Abstract

As the baby boomer generation retires and faces the challenges of old age, the young face the challenges of caring for the elderly. With a larger population of elderly than ever before, utilizing modern technology to increase efficiency and safety of patient monitoring will be more relevant in the coming decade than in any previous time. To help with this matter, we present a gesture recognition system intended for use in home monitoring of the sick or elderly. This system will detect when the monitored user is injured or in pain and will automatically notify an emergency care staff.

Introduction

A large portion of the elderly (as well as those with physical or mental conditions) are able-bodied enough to live on their own, but have a condition that could put them at risk of serious harm. This condition, when triggered, could leave the victim unable to perform the basic actions required to contact help. People in this situation are left with only a few options. They may move into a patient care facility, hire a live-in nurse, or install surveillance cameras inside their home to allow medical staff to monitor them. All three of these options remove the independence and privacy of the patient.
A fourth option is available in the form of a small worn device that can be activated by the user in an emergency to contact an ambulance. While this may cover some cases, the downside of this option is that it requires the patient to manually press the button. In many emergencies, the patient may not think to press the button, or may be unconscious. Additionally, the responding personnel have no information as to the type of emergency, and thus cannot prepare in the ambulance as they drive to the victim’s location.

To address these issues, we introduce a Kinect-based natural gesture recognition system. This system, when placed in a user’s home, will allow constant, automatic monitoring of the patient. Upon sign of distress, the system will send a message to an emergency care staff that will display the location and severity of the pain. Since the monitoring system is automated, the privacy of the user is not violated, and their autonomy is not infringed.

**Motivation of Natural Gestures**

Gesture recognition has been more popular lately, and has been integrated in consumer devices for the purpose of entertainment. One such device is the Microsoft Kinect, which uses gestures to allow a user to control games and perform other tasks such as starting a movie. These gestures are typically intuitive and relatively simple, but still must be learned and practiced before the user will be completely proficient.

In an emergency situation, it may not be reasonable to expect a user to perform a specific gesture. They may be unable or unwilling to perform due to injury, panic, loss of consciousness, or unawareness of the severity of the situation. For these reasons, we introduce the concept of “Natural Gestures”. A natural gesture is defined as a gesture made by an untrained human being in a particular situation. Obviously, this will vary from person to person and will differ slightly between cultures, but there are common themes with gestures related to personal health. For example, a person will instinctively clutch an open wound or painful area, even if there is nobody around to observe the action. By recognizing this as a natural gesture, we can determine that the user is in pain and send appropriate help.
System Design

Our system works by capturing the image of a person with the Kinect device, checking for any predefined gestures, and then sending a message to a visualization application that describes which body parts, if any, are in pain. We now give a breakdown of the system.

The Kinect acts as the eyes and ears of the system, so to speak. This device contains two cameras and a microphone, which allows it to determine distances accurately. Since we must have vision of a gesture in order to recognize it, these Kinect devices should be placed in multiple locations around the area where the patient is living. A separate system can be used to choose the active Kinect based on where the patient is. Our system is simple, and uses only a single Kinect.

Using the OpenNI framework, we interface with the Kinect, and allow it to calibrate to a patient. After the Kinect has been calibrated, it begins tracking their movements. This is done using a limb-based skeleton model.

Figure 1: The system is calibrating to a user
Each frame the Kinect captures is checked against a list of positions that may indicate pain. This is done by comparing the user’s joints to bounding spheres around the joints in the reference positions. If the patient’s joints fit inside the bounding spheres, they are verified as being in that position. After the patient is judged to be in pain, the system then rates the severity of the pain. This is calculated based on a combination of how long the position is held and which position is held. A pain in the chest is deemed more serious than a pain in the arm, and thus becomes more severe more quickly. A physician could also adjust this if necessary to account for special cases.

Note that the patient does not need to remain perfectly still. For example, one position is hand over the heart. This is intended to capture chest pains that may be indicative of a heart attack. While clutching the heart, the patient could walk around, bend over, or fall to their knees. However, as long as their hand remains over their heart, they are considered to be in the “chest pain” position. This is important, because people rarely stand perfectly still when in pain.

The system then encodes this information and sends it in a TCP packet to the visualization application. The packet contains a pain rating for each of the ten body parts currently defined in the system. This packet is sent once per second, regardless of whether pain was detected or not. This both
for simplicity and safety. If the visualization application does not receive a packet for a short period of time, then there is a malfunction and emergency personnel can be sent to repair the system to ensure constant monitoring.

The visualization system receives this TCP packet and displays areas of pain on a 3D model. This allows simple monitoring of dozens of patients from a single workstation. For more information on the visualization system, please see Ruhsary Rexit’s paper on the subject.

Results

We found that the Kinect-based natural gesture monitoring system worked well for an initial attempt. We were able to recognize several painful conditions, such as headache, choking, chest pain, stomach pain, and arm pain. After tweaking the parameters for the bounding spheres, we found very few misclassification errors, although we did have some false positives. This could be reduced by shrinking the bounding spheres, as well as adding more spheres to ensure the patient is actually in the determined position.
Each position was tested and found to be detectable under most circumstances. Some positions had multiple definitions to account for both hands, and these functioned well with either hand. One major source of error was when the hand became obscured. The OpenNI framework did not handle this well, and often showed the arm in strange positions. This is not a problem with our system, but rather the technology underlying it.

The system is responsive enough to be used in real-time, performing both data capture and position checking on-the-fly. This is obviously important in an emergency situation, as time is critical when a person is injured. Additionally, because the data sent is so small, the user should not notice any impact on their internet service. In fact, if the Kinect device was concealed well, the user might very well forget about it until the day they need it!

Overall, we believe that the results are very promising considering the low cost of the Kinect used and the widespread availability of the technology.

**Future Work**

There is a lot of work that could be done to expand upon this project. Firstly, as mentioned above, multiple Kinect devices could be used in order to cover multiple rooms or multiple angles. Additionally, switching from the OpenNI framework to the Microsoft framework could eliminate many accuracy problems. These two changes would be big steps toward the viability of this system.

Another important task is to create a system for physicians to define custom positions. A patient might have a rare condition that shows strange symptoms, and we would like to pick up on these. Along these same lines, we should strive to refine the definitions of positions in order to reduce false positives. This can be done by defining more bounding spheres, attempting to recognize facial expressions, or by combining multiple painful positions into a single trigger.

There are a few less critical tasks that would be nice to include in a future system. The Kinect system has a microphone system, so this should be utilized to detect if the user is yelling for help. In addition, we could set up a
speaker in the area that would notify the user that help has been contacted, which would be comforting to an ill patient.

Overall, the system has a lot of room for improvement. However, we believe that these important first steps show the possibilities of the system, and inspire future improvements.