2011

V-Pollock

Summer School of Image Processing
2011 - Szeged

Bogdan PETROVAN – Technical University of Cluj-Napoca, Romania
Teodora SZASZ – Technical University of Cluj-Napoca, Romania
Gabor KATAI – University of Szeged, Hungary
Marco PEREANEZ – University Pompeu Fabra, Spain

7/14/2011

Table of Contents

Goals

By using a Microsoft Kinect sensor, we propose to create a painting application for a user to paint. This is Team A's project at Summer School of Image Processing 2011 (Szeged) 19th edition.

User will interact with his finger, pointing the line to be drawn and the thickness of the brush, in what we called "volumetric pad". By moving his finger closer to the sensor, he will increase the brush thickness.

Some key features planned for the application:

- user can define the virtual volumetric pad
- user will select either solid color or textured brush from predefined palettes
- undo functionality
- save the painting as an image file
- except painting, the user will interact with the application by voice commands and gestures
- short help manual available at application runtime

[check with marco's site]

Kinect Sensor Technical Specifications & API

Microsoft Kinect sensor outputs three different kinds of streams:

- image streams: color images, depth images and player segmentation images;
- audio stream: from the microphone array on the device
- skeleton: computed skeleton data for each player

Image Streams

The information from these streams can be retrieved by pooling mechanism or event based.

Color Images

This kind of data is offered at two-quality level and at two different formats:

- RGB – 32 bpp X8R8G8B8
- YUV – 16bpp UYVY

Color data is not used in this project, for more information see [1].

Depth Images

Frame sizes of this data are:

- Frame size 640x480
- Frame size 320x240
- Frame size 80x60

Pixel format is 16 bpp, low-order 12 bits represent the distance from the sensor to the object and the other 4 bits are not used. Special meaning is the 0 value, that means that no depth data could be detected, the object being either too close to the sensor or too far from it.

4 bits - Unused	12bits – Depth information

Player Segmentation

Even though it is other type of data and is separately processed, this data is sent by the device along with depth data, forming a depth and player segmentation stream. Pixel information packing is almost the same, lower 3 bits are used for player index and the remaining 13 bits are used for depth data.

13 bits – Depth information	3 bits – Player index

Skeleton Stream

Offers computed skeleton data for each player, either by tracking the depth field or by inferring from past frames. A skeleton is structured in multiple joints, for each joint 3D coordinates are given in meters. Coordinates reference is the Kinect sensor.

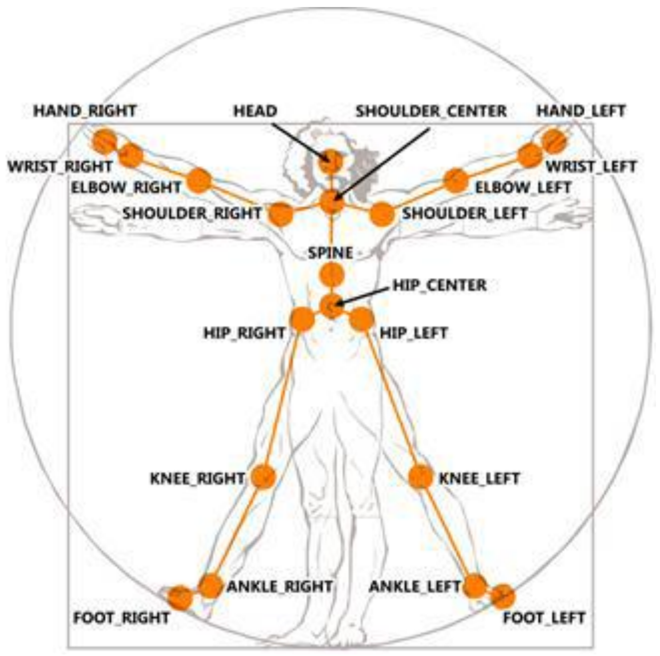
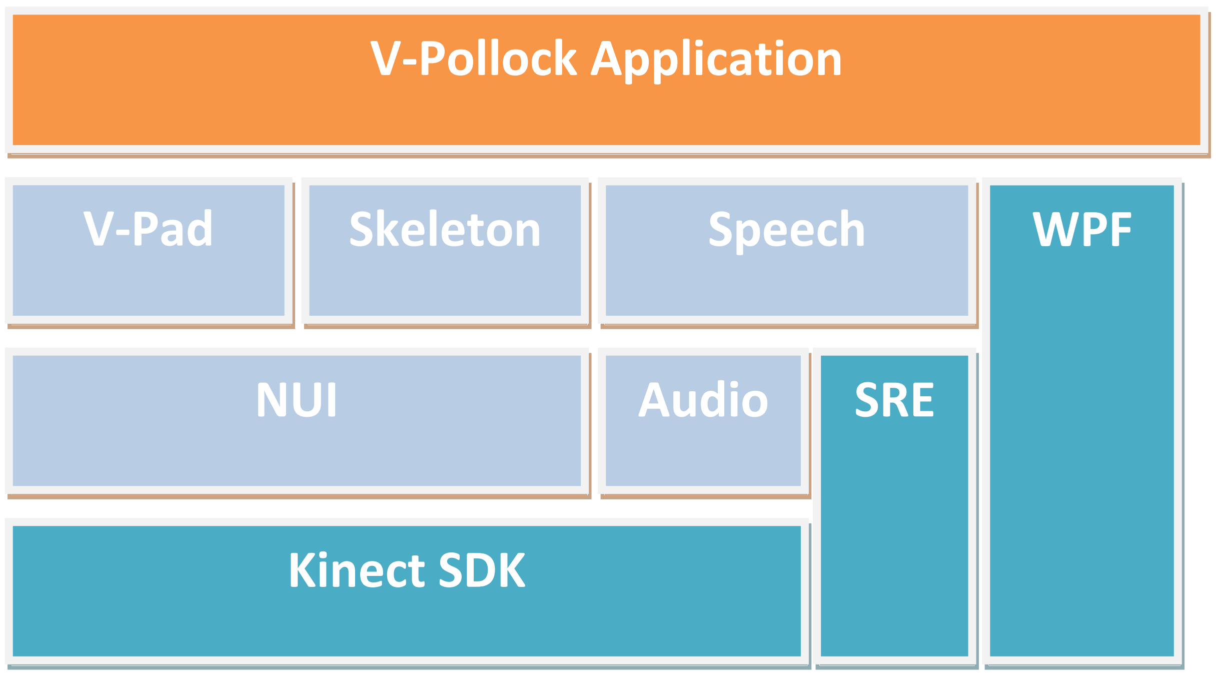


Figure 1 Skeleton Joints

Project Development

At first, we defined the application design and major tasks for implementation. Also after having the tasks defined, we identified main risks for our project success.

Design



Volumetric pad (v-pad)

Is built on top of the NUI module of the Kinect API and by using the depth frame information tracks the user’s finger in an event driven model.

Skeleton

This module is built on top of the NUI module of the Kinect SDK and it detects user gestures, offering an event driven interaction with the main application.

Speech

This module is integrating the Audio module from Kinect SDK with the Microsoft Speech Recognition Engine (SRE), offering an event based interaction with the main application.

Windows Presentation Foundation(WPF) is the API used for creating the graphical user interface (GUI).

APIs used

Kinect SDK

It offers means for the programmer to access easily the data sent by the Microsoft Kinect Sensor. The SDK is still in Beta version from Microsoft Research, helping researchers and technology enthusiast to develop easily rich experience applications using Kinect for Xbox 360.

SDK requirements:

- OS: Windows 7 (not virtualized)
- Windows 7- compatible graphics card that supports Microsoft DirectX 9.0c capabilities
- 2 GB RAM
- Kinect for Xbox 360 sensor

Image streams can be accessed by an event driven model, packing the information in frame objects.

Other

Technologies and software used:

- Microsoft .NET Framework 4
- Microsoft Kinect SDK (Beta version)
- Microsoft Speech Recognition Engine

Windows Presentation Foundation WPF

Application Development

Finger Detection

In our application, only one player observed, so we do not use the payer number. To convert the dept data to 16 bit unsigned integer (UINT16) we shift the bits of the first byte (byte 0) right three positions and the second byte (byte 1) left five positions. After that we perform a bitwise OR operation on the shifted data. Te result is presented in fig. 2, having the depth information scaled to 8-bit grayscale.

The user can draw with our application by using his stretched finger. The main idea is to detect the closest point from the camera, which is the position of the finger.

After we converted all of the pixels, we determine the minimum of the dept image. First we replace the invalid dept values (0) by the maximum value (2^{13}). We use a lower and a higher threshold level to filter data, which are not in the working area. The minimum searching implemented in one loop. After, we determine the square, which contains all of the minimal pixels. The center of the square represents the position of the finger.

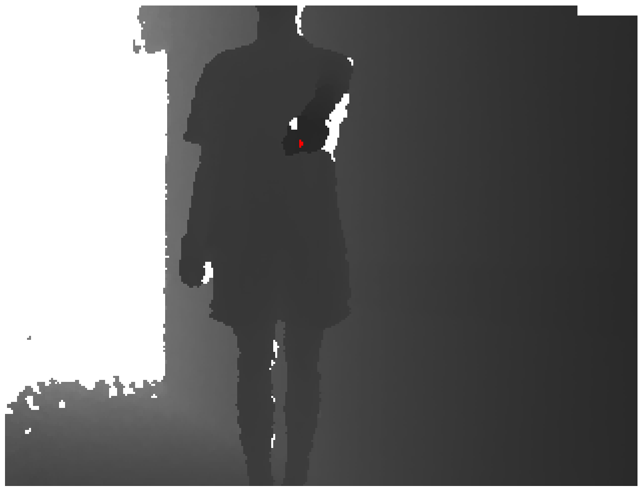


Figure 2 Depth frame and finger detection

Daily Tasks

Friday (07/08/2011)

Team Member	Tasks
Bogdan, Dora, Gabor, Marco	• Development & Features Planning • Sensor testing • Research for Kinect sensor and speech recognition

Monday (07/11/2011)

Team Member	Tasks
Bogdan	- find a way for save/load depth images to be used for testing initial finger detection algorithm - continue on building GUI and prepare modules for Speech, Drawing and Skeleton gesture detection - offer support for C#/.NET development to Gabor and/or Dora - implement skeleton data comparison with model - (maybe refactor the current Main Application)
Dora	- use the Speech Recognition sample from Microsoft website - try to modify in order to use microphone source instead of Kinect mikes

	- build a small application that will trigger actions based on Speech Recognition (get help from Bogdan on this one)
Gabor	- analyze depth data from the senzor - try to detect the finger's position, by thresholding the depth field and finding some center (gravitational center I believe it should be the easiest) - test it on the real device, ask Bogdan for help for saving the test results in a file such that we can analyze that later (ideea is to implement a FIR/IIR filter on the coordinates obtained such that the detected movement is continuous)
Marco	- Documentation: o Introduction (also see if there are similar works in this area) o Technical description of the sensor o Application architecture (ask Bogdan for that) - Web Site o Template o Page Structure (something like a sitemap with some minor explanation) - think of some funny way to make a team photo, or individual photos, to be used in the presentation

Tuesday(07/12/2011)

Team Member	Tasks
Bogdan	• contibute to the website template!!! • Take some skeleton models and test the real-time comparison • contribute to implementation of depth field finger detection • (maybe refactor the current Main Application)
Dora	• extend your application such that it allows menu browsing • integrate into existing application
Gabor	• test it on the real device • prepare some code to store a sequence of detected coordinates in order to analyze them and maybe to test different filtering techniques
Marco	• Documentation: • Introduction (also see if there are similar works in this area) • Technical description of the sensor • Application architecture (ask Bogdan for that) • Web Site • Template • Page Structure (something like a sitemap with some minor explanation) • think of some funny way to make a team photo, or individual photos, to be used in the presentation

Wednesday (07/13/2011)

Team Member	Tasks
Bogdan	• Speech recognition integration + menu UI • texture brush menu • Skeleton action detection + Take some skeleton models and test the real-time comparison • movement/angle limitiation (or other methods) for removing noise on the drawing path
Dora	• Speech recognition integration • short movie for presenting application features
Gabor	• movement/angle limitiation (or other methods) for removing noise on the drawing path (if needed) • testing
Marco	• Documentation: • Introduction (also see if there are similar works in this area) • Technical description of the sensor • Application architecture (ask Bogdan for that) • Menu Icons (gesture & action icon) • short movie for presenting application features

Users Manual

First Use

Interaction with the application will be done in the space in front of the sensor, having the user moving his finger.

Before using this functionality, the user must calibrate the system by defining his drawing canvas. The calibration process is started by saying 'calibrate' or by entering in the menu (pressing [A] of by saying 'menu') and the selecting Calibrate (clicking or saying 'calibrate'). After the user defines the top-leftmost corner and the right-bottom corner, he will say 'done' and the system will have the drawing canvas calibrated.

The user can start painting by moving his finger on the specified canvas. One key feature is that the user will be modifying the brush thickness by moving his finger closer to the sensor and stopping the drawing by moving his finger behind the place where he defined a canvas.

Functions

Menu – it shows the list of functions aside the painting functionality. It can be accessed by saying 'Menu' or by pressing key 'A'

Save saves the current painting as a file in the application folder. It can be accessed by pressing key [V], by saying 'Save', by accessing the menu and selecting the Save option or by simply spreading the arms at level of shoulders.

Color palette show the color palette and select a solid color for painting either by clicking on the color or by saying the color's name that is written under the color preview.

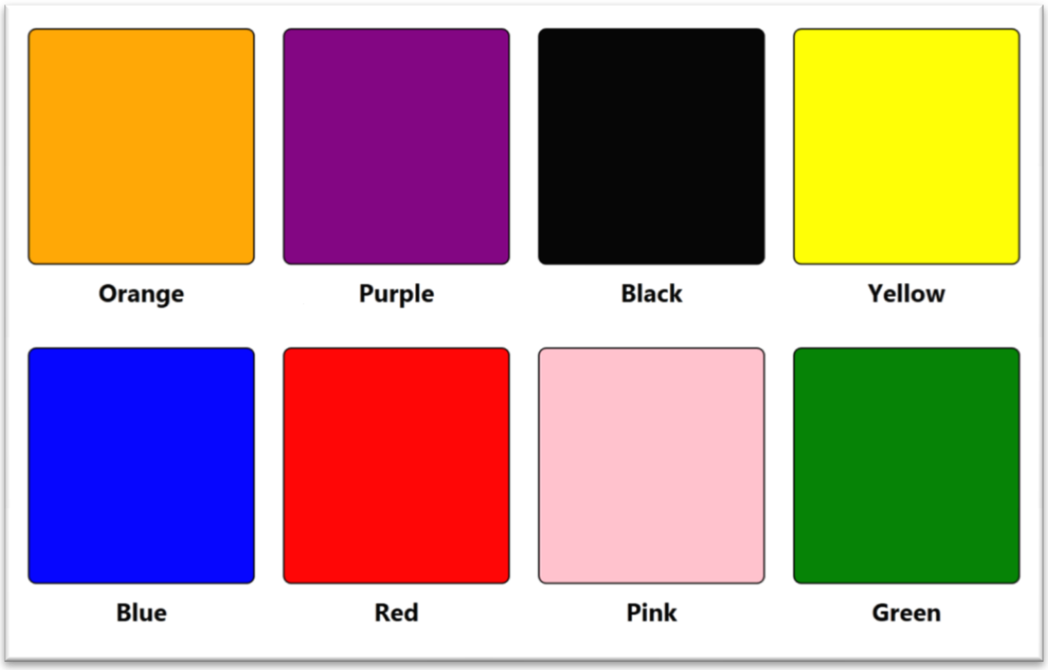


Figure 3 Color Palette

Texture palette shows the textures palette. The selection is the same as in the color palette case.

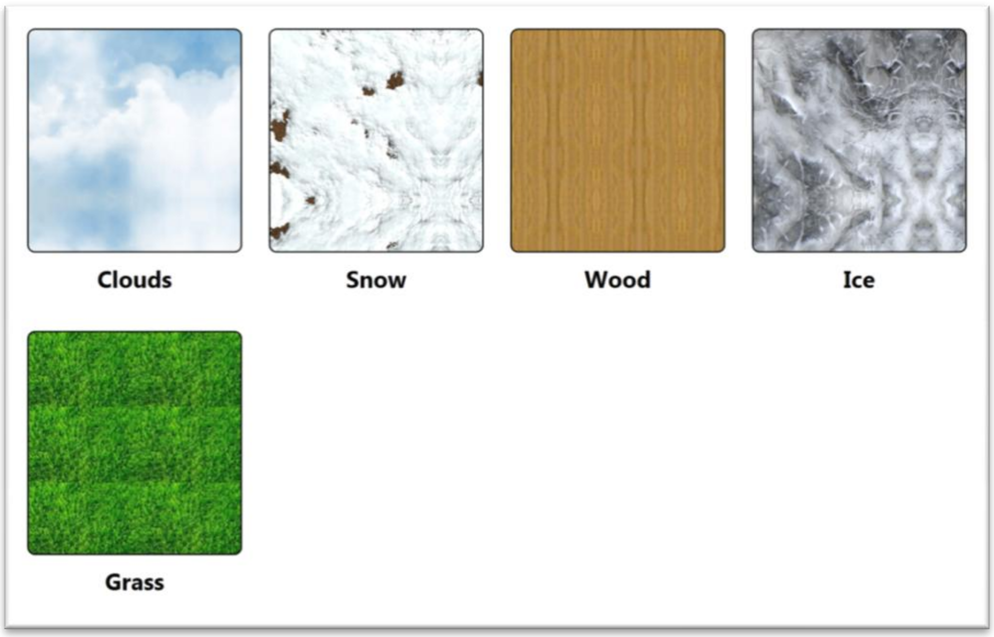
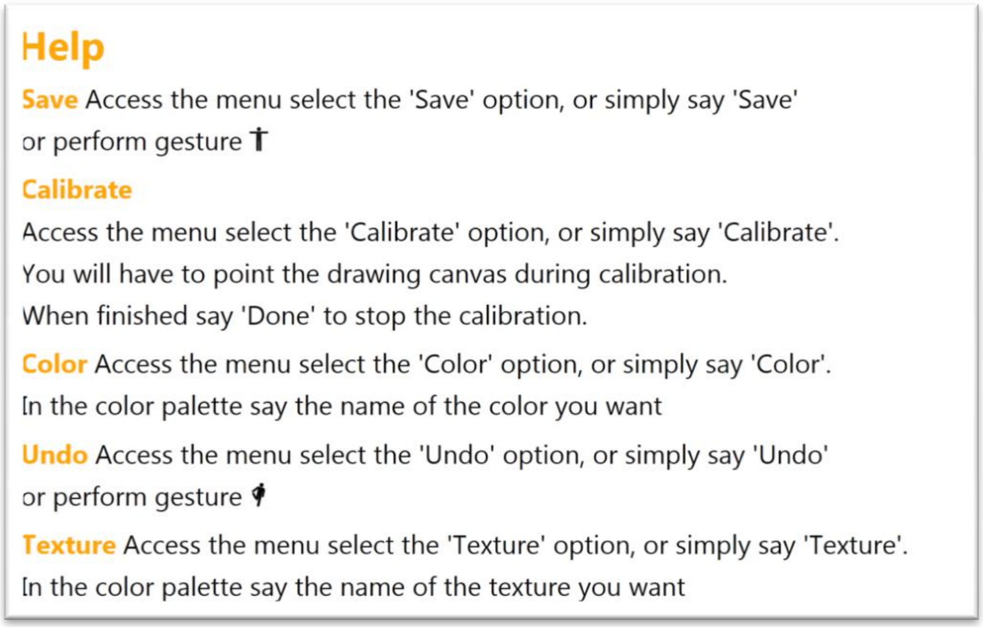


Figure 4 Textures Palette

Undo removes the last drawn figure. It can be accessed by saying ‘undo’, by pressing ‘U’, by accessing the menu and selecting Undo option or by grabbing his ankle.

Help shows basic voice and gesture commands; can be accessed by saying ‘help’ or by accessing the menu and selecting the ‘help’ option.



Help

Save Access the menu select the 'Save' option, or simply say 'Save' or perform gesture

Calibrate

Access the menu select the 'Calibrate' option, or simply say 'Calibrate'.
You will have to point the drawing canvas during calibration.
When finished say 'Done' to stop the calibration.

Color Access the menu select the 'Color' option, or simply say 'Color'.
In the color palette say the name of the color you want

Undo Access the menu select the 'Undo' option, or simply say 'Undo' or perform gesture

Texture Access the menu select the 'Texture' option, or simply say 'Texture'.
In the color palette say the name of the texture you want

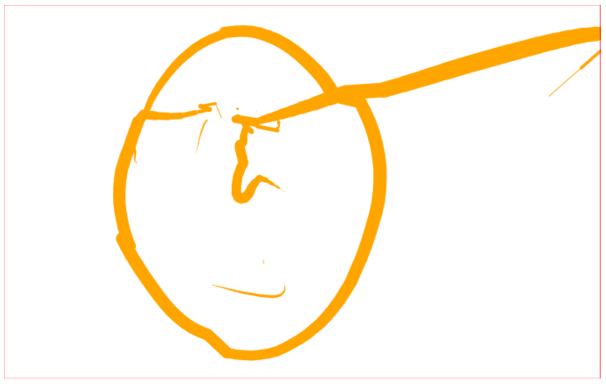
Figure 5 Help

Results

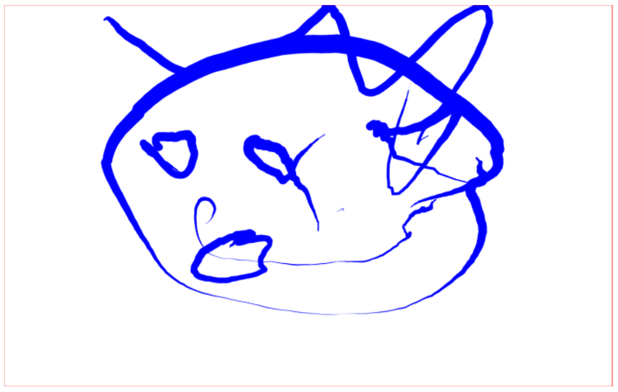
Self-portraits ☺

Bogdan

Gabor



Marco



Dora

Conclusions

The main feature of finger painting was implemented and working in real-time application. Also extra features such as:

- brush thickness control by using finger to sensor distance
- voice commands for the all actions
- gesture commands for save, undo and help

Besides working in an international team of students and project tasks analysis, we also developed other **personal achievements**:

- Bogdan – for the first time taking project and team management tasks;
- Dora – use for the first time C# language and Speech Recognition Engine;
- Gabor – use for the first time C# language and depth information;
- Marco – created for the first time a website;

References

[1] http://research.microsoft.com/en-us/um/redmond/projects/kinectsdk/default.aspx

[2] J. Shotton, "Real-Time Human Pose Recognition in Parts from Single Depth Images", Microsoft Research