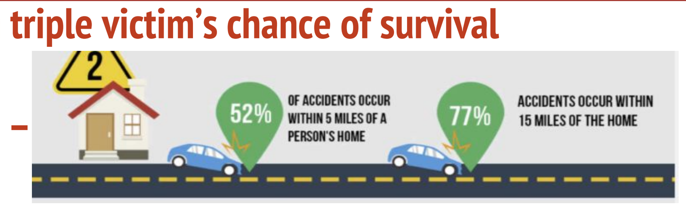
HeartWatch

A Smartwatch and Smartphone Based Realtime CPR Aid

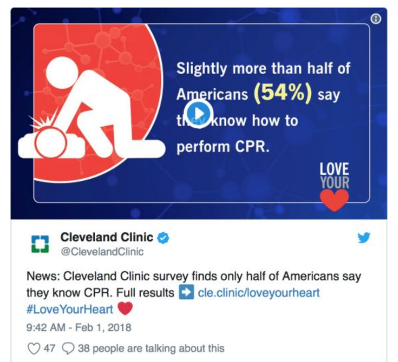
Daniel Greenberg, '19 Princeton University and Prof. Alan Kaplan, Princeton University

Motivation

- CPR is a lifesaving technique that can triple victim's chance of survival
- Many medical emergencies occur where bystanders are present to help – 77% of car accidents occur within 15 miles of one's home, and 88% of cardiac arrests occur in the home



- Unfortunately, only 11% of EMS responders record bystander intervention, and the AHA states that only 32% of cardiac arrest victims are administered CPR by a bystander
- Investigating this, the reason becomes apparent – only 51% of Americans know CPR, and only 30% feel confident enough to administer it



- An accessible realtime CPR aid may help in supplying CPR to victims in need

CPR Basics

- Hand Placement: Hand placement during CPR is crucial for correct compressions
  - Infants: Use two fingers, straight and held together, placed in center of chest, just below nipples
  - Other: Place heel of dominant hand 2in. above sternum, place other hand on top and interleave fingers

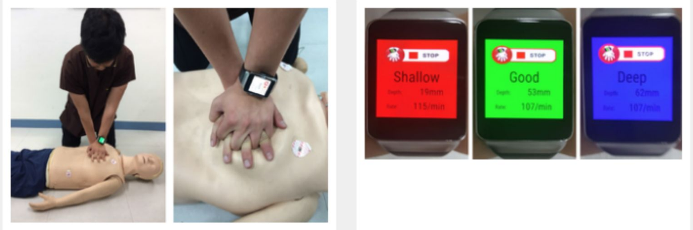
Infant Placement

Other Placement

- Compression Depth: Correct compression depth ensures proper blood circulation
  - Infants: Approx. 1.5 inches or 3.81 cm
  - Other: Approx 2.0 inches or 5.08 cm
- Compression Pace: Compressions administered at pace of 100-120/min
- Compressions and Breaths: Administer 30 seconds of compressions/ 2 breaths

Related Work

- A New Chest Compression Depth Feedback Algorithm for High-Quality CPR Based on Smartphone (Y. Song, 2015):
  - Proved viability of compression algorithm – mean error of 0.143cm
  - Because smartphone based, phone placed between victim's chest and administrator's hands – proving to be quite awkward
  - Not open source
- Smartwatches as Chest Compression Feedback Devices: A Feasibility Study (Y. Song, 2016):
  - Implemented same compression algorithm using a smartwatch, offering a more natural alternative
  - Results prove better than smartphone
  - Lack of accompanying smartphone application means no visualization/trends



Y. Song Smartwatch Approach

- Laerdal CPRMeter 2:
  - Well reviewed consumer option – more accurate due to use of pressure sensor
  - Minimum price of 695$, not an implementation using hardware consumers already have
  - Pressure sensor necessitates awkward chest placement

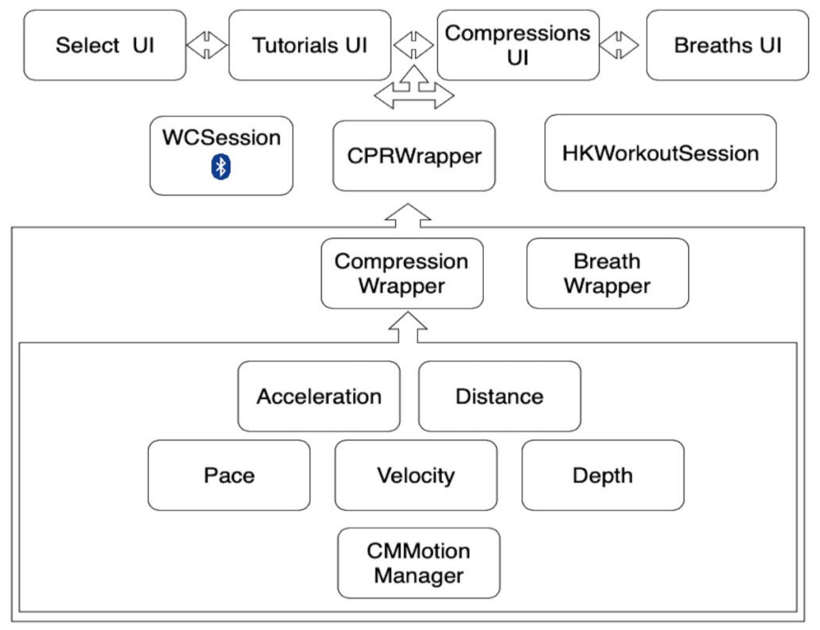
CPRMeter 2

Acknowledgements

I would like to thank my advisor, Professor Alan Kaplan, and our graduate TA, Molly Pan, in addition to the rest of Princeton's Computer Science staff for all of their help this semester - HeartWatch truly would not have been realized if it was not for their time and willingness to help.

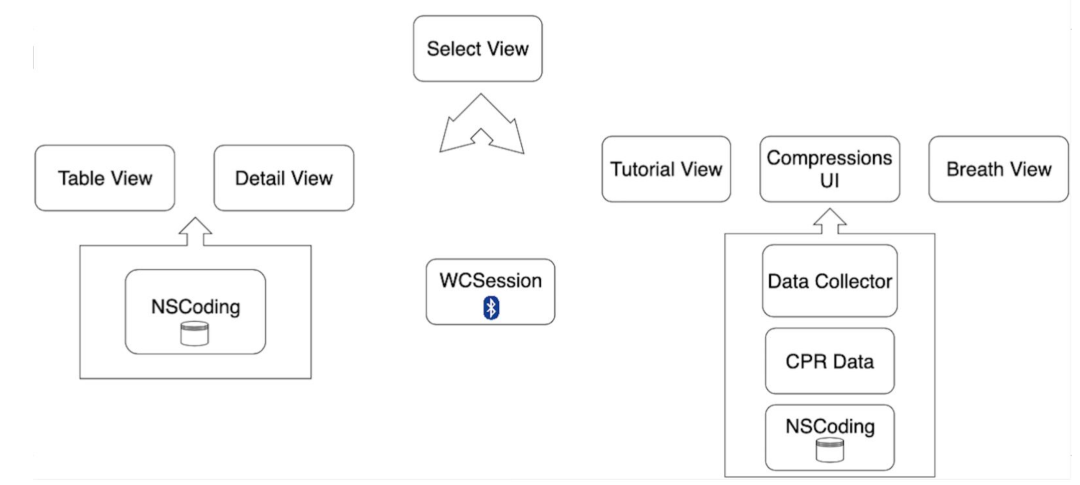
Application Architecture

watchOS



- WCSession: Wraps all Apple Watch - iPhone communication
- CMMotionManager: Supplies all Apple Watch sensor data
- HKWorkoutSession: Enables HeartWatch to run in background

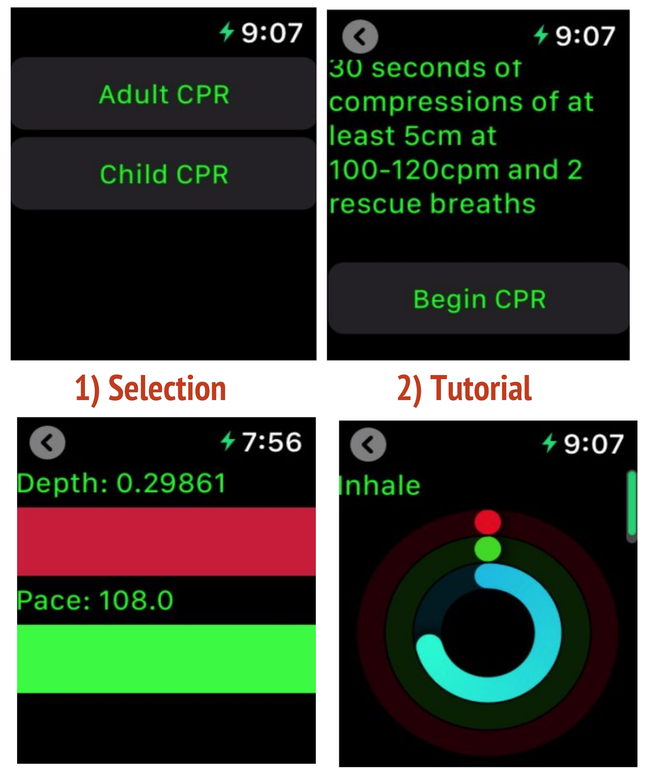
iOS



- Compression Wrapper: Wraps all CPR compression related logic
- Breath Wrapper: Wraps all CPR breath related logic
- Acceleration: Obtains and denoises raw accelerometer data, performs trapezoidal integration to obtain raw velocity values
- Velocity: Denoises raw velocity data, integrates to obtain distance
- Distance: Obtains and denoises raw distance data, passes to Depth and Pace

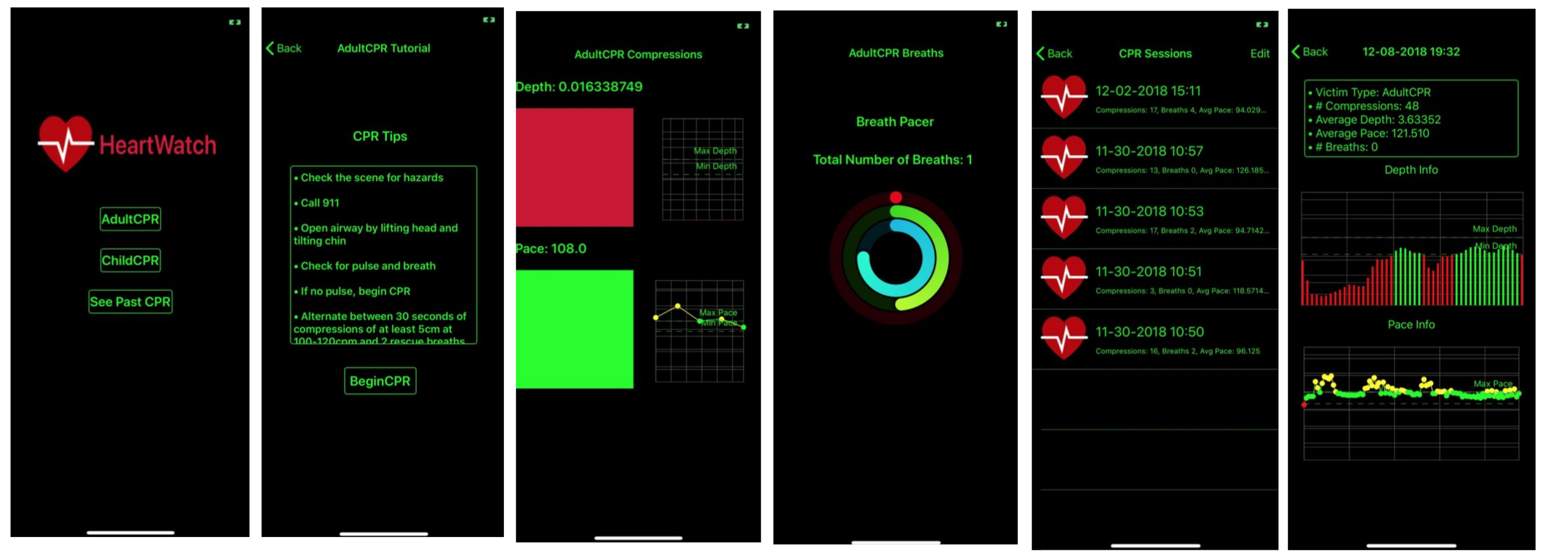
UI Overview

watchOS



1) Selection
2) Tutorial
3) Compressions
4) Breaths

iOS

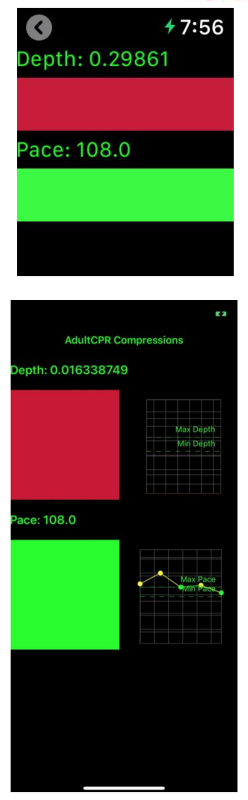


1) Selection 2) Tutorial 3) Compressions 4) Breaths 5) Saved 6) Detail

1. Selection: User selects victim type to ensure HeartWatch gives victim-specific feedback
2. Tutorial: Victim type specific CPR facts are displayed to the user
3. Compressions: Realtime CPR Compression feedback reported, after 30sec switches to Breaths
4. Breaths: Realtime Breath feedback given to the user, after 2 breaths switches to Compressions

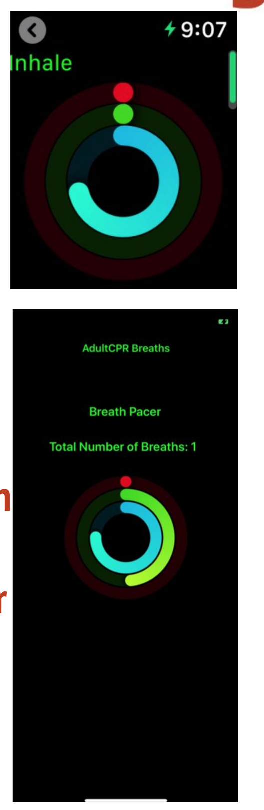
Detailed UI

Compressions



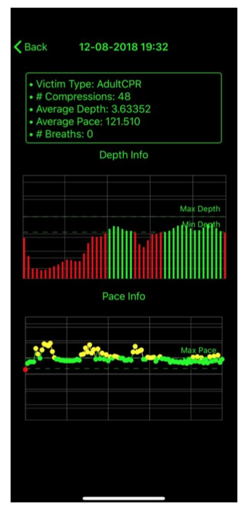
- Whenever a compression peak is detected, depth and pace values updated
- Colors communicated if values are appropriate:
  - Red = too little
  - Green = appropriate
  - Yellow = too much
- Vibrates every 0.5 seconds to guide compressions at 120/min
- Every 5 seconds, gives audio and haptic feedback to the user to communicate if values appropriate
- iOS application displays realtime charts

Breaths



- Guides users through 2 rescue breaths:
  - Inhale for duration of blue circle
  - Exhale for duration of red circle
  - Green circle is counter
  - Beeps and vibrations mark start/end of breath
- iOS application displays cumulative number of breaths for CPR session
- Switches to Compressions after 2 breaths

Detail



- Whenever paired with iPhone, HeartWatch saves CPR session data
- Detail screen displays:
  - Victim type
  - Total # compressions
  - Average depth
  - Average pace
  - Total # breaths
- Depth and pace data displayed graphically, marking optimal range on graph and using colors to communicate where datapoints are relative to correct range

Denoising

Singular Value Decomposition

$$H_x = \begin{bmatrix} x_1 & x_2 & \cdots & x_k \\ x_2 & x_3 & \cdots & x_{k+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_m & x_{m+1} & \cdots & x_N \end{bmatrix}$$

Hankel Matrix

$$H_x = U_x \Sigma_x V_x^T$$

SVD

- Method chosen to denoise raw accelerometer values
- Raw Value = Real + Noise

1. Construct Hankel Matrix of raw values – all values on counter diagonal same
2. Compute SVD of Hankel Matrix
   - Singular values "weighting factors"
   - Noise values orders of mag smaller ~ smaller SVs
3. Run heuristic to zero small SVs
   - Acts to zero out noise contribution
4. Matrix multiply to get denoised values

Transient Component Emphasis

$$RAW = R + C$$

$$RAW_2 - RAW_1 = (R_2 + C) - (R_1 + C) = R_2 - R_1$$

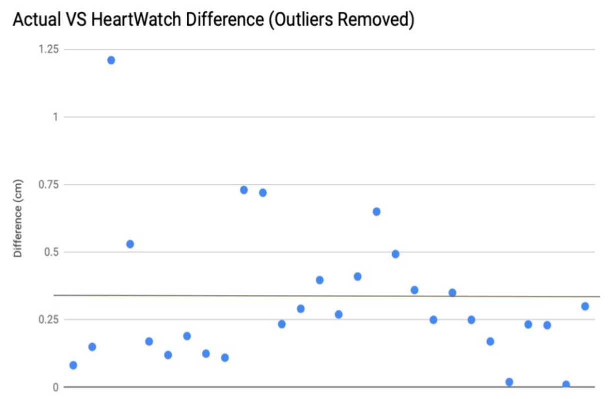
$$NEW = OLD + (R_2 - R_1) = OLD + (RAW_2 - RAW_1)$$

TCE

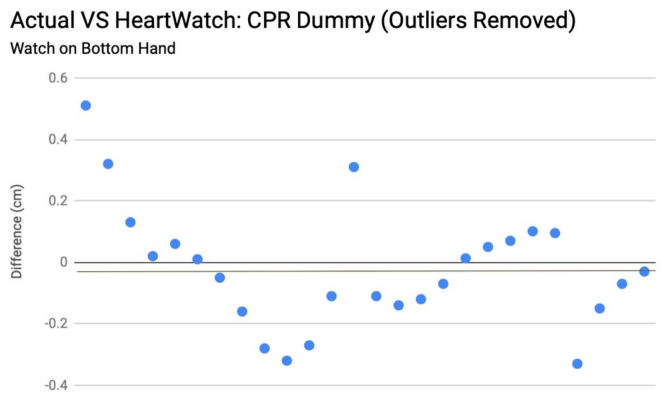
- Method chosen to denoise raw velocity and distance values
- Since these values not from sensor, sensor noise not issue
- Instead, as they are obtained via integration, integration constant biggest problem
- TCE allows for removal of constant:
  - Subtracting two integrated values means subtraction of integration constant – therefore canceling it out
  - Only change between real values remains

Depth Evaluation

Ideal



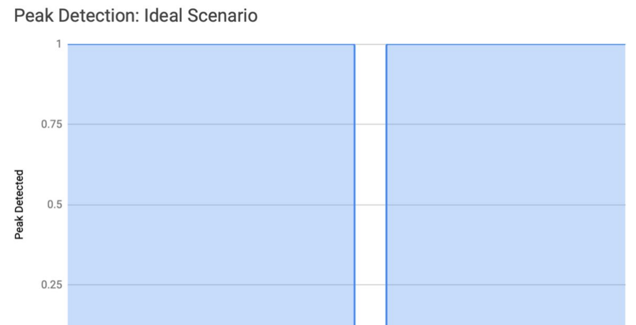
Watch on Bottom



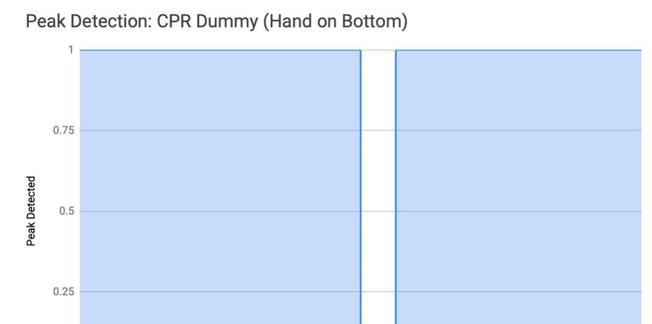
- Mean: 0.323cm – StdDev: 0.259cm
- Mean: 0.02cm – StdDev: 0.198cm
- For both Ideal and Real World Scenarios, move the Apple Watch a known distance and compare this to HeartWatch reading
  - Ideal used ruler and video to determine true distance
  - Real World used CPR mannequin (makes noise at ~3cm) to determine
- Compared to 0.143cm Mean and 0.100cm StdDev from the Song research group, HeartWatch results are quite promising
- Errors with testing hardware led to inaccuracies – more testing necessary

Peak Evaluation

Ideal



Watch on Bottom



- 94.3% of peaks successfully detected
- 93.8% of peaks successfully detected
- In both Ideal and Real World Scenarios, count number of actual peaks VS number of peaks HeartWatch registers, as determined via video recording
- Outliers in Depth Evaluation followed peaks that were not detected
- Compression Pace Evaluation not explicitly performed, since pace is just a simple mathematical transformation of the time between two registered peaks, determined as the number of timer firings in between peaks (60/sec)
- Further work in peak algorithm would make HeartWatch more consistent