw the fist to the air
	To point at the user and to move the head
	To Move the head affirmatively
	To put the big finger up
	To do the OK symbol with the hand

To attract the attention

- **To be seen**:
    - Reactive
        - Verbal

Human	Agent

- Non verbal

Human	Agent

- Deliberated
    - Verbal

Human	Agent
	“Look at me!!!!”
	To whistle
	“Buuu” (going to the screen)

- Non Verbal

Human	Agent
	To jump moving the arms
	To do a sensual movement
	To kiss to the user
	To shake the hands

5.2.3 After the song

<u>To leave a good memory</u>

- **To Congratulate**:
    - Reactive
        - Verbal

Human	Agent

- Non verbal

Human	Agent

- Deliberated
    - Verbal

Human	Agent
	“I have enjoyed”, “Have you enjoyed?”, “Well done!!”
	“I hope to arrive to your level some day”
	“it has not been bad but you should practice more”
	“I think that you could dance better”
	“Your mother can be proud”
	“Don’t lie me, you have taken some dance lessons”
	“Don’t worry, the next time will go better”
	“At the end you have improved a lot!”

- Non Verbal

Human	Agent
	To bow, to clap, to wink the eye
	To throw the fist to the air
	To point at the user and to move the head
	To move the head affirmatively
	To put the big finger up
	To do de OK symbol with the hand

- **To say good bye**:
    - Reactive
        - Verbal

Human	Agent

- Non verbal

Human	Agent

- Deliberated
    - Verbal

Human	Agent
	“See you!”, “Bye bye”
	“I hope to see you again”
	“Come back when you want”
	“I will be waiting for you”
	“Don’t leave me here!!!, well see you…”
	“Practice a little bit and show me your improvement”
	“You seems tired, take a break, I wait you here”
	“Do we rest?”

- Non verbal

Human	Agent
	Good bye hand movement, To up one hand
	To point at your heart and to point at the user
	To up the head
	After other movements to turn herself and to go away
	To throw some kisses
	To do an army salute

5.2.4 Between songs

To entertain

- **To Encourage the User**:
  - Reactive
    - Verbal

Human	Agent

    - Non verbal

Human	Agent

  - Deliberated
    - Verbal

Human	Agent
	“Hey, it’s not bad, but you can improve a lot”
	“Well done, but now I want more rhythm”
	“Do you only know to do it?”
	“Could you keep this rhythm?”
	“Don’t tell me that you are tired”
	“That’s all? I’m sure you can do it better”
	“The next song will be better”
	“I have already seen your level, now, follow me”
	“Now I’m going to follow you”

    - Non verbal

Human	Agent
	To clap, To put the big finger up
	To throw the fist to the air
	To point at the user and to move the head affirmatively
	To do the OK symbol with the hand

- **To explain some step**:
  - Reactive
    - Verbal

Human	Agent

    - Non verbal

Human	Agent

  - Deliberated
    - Verbal

Human	Agent
	"Look at me and try to learn this step" (and do some steep)
	"At the next song we are going to do this step" (and do some steep)
	"Have you seen it before?" (and do some steep)
	"Follow me!" (and do some steep)
	"Look at this" (and do some steep)
	"Could you do it?" (and do some steep)

    - Non verbal

Human	Agent
	To point at your eye and after, to point to your feet (and do some steep)
	To point at your feet with booth hands (and do some steep)
	To go to the screen and to point at your eye (and do some steep)

5.3 Interaction State Machine

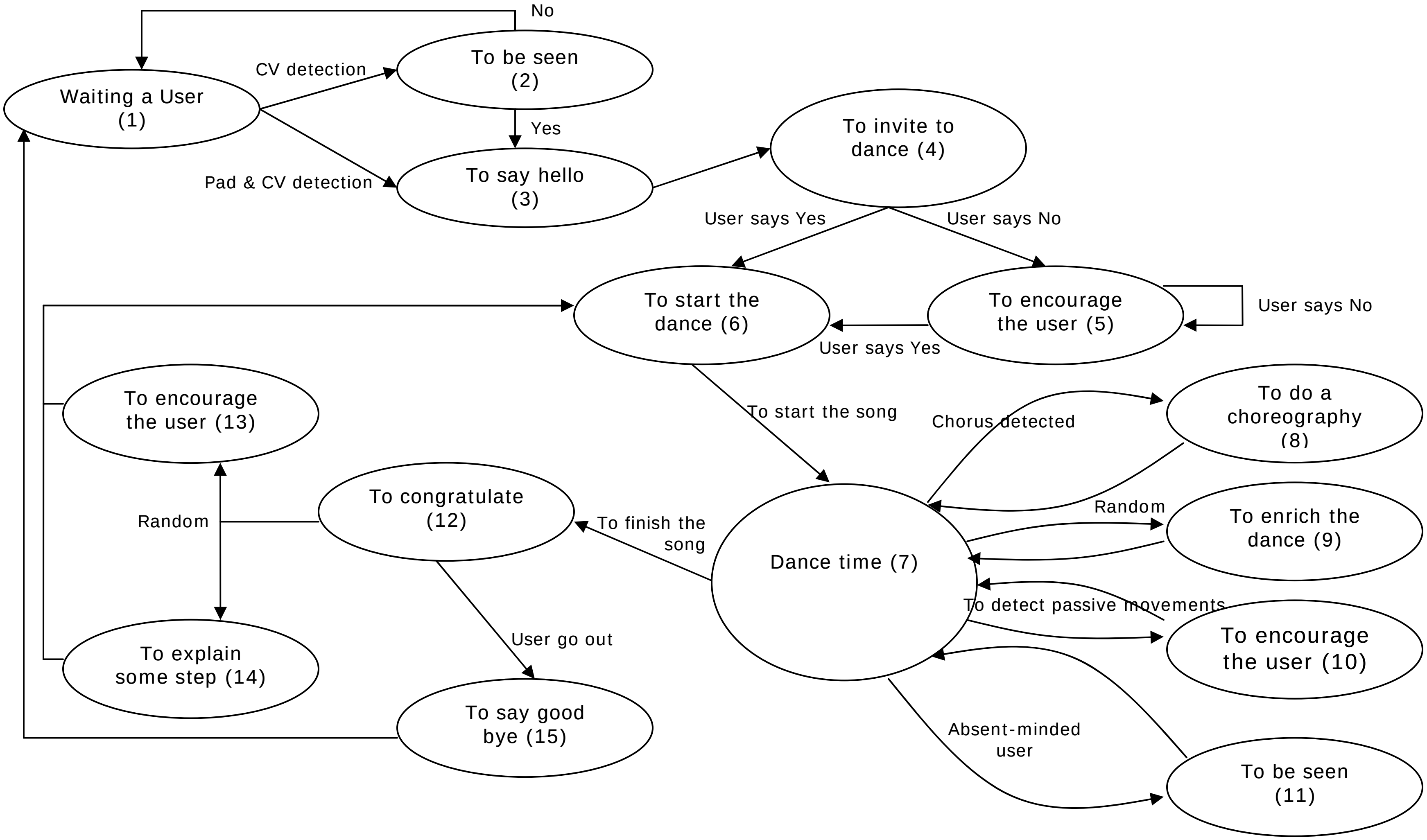


Figure 15. Interaction state machine

Here we can see the interaction state machine which shows us the different states and transitions included in the interaction between the user and the application. In brackets we show the number of the sate.

5.4 Available variables

In this section we can see all the variables that remain at our disposal. We can use variables from the Computer Vision, from the Pad and from the beat predictor. Not all the variables that are explained here are used to do the project, but to make the study we kept in mind all the possibilities, all the available variables.

After seeing the variables that were available, we saw the need to create new ones, like for example the X, Y and radius variation between frames. The created variables are shown in chapter 6.

5.4.1 CV Variables

CVInfoFrame class

Name	Type
PersonPresent	Boolean
XCenter	Int
YCenter	Int
LeftHandTop	Boolean
RightHandTop	Boolean
Radius	Int
PercentageLeftHandTop	Int (1-100)
PercentageRightHandTop	Int (1-100)
HorizontalActivity	Int
VerticalActivity	Int
RadiusChange	Int

5.4.2 PAD Variables

We show the pad variables although we do not use it

DancePadInfoFrame class

Name	Type
AvgChangeTime	Double
LastChange	Long

DancePadSensor class

Name	Type
Buttons	(L, LU, U, RU, R, RD, D, LD)
Changes	Int
AvgChangeTime	Double
MaxChanges	Int
PrevUpdateTime	Double
Update_Time_Threshold	Int
Steepof_Time_Threshols	Int
Active	Boolean

5.4.3 BeatPredictor

The same occurs with the BeatDetector variables, we do not use it.

Name	Type
BeatPredictor	BeatDetector
Clock	SystemClock

5.4.4 New Variables

Name	Type	Description
STATE	Double	Current state in the state machine

ANSWER	Boolean
	True=Yes;
	False=No;
		If the user answers us (YES or NO) our questions

CHORUS_DETECTED	Boolean	If the application detects that is arriving a chorus in the song, or the user is doing some kind of chorus.
RANDOM_ACTION	Boolean	Variable that is activated during the dance to indicate that would be interesting to do some new movement to enrich the dance.

USER_MOOD	Integer	Variable that show the mood state of the user
		0 = normal;
		1 = passive;
		2 = absent-minded;
		3 = exited;

FINISH_SONG	Boolean	Variable that tells us when the song is finished

Functions to fill the variables

```
• BOOLEAN FILL_ANSWER( )
  {
    IF increase the AvgChangeTime
    THEN ANSWER = TRUE
    ELSE ANSWER = FALSE
  }
```

- BOOLEAN FILL_CHORUS_DETECTED( )

```
{
   RESTART FILL_RANDOM_ACTION
}
```

- BOOLEAN FILL_RANDOM_ACTION ( )

```
{
   To wait some time and in a random moment
   THEN    RANDOM_ACTION = TRUE
}
```

- INT FILL_USER_MOOD ( )

```
{
   RESTART FILL_RANDOM_ACTION
   IF     AvgChangeTime << BeatPredictor
      AND       PercentajeLeftHandTop is very small
      AND       PercentajeRightHandTop is very small
   THEN         USER_MOOD = 1

   IF     AvgChangeTime >> BeatPredictor
   THEN    USER_MOOD = 3

   IF     AvgChangeTime <<>> BeatPredictor
   THEN    USER_MOOD = 2

   DEFAULT   USER_MOOD = 0
}
```

- BOOLEAN FILL_FINISH_SONG ( )

```
{
   IF
   BeatPredictor.GetTempo() is null
   THEN         FINISH_SONG = TRUE
}
```

5.5 Animations and Transitions

In this section we think about what animations we could use in each state, as well as the conditions that have to occur to go from one state to another one.

5.5.1 Animations

Bored

Animation	Description	Duration (s)	End Points
bored_take3.xml (animation 1)	bored 1	26,56	1
bored_take4.xml (2)	bored 2 (yawn)	44,64	8
rapper_19_4_pantomime.xml (3)	Touching an invisible glass	15,08	1
rapper_4_1_bored_watch.xml (4)	Warming	13,567	1

Goodbye

Animation	Description	Duration (s)	End Points
Goodbye.xml (animation 32)	To say goodbye with the hand	10	1
rapper_23_1_goodbye_tired.xml (33)	Bored, I do not want to dance more	10,2	1
“”_24_2_goodbye_wave_low.xml (34)	Hello / bye bye	11,44	1
rapper_25_1_goodbye_sad.xml (35)	Come on!!!!	8,76	1

Invite

Animation	Description	Duration (s)	End Points
invite_take5.xml (animation 36)	Saying hello to the people	35,24	8
invite_take6.xml (37)	Walking and saying hello	38,4	6
rapper_26_1_invite_letmehearyall.xml (38)	Come here, encouraging	3,52	1
“”_jump.xml (39)	I do not listen you, come on!!!	10,88	1
take10_invite_various.xml (40)	Showing a step	19,68	1

Dancing

Animation	Description	Duration (s)	End Points
rapper_11_2_yo_two_hands.xml (5)	Throwing the hands to the air	3,6	6
rapper_12_10_cross_hands.xml (6)	To cross the hands in the air	2	3
rapper_12_1_veegschouders.xml (7)	To quit your dandruff	6	12
rapper_12_4_handenindelucht.xml (8)	To put your hands in the air	6	8
rapper_12_5_handenbreed.xml (9)	To encourage	1,767	4
rapper_12_7_knielaag.xml (10)	To kneel down moving the hand	9	4
“”_12_8_laaghandenlucht.xml (11)	To surfer moving the arms	5	8
“”_12_9_draaihandenbreed.xml (12)	To do the plane	10	26
rapper_12_9_spin_around.xml (13)	To do the other plane	10	26
rapper_18_1_air_guitar.xml (14)	To play the guitar	13,28	27
rapper_18_2_scuba_fish.xml (15)	To up and down like scubbing	4,68	11
rapper_19_1_grease_point.xml (16)	Saturday night movement	3,88	6
“”_19_2_grease_point_alt.xml (17)	Grease movement	6,28	11
rapper_19_3_wax_on.xml (18)	To clean the window	4,64	11
“”_21_3_aggressive_kick.xml (19)	Kick boxing	8,68	18
rapper_21_4_punch_jump.xml (20)	Rocky	5,92	13
“”_2_1_stapheenenweer1b.xml (21)	left - right - left - right	5,634	12
rapper_2_2_stapheenenweer2b.xml (22)	left - right - left - right Moving the arms	6,434	12
rapper_2_3_stapheenenweer3b.xml (23)	left - right - left - right Most animated	4,833	10
rapper_2_6_zwaaihoog_annotated1.xml (24)	left - right - left - right Clapping the hands in the air	6,333	11
Rapper_2_7_stapvoor.xml (25)	front - back	7,5	17
Rapper_2_8_kniehoog.xml (26)	To kneel up	11,067	18
“”_2_9_zwaaiheenenweer1.xml (27)	Carton's movement	4,76	8
Rapper_3_1_basis_introvert.xml (28)	Shy dance	9,266	20
Rapper_4_2_cross_arms.xml (29)	Crossing the arms	6,1	13
Rapper_4_3_point_around.xml (30)	Throwing the hands - front	11,334	18
Rapper_5_1_stapuitklein1.xml (31)	front - right - front - right	3,9	7

5.5.2 State Transitions

In all these states, when we do an animation, we memorize it to avoid repeat it.

Waiting User (state 1)

Initial state

Animations: 1, 2, 3

- CV detection (I am going to be seen)

  IF PersonPresent = TRUE AND Buttons = Everything False

  THEN State = 2

- Pad & CV detection (I am going to say hello)

  IF PersonPresent = TRUE AND Buttons = Something TRUE

  THEN State = 3

To be seen (state 2)

We are going to detect if we attract the attention of the user

Animations: 36, 37

- No (The user has not seen me)

  IF PersonPresent = FALSE AND Buttons = Everything False

  THEN State = 1

- Yes (The user has seen us, and we are going to say hello him)

  IF PersonPresent = TRUE AND Buttons = Something TRUE

  THEN State = 3

To say hello (state 3)

We are going to say hello to the user.

Animations: 37 until first point

- (The application has said hello and we are going to invite him to dance)

  THEN State = 4

To invite to dance (state 4)

We are going to offer the user to dance with us.

Animations: 38 modified until first point

- User says Yes (The user has accepted to dance with us)

  IF ANSWER = TRUE THEN State = 6

- User says No (The user has not accepted to dance with us)

  IF ANSWER = FALSE THEN State = 5

To encourage the user (state 5)

We are going to continue trying to persuade the user to dance.

Animations: 39, 35

- User says Yes (The user has accepted to dance with us)

  IF ANSWER = TRUE THEN State = 6

- User says No (The user has not accepted to dance with us)

  IF ANSWER = FALSE THEN State = 5

To start the dance (state 6)

We are going to tell to the user that we start to dance.

Animations: 4, 5

- Start the song

  THEN State = 7

Dance time (State 7)

We are going to go calling all the functions to get an intelligent dance.

Animations: dance animations

- Chorus detected (Chorus or user choreography detected, to do a choreography)

  IF CHORUS_DETECTED = TRUE AND USER_MOOD = 0
  THEN State = 8

- Random (To do some movement in order to do a funnier dance)

  IF RANDOM_ACTION = TRUE AND USER_MOOD = 0
  THEN State = 9

- Detect passive movements (The user seems discouraged)

  IF USER_MOOD = 1 THEN State = 10

- Absent-minded user (The user seems absent-minded)

  IF USER_MOOD = 2 THEN State = 11

- Finish the song (The song has finished)

  IF FINISH_SONG = TRUE THEN State = 10

To do a choreography (State 8)

Animations: 11, 15, 16, 17, 30

THEN State = 7

To enrich the dance (State 9)

Animations: 6, 7, 10, 12, 13, 14, 18-29, 31

THEN State = 7

To encourage the user (State 10)

Animations: 8, 9

THEN State = 7

To be seen (State 11)

Animations: 39

THEN State = 7

To congratulate (State 12)

The song has finished, and we are going to congratulate the user.

- User go out (The user goes out after our congratulation, then we are going to say good bye)

  IF Buttons = Everything False
  THEN State = 15

- Random (The user is going to dance another time)

  IF Buttons = Something TRUE
  THEN State = 13 OR State = 14 (we do a random to choose one)

To encourage the user (State 13)

We are going to encourage the user to dance another song.

Animations: 39, 35

- (We have shown the animation, and we are going to start the dance)

  THEN State = 6

To explain some step (State 14)

We have explained some step to the user and now we are going to start another song.

Animations: 40

- (We have shown the demonstration, and we are going to start the dance)

  THEN State = 6

To say good bye (State 15)

The user does not want to dance more, and we are going to say him good bye.

Animations: 32, 33, 34

- (We have shown the animation, and we are going to return to the waiting state)

  THEN State = 1

5.6 Possible Improvements

5.6.1 Animations

- To say hello with the hand, looking to the center of the screen. (without dancing).
- To bow to the user (without dancing).
- To clap to the user (without dancing).
- To say hello putting the head up.
- To point at the ear to indicate that I do not ear or that I am going to say something.
- To point at the eye to indicate that I want to keep the attention of the user.

5.6.2 Outputs

- That our dancer could speak.

5.6.3 Inputs

- To recognize users' speech.
- To recognize head's movements.

5.6.4 Pattern Recognition

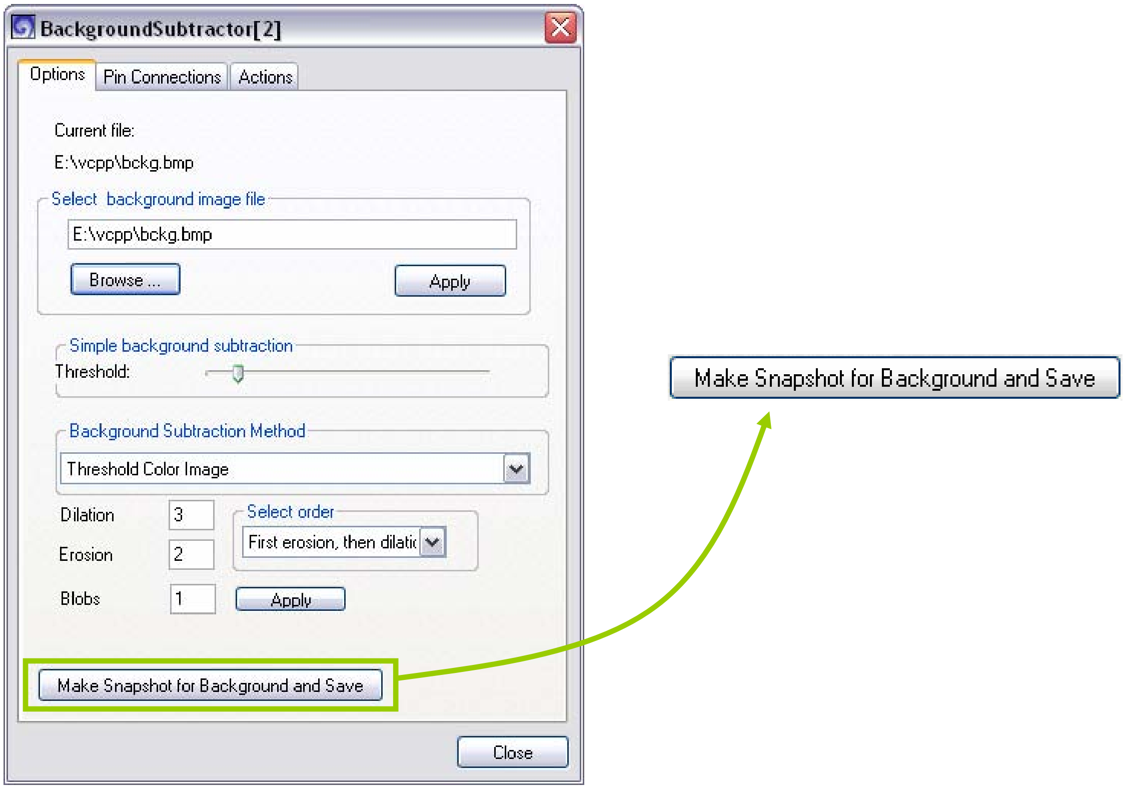
- Simple movements' recognition.
- Patterns recognition.
- Creation of patterns automatically.
- Search of Patterns.

Chapter 6. The Built System (CV)

In this chapter we are going to explain the modifications done in the computer vision system ParleVision. The new created and sent variables to the Artificial Intelligent part, the new features added to the application, and so on.

6.1 New Added Features

One of the first steps to use the CV application is to make a Snapshot. We need it in the BackgroundSubtractor Processor to do one of the most important actions in the CV, to recognize the silhouette of the user. The problem was that every time that we restarted the application we had to make a snapshot because it was not saved, it was stored in a dynamic memory.


BackgroundSubtractor[2]
Options
Pin Connections
Actions
Select background image file
Browse ...
Apply
Simple background subtraction
Background Subtraction Method
Select order
Apply
Make Snapshot for Background and Save
Close
Make Snapshot for Background and Save

Current file:
E:\vcpp\bckg.bmp

E:\vcpp\bckg.bmp

Threshold:

Threshold Color Image

Dilation 3
Erosion 2
Blobs 1

First erosion, then dilatic

Figure 16. BackgroundSubtractor Menu with the Background save button

The first thing that we did was to give the possibility to store the background in a not volatile memory. Now if you close the application the snapshot is not missed.

6.2 New Created Pipelines

In this chapter we are going to explain the different pipes used to do the application as the default pipe as the new created pipes.

6.2.1 Default Pipeline

To begin, we used the default pipeline, with the following Processors:

- CameraProducer (to take images with the camera)
- BackgroundSubtractor (to erase the background)
- ShadowDetector (to detect the users silhouette)
- Masks (to invert the image)
- CalculareCenterOfMass (to know where the user is, in the 2D axis)
- MaskFromCenter (to take a part of the image: topleft, topright and so on)
- EnterLeaveDetector (to detect the user's presence)
- RadialHistogram (to detect the radial activity)
- ServerSink (the server, by means of it the two applications can communicate between them)

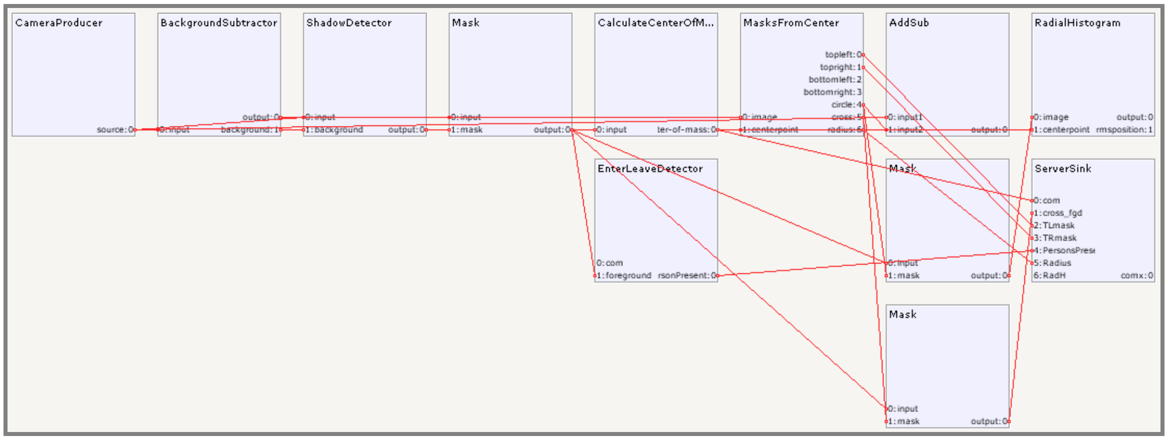


Figure 17. Default Pipeline

To start our work it was enough, but when we wanted to recognize complex patterns we needed a little more complex pipe. Moreover, we tried to do our work efficiently; then we created the save Video and the From Video Pipeline.

6.2.2 Save Video Pipeline

Every time that we wanted to test our progress, we had to put ourselves in front of the camera and dance. We thought that was better to record a video with a dance and to use it to test our application. To do it we created a new pipe that takes the images from the camera and saves it to the hard disk.

This is the pipe that we created to do this action:

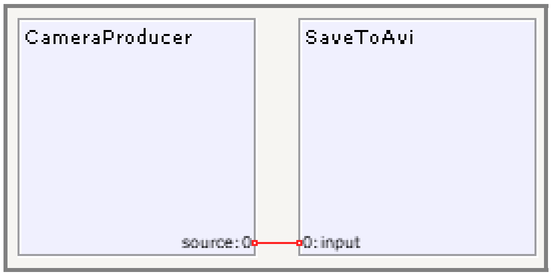
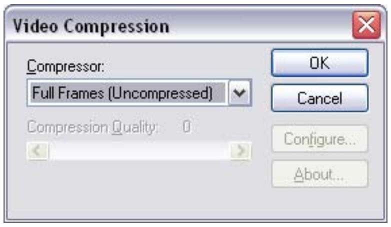


Figure 18. Save Video Pipeline

We can select the compression of the video. In our case we select all the frames, to prevent quality loss.

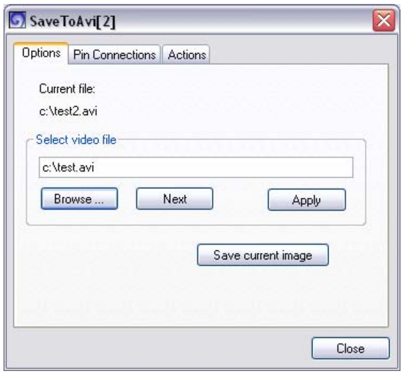


Compressor: Full Frames (Uncompressed)

Compression Quality: 0

Figure 19. Compression of the Video Menu

And the route where we want to save the video:

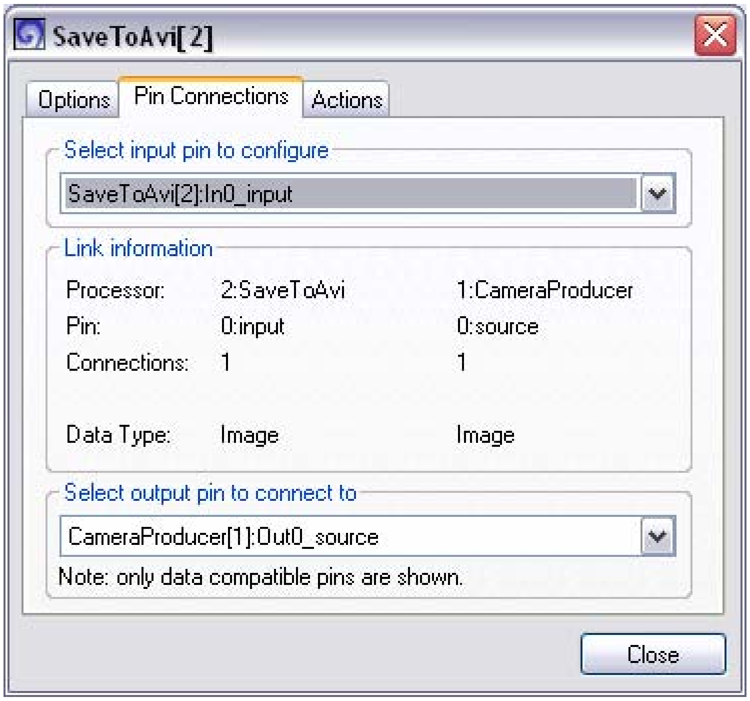


Current file:
c:\test2.avi

Select video file
c:\test.avi

Figure 20. Video Save Route

This pipe is a good example to see how the connections between processors have to be done.



Select input pin to configure

SaveToAvi[2]:In0_input

Link information

Processor:	2:SaveToAvi	1:CameraProducer
Pin:	0:input	0:source
Connections:	1	1
Data Type:	Image	Image

Select output pin to connect to

CameraProducer[1]:Out0_source

Note: only data compatible pins are shown.

Figure 21. Pin Connections Menu

In the figure 18 we can see the two processors (SaveToAvi and CameraProducer) and how they are connected.

The CameraProducer out 0 pin is connected to the SaveToAvi input 0 pin. Both input and output pins have to have the same type.

6.2.3 From Video Pipeline

To be able to use the recorded video we had to make a new pipeline where the first process was the VideoProducer. This process allowed us to use a video instead of the camera.

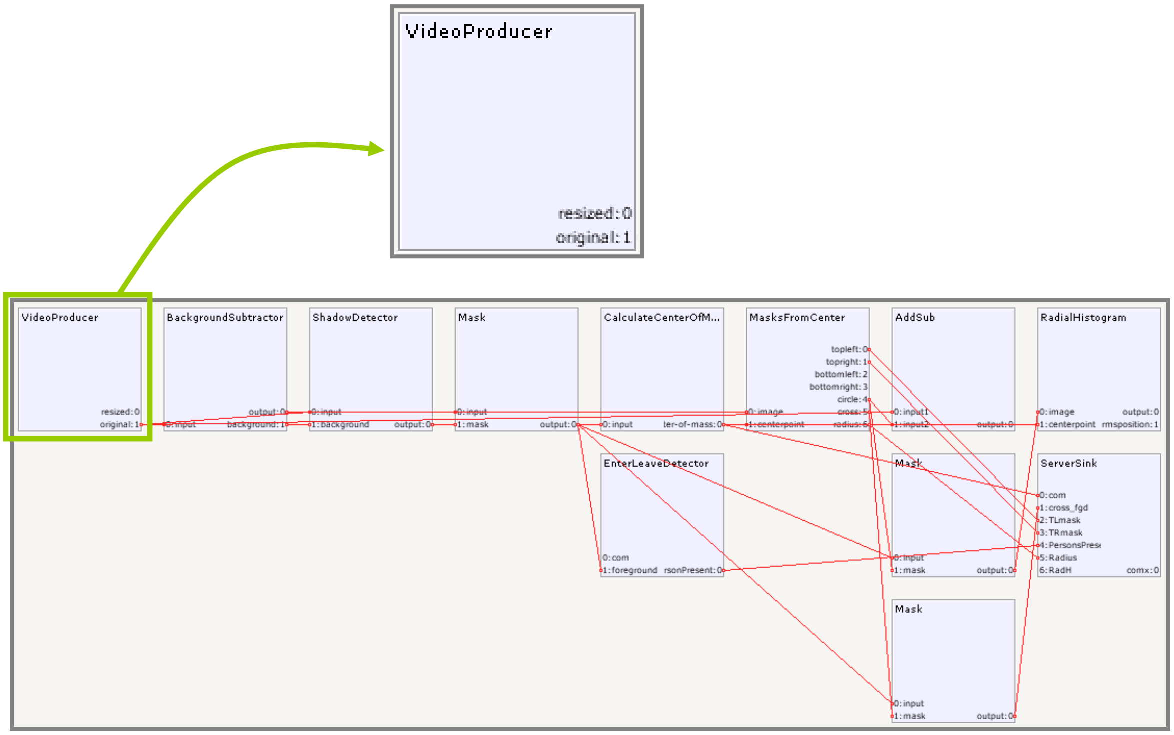
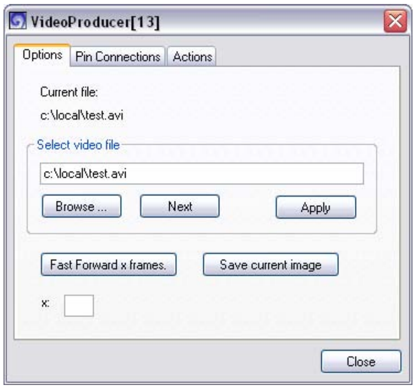


Figure 22. From Video Pipeline

We can select the route where the video is, using the select video menu:



Current file:
c:\local\test.avi

c:\local\test.avi

x:

Figure 23. Select Video Menu

And the other processes are the same than the default pipeline.

6.2.4 Pipeline with Radial Histogram

Finally we created a pipeline that could send the radial histogram. We had to create a new input in the ServerSink processor to allow the send because in that moment we did not receive it in the ServerSink. When we created the new input we only had to connect it with the RadialHistogram output.

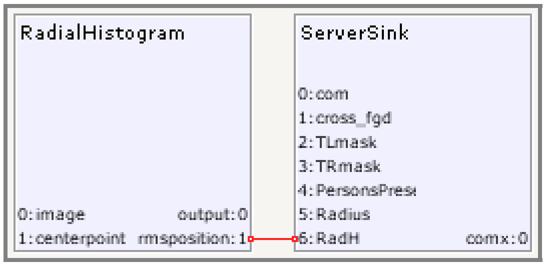


Figure 24. Connection between Server Sink and Radial Histogram

On the Figure 25 is possible to see the last created pipeline; the pipeline that we have used to test our application. To see the new application's improvements is necessary to use this pipeline.

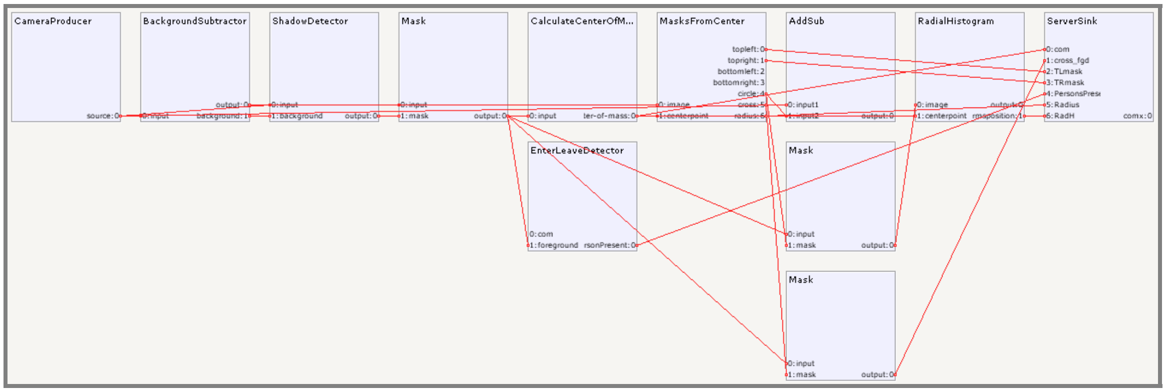


Figure 25. Pipeline with Radial Histogram

6.3 New Sent Variables

In this chapter we show the new variables sent to the recognition application.

cumulativeHorActivityNormalized

We send the variation between the actual frame and the previous frame in order to control if the user is moving to the left or to the right.

cumulativeVertActivityNormalized

The same occurs with the vertical movement. We use the vertical variation to know if the user is jumping, ducking or rising.

lHand

In this variable we have the most active left quadrant.

rHand

The same that the lHand variable but in this case with the right quadrant.

We have the Radial Histogram divided in 16 parts. With green circles you can see the interesting parts for us. We recognize 6 different hands' positions: 7, 8 and 9-10 for left hand and 13, 14-15 and 0 for the right hand. From 1 to 6 are the head and 11 and 12 are the body.

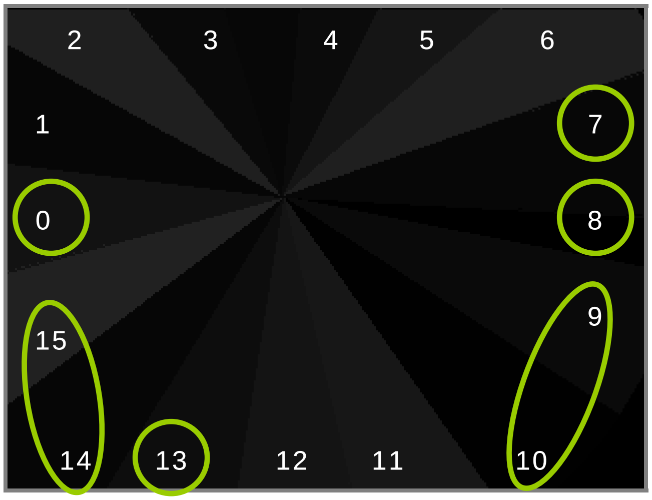


Figure 26. Pipeline with Radial Histogram

cumulativeRActivityNormalized

In this variable we have the radius variation. We use it to recognize the twister.

r

We use the “r” variable to calculate the horizontal and vertical threshold. With the threshold we omit the little movements.

x:

To know the actual horizontal position. We use it to do previous calculations.

y

To know the actual vertical position. We use it to do previous calculations and to calculate the vertical movement.

To see the exactly use of these variables look at the Dance Movement Patterns Recognition (Part II)

Chapter 7. Conclusions

7.1 Reached Goals

Our principal goal was to recognize patterns. We started recognizing simple movements and after it we achieved to recognize groups of simple movements (patterns). We also wanted to create an automatic pattern generator and we achieved our purpose.

7.2 Not Reached Goals

The last goal that we tried to reach was to find patterns into a group of dance movements without a reference. The idea is to find patterns but without knowing the patterns that we want to find. We were searching information about how we could implement it, but we only found searchers with something known.

The problem is that we do not know what we are searching, we do not know if there are patterns to be found, we do not know the length of the possible existing pattern, and so on.

7.3 Found Problems

The first problem that we have found has been that the computer vision application has great environment dependence. Depending on the light, the background and so on the output results change.

This is so bad for us because our application depends on the computer vision variables. A little change in the light could vary the generated results by the computer vision and could force the application to fail the recognition.

Another problem is the mentioned in the chapter 7.2. How we could search something if we do not know what we are searching. It is a deep and interesting question, I think.

We have had problems to decide when a step finishes and starts the next one. And the same problem occurs with the patterns; when does a pattern finish? Are there patterns inside other patterns? Is better to recognize small patterns or big patterns? and so on.

7.4 Possible Improvements

- The goal not reached.
- To improve the computer vision.
- To recognize new movements.
- More complex patterns.
- More than one user dancing capable.

7.5 Personal Opinion

Firstly I want to say that I am very pleased to have worked in this project.

To work in group has been a good experience, sometimes easier than alone sometimes the other way around. But globally it has been a rich experience.

The application results have been quite good but could be improved. This is normal because we only have been working for five months, and I think that with more time the results could be better. For me it is the first time that I have worked in this theme and I am really fascinated.

I think that the interaction between people and machines is an interesting topic to be studied deeply.

Chapter 8. References

8.1 Books

1. Dennis Reidsma, Herwin van Welbergen, Ronald Poppe, Pieter Bos and Anton Nijholt : Towards Bi-directional Dancing Interaction. (2006)
2. Jurgen Braam : ParleVision 4.0 a framework for development of vision software. (2004)
3. Radford M. Neal : Probabilistic Inference Using Markov Chain Monte Carlo Methods. (1993)
4. Saeed V. Vaseghi : Advanced Digital Signal Processing and Noise Reduction. (2000)
5. A. S. Poznyak, K. Najim and E. Gómez-Ramírez : Self-Learning control of Finite Markov Chains. (2000)
6. Wilhelm Huisinga and Eike Meerbach : Markov Chains for Everybody. (2005)
7. Troy L. McDaniel : Java HMMPak v1.2 User Manual. (1996)
8. Stuart J. Russell and Peter Norvig : Artificial Intelligence a Modern Approach. (1995)

8.2 Articles

9. Jean-Marc François : Jahmm v.0.6.1. (2006)
10. Atsushi Nakazawa, Shinichiro Nakaoka, Katsushi Ikeuchi, Kazuhito Yokoi : Imitating Human Dance Motions through Motion Structure Analysis. (2002)
11. Roberto A. Pava : Búsqueda de repeticiones en secuencias de ADN. (2006)
12. Richard D. Green and Ling Guan : Quantifying and Recognizing Human Movement Patterns From Monocular Video Images - Part I: A New Framework for Modeling Human Motion. (2004)
13. J. P. Lewis : Fast Normalized Cross-Correlation. (1995)
14. Lucian Vintan, Arpad Gellert, Jan Petzold and Theo Ungerer : Person Movement Prediction Using Neural Networks.
15. Susan Loman : Employing a Developmental Model of Movement Patterns in Dance/ Movement Therapy with Young Children and Their Families. (1998)

8.3 URLs

16. http://hmi.ewi.utwente.nl/showcases/The%20Virtual%20Dancer

INTERACTING WITH A VIRTUAL DANCER

17. http://hmi.ewi.utwente.nl/showcases/parlevision

PARLEVISION: RAPID DEVELOPMENT OF COMPUTER VISION APPLICATIONS

18. http://www.public.asu.edu/~tmcdani/hmm.htm

Hidden Markov Models

19. http://www.run.montefiore.ulg.ac.be/~francois/software/jahmm/example/#datafiles

Jahmm - An HMM library

20. http://www.puk.unibe.ch/aa/sozphynv.html

HEAD-MOVEMENT PATTERNS OF SCHIZOPHRENIC PATIENTS AND NORMAL CONTROLS

21. http://www.puk.unibe.ch/aa/sozphynvana.html

Recording and Quantification of Head-Movement Patterns

22. http://www.sasked.gov.sk.ca/docs/physed/physed1-5/ep_perspective2.html

Movement Perspective

University of Twente
Enschede - The Netherlands

Universitat
Autònoma
de Barcelona

Project carried out by Israel Solà Cerdán

Student of Escola Tècnica Superior d'Enginyeria at the Universitat Autònoma de Barcelona,

Signed: ....

Enschede, 09 of February of 2007

Resum

La proposta d'aquest projecte sorgeix per la necessitat d'obtenir una major interacció entre persones i maquines. En el nostre cas, entre un usuari i una aplicació que reprodueix una ballarina.

Per realitzar el següent projecte s'ha utilitzat l'aplicació "The Virtual Dancer", que te com a objectiu principal ballar amb l'usuari. L'aplicació capta els moviments de l'usuari (mitjançant visió per computador, implementada amb C++), els processa i actua segons ells (implementat amb Java).

Per obtenir una major interacció s'ha optat per fer que l'aplicació, no nomes capti els moviments, sinó que sàpiga que moviments esta captant per a que d'aquesta manera d'interacció sigui mes "rica".

Una vegada es sap quins moviments fa l'usuari, el següent pas es reconèixer conjunts de moviments (coreografies), es a dir, patrons de moviments de ball.

Reconeguem-te patrons de moviment, l'aplicació pot avançar-se a l'usuari en la següent tornada, introduint nous passos, variant la coreografia, animant l'usuari, etc. Però d'una manera interactiva, d'una manera que fa que l'usuari no percebi talls, ni moviments bruscos i es senti còmode.

Resumen

La propuesta de este proyecto surge a raíz de la necesidad de obtener una mayor interacción entre personas y maquinas. En nuestro caso, entre un usuario y una aplicación que reproduce una bailarina.

Para realizar dicho proyecto contamos con la aplicación “The Virtual Dancer”, cuyo objetivo principal es el de bailar con el usuario. La aplicación capta los movimientos del usuario (mediante visión por computador, implementada con C++), los procesa y actúa según ellos (implementado con Java).

Para una mayor interacción se ha optado por hacer que la aplicación, no solo capte los movimientos, sino que sepa que movimientos esta captando para que de esta manera, la interacción sea más “rica”.

Una vez se sabe que movimientos hace el usuario, el siguiente paso es reconocer conjuntos de movimientos (coreografías), es decir, patrones de movimientos de baile.

Reconociendo patrones de movimiento, la aplicación puede adelantarse al usuario en el siguiente estribillo, introduciendo un nuevo paso, variando la coreografía, animando al usuario, etc. pero de una manera interactiva, de una manera que hace que el usuario no perciba cortes, ni movimientos bruscos y se sienta cómodo.

Summary

The proposal of this project arises as a result of the necessity to obtain a greater interaction between people and machines. In our case, between the user and the application that reproduces a dancer.

In order to perform this project we work with the application “The Virtual Dancer”, of which main goal is to dance with a human.

The application picks the movements of the user up (by computer vision, implemented with C++), it processes them and it acts according to them (implemented with Java).

For a greater interaction the application not only catches the movements. Now the application knows what movements it is receiving. In this way the interaction is “richer”.

Once the movements made by the user are known, the following step is to recognize groups of movements (choreographies), in other words, patterns of dance movements.

While recognizing movement patterns, the application analyses the next chorus, introducing a new step to the user, changing the choreography, encouraging the user, and so on. It does it in an interactive way; a way that makes the user feels more comfortable, without cuts and abrupt movements of the virtual dancer.