

Progress Report II

I. Pandaboard Configuration

Since the previous report, we have had to swap the Xbox Kinect for an ASUS Xtion. The reason for this was due to incompatibilities with the available drivers and software. The NITE software, which is used to detect and find the parameters of people, is closed source and no version was available for the ARM architecture which was also compatible with the Kinect drivers.

In addition to the sensor change, we also had to change which operating system we were using in order to use the NITE binaries we managed to locate online. The binaries are not compatible with Ubuntu 12.04, which we initially were trying to use, so we had to change to version 11.10. This was not entirely trivial as there are some bugs in the version 11.10 installer, and a workaround had to be discovered in order to get this to work.

With these changes we have been successful in getting the NITE software to work with the newly purchased ASUS Xtion. There were three pre-compiled binaries we located online, one is the OpenNI, another for NITE, and a third for the sensor driver. We found a website where these were provided for us, making the task of getting the sensor working much easier. One problem we have begun to encounter however is the software is not entirely stable, and occasionally will result in a 'segfault' error, which may complicate any design day demonstrations.

Lastly, a wireless access point and SSH has been set up on the Pandaboard. This was done using a tutorial found online where someone turned their Pandaboard into a wifi router. Open source software called 'hostapd' was used, along with a DHCP server package. Accomplishing this is a huge milestone in making our product actually useful. We no longer will need to connect a monitor or keyboard to the Pandaboard every time we need to use it. All we need to do now is boot the board, and wait for the SSID "AirForce1" to appear on our laptops wireless network list, connect, and run an SSH client to control the board.

II. Software

The Pandaboard software is mostly completed. It may not be as useful or as accurate as we would like, however. It appears that the software sometimes returns values that do not make sense for the position of our users. The software is supposed to return real world coordinates of various parts of a target's skeleton, such as their head, neck, shoulders, elbows, hands, etc. This is definitely a problem and may prevent us from realizing true detection

software. This is not really something that we can do anything about, however, except for to wait for the next release of a more accurate and stable version of the software package we are using. Only one package exists for us to use at this time, and it is not something we could develop on our own time without several years and a team of PhDs.

As mentioned in the previous section, the software, in addition to not being completely accurate, is also unstable. This results in our program randomly not started, or randomly crashing for no real reason. The bugs are in the closed source software we are using, and though we may be able to work to limit the occurrence of crashes, there is no way we can eliminate them completely. Since this is only a prototype it is not a huge problem, with the exception that we may have some issues with design day demonstrations.

Another problem we have is the AFRL software. It is not complete at this time. Our sponsor is aware of this, since they wrote it, and no longer expect us to have a complete human scattering simulator functioning. They have made their requirements for our success very minimal -- they only request that we have a portable box for them, and that our software is able to communicate with their software. We have only recently received their software, and are still working on our second objective of getting it to talk to the PandaBoard, though it is not expected to be a difficult task. Since the AFRL software is not complete, we are currently also developing a separate client to potentially be used on design day, in order to positively show off our project.

Software has been written to that will be able to gather the parameters that the PandaBoard sends. It will then write these parameters to a file. As per the AFRL's request, we will then modify their code slightly to read in this file. This will demonstrate to our sponsors we have completed their task.

III. Battery and Enclosure

The battery has been successfully tested and confirmed to work with the PandaBoard. We ended up deciding to go with a lithium based charger pack as this eliminated the cost and complexity associated with designing our own charger circuit. The battery is outputting a regulated 5.3V and PandaBoard appears to be accepting this value without any instability. The end goal is to allow the user to manually select which source they would like to power the enclosure with. The battery will be used for short, portable sessions and a wall adapter will be used for more extended sessions. The two sources will be selected through a toggle switch.

We decided to order a prefabricated plastic case and mod it to suit our needs. We found that we had to cut out a few pieces of the case on the inside to make room for the PandaBoard. The stand of the Xtion was bolted to the top of the case, and a power switch, toggle switch, and DC input connector were attached to the side. We plan to wire these switches and have enclosure completed by next week. The battery sits below the PandaBoard in the case, and

everything fits together nice and snugly. There are no apparent problems with overheating on any of the components.