Smart Health 15 (2020) 100089

Contents lists available at ScienceDirect

Smart Health

journal homepage: www.elsevier.com/locate/smhl

RadSense: Enabling one hand and no hands interaction for sterile manipulation of medical images using Doppler radar

Elishiah Miller [a,*], Zheng Li [a], Helena Mentis [b], Adrian Park [c], Ting Zhu [a], Nilanjan Banerjee [a]

[a] Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, United States
[b] Department of Information Systems, University of Maryland, Baltimore County, United States
[c] Simulation to Advance Innovation and Learning Center, Anne Arundel Medical Center, Annapolis, MD, United States

ARTICLE INFO

Keywords:
Healthcare
Human centered computing
Wearable devices
Gesture recognition
Busy hand interaction

ABSTRACT

In this paper, we show how surgeons can interact with medical images using finger and hand gestures in two situations: one hand-free and no hands-free interaction. We explain how interaction with only one hand or a couple of fingers is beneficial and can help surgeons have continuous interaction, without the need to release their tools and leave the operating table, saving valuable patient time. To this end, we present *RadSense*, an end-to-end and unobtrusive system that uses Doppler radar-sensing to recognize hand and finger gestures when either one or both hands are busy. Our system permits the following important capabilities: (1) touch-less input for sterile interaction with connected health applications, (2) hand and finger gesture recognition when either one or both hands are busy holding tools, extending multitasking capabilities for health professionals, and (3) mobile and networked, allowing for custom wearable and non-wearable configurations. We evaluated our system in a simulated operating room to manipulate preoperative images using four gestures: circle, double tap, swipe, and finger click. We collected data from five subjects and trained a K-Nearest-Neighbor multi-class classifier using 15-fold cross validation, achieving a 94.5% precision for gesture classification. We conclude that our system performs with high accuracy and is useful in cases where only one hand or a few fingers are free to interact when the hands are busy.

1. Introduction

Healthcare professionals, such as surgeons, use imaging technology on a daily basis for providing patient care. One challenging use-case that surgeons face, involves interacting with medical images in a sterile way, when either one or both hands are busy. The challenges are apparent as surgery is a collocated collaborate practice Mentis (2017) and the ability to learn and see the body is critical and difficult to achieve Mentis, Chellali, and Schwaitzberg (2014). To effectively communicate peculiar details of the anatomy, surgeons must talk, point, instrument tools, and interact with medical images while abiding by requirements for sterility Mentis (2017). When surgeons want to interact with images they are often holding tools in hand. To release or not release the tools is an important decision that the surgeon must make. In most cases, image interaction needs to occur while holding a surgical tool. In O'hara et al. (2014a, 2014b) they showed that there was a clinical need to provide image control while holding surgical instruments. For example, in minimally

* Corresponding author.
E-mail addresses: eli11@umbc.edu (E. Miller), nilanb@umbc.edu (N. Banerjee).

https://doi.org/10.1016/j.smhl.2019.100089

Available online 15 November 2019

invasive procedures, where surgeons use small cuts and a camera to see inside the body, surgeons must hold a laparoscope and scissor tool while interacting with medical images Mondada (2003). How to control and interact with an imaging system while utilizing both hands is therefore critical and challenging. It was suggested in O'hara et al. (2014a, 2014b) that image manipulation, using gestures and voice commands, can be combined with the use of surgical tools, preventing the surgeon from removing gloves and re-scrubbing, which takes precious time. From this understanding, we explain how previous systems will not meet these requirements and how we developed a new system that allows for finger and hand gestures in two situations: one hand-free and no hands-free interaction while holding surgical tools.

Previous gesture recognition systems, computer vision Mondada (2003); Tan, Chao, Zawaideh, Roberts, and Kinney (2013); Jacob, Wachs, and Packer (2013); Ebert, Hatch, Ampanozi, Thali, and Ross (2012); Feng et al. (2018); Ruppert, Reis, Amorim, de Moraes, and da Silva (2012) and wearable based Jalaliniya, Smith, Sousa, Büthe, and Pederson (2013); Schwarz, Bigdelou, and Navab (2011); Hettig et al. (2015), have been developed to help surgeons in fields such as radiology and urology interact with images in a sterile way. Majority of these systems require the surgeon to release their tools and are uncomfortable and get in the way. For example, vision-based systems, Microsoft Kinect being the most popular, show that interaction with images, requires the surgeon to use large hand motions to perform actions such as zooming in and out, panning, rotating, or changing the brightness. To perform these actions surgeons must stand at least four feet from the Microsoft Kinect, release their tools, and hold their hands positioned towards the device and monitor. In addition, tracking fingers while holding an object in hand is often a difficult task for the Kinect. Vision-based systems also require line-of-sight for interaction, are impacted by lighting, and fail when obscured by sterile drapes and sheets. On the other hand, wearable solutions do allow for interaction while hands are busy, but they are cumbersome and get in the way, as they most often use inertial measurement units (IMUs) or Electromyography (EMG) sensors, which require adherence to the skin or the area being observed for gesture recognition.

To mitigate the disadvantages presented by computer vision and wearable devices, researchers Chi et al. (2018, 2016, pp. 1–10); Yao et al. (2018); Li and Zhu (2016); Khan et al. (2016); Li, Robucci, Banerjee, and Patel (2015) have proposed the use of ubiquitous ambient signals such as WiFi, Bluetooth, and radio frequency (RF) for robust gesture and activity recognition. Ambient signals allow for sensing in different environmental conditions without devices that have to be worn. Majority of these systems have not been evaluated for gesture recognition with busy hands and utilize existing WiFi infrastructure, which we argue is problematic in a hospital setting. For example, devices connecting and reconnecting to the network, limit network bandwidth, which affects system accuracy and response. The WiFi access points are typically omni-directional and pick up background motion, which adds noise to the system. Also, hospitals have multiple windows, floors, and rooms, which creates high risk for spotty (WiFi dead zone) connections with not enough coverage.

To this end, we developed a new hybrid, both wearable and non-wearable, gesture recognition system we call *RadSense*, short for radar sensing. *RadSense* is an end-to-end, mobile, and unobtrusive directional radar system, that uses the Doppler Effect to sense in-air hand and finger gestures for continuous sterile medical image interaction when one or both hands are busy. The system captures motion using radar and wirelessly transmits the motion via Bluetooth Low Energy (BLE) network to a computer for gesture classification and image control. The system can be worn on the human body or attached to an object with Velcro enabling unobtrusive gesture detection, as it is small and does note require adherence to the skin. Accurate gesture detection can also occur when the system is covered by a sterile gown in order to meet requirements of sterility. We evaluated our system in a simulated operating room (OR) (see Fig. 1) to manipulate preoperative images using four gestures: circle, double tap, swipe, and finger click. We show how surgeons can use these gestures to interact with medical images two situations: one hand free and no hands free interaction. We collected data from five subjects and trained a K-Nearest-Neighbor multi-class classifier using 15-fold cross validation, achieving a 94.5% precision for gesture classification. The design, implementation, and evaluation of our system provides the following research contributions:

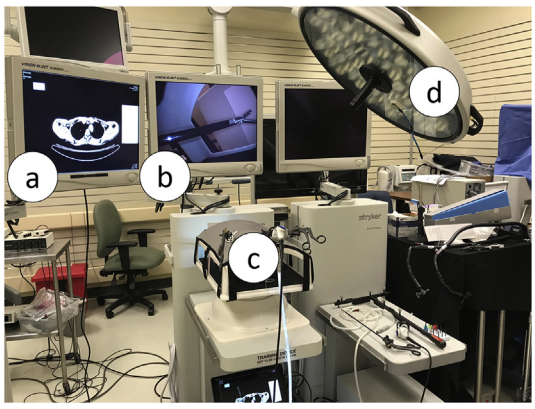

Fig. 1. Simulated OR: the MicroDicom CT scan image viewer application (a), the laparoscopic view (b), the laparoscopic trainer (c), the radar attached to an overhead light (d).

1. Design of a new end-to-end and unobtrusive gesture recognition system that uses radar and the Doppler effect for interacting with medical images. The system can be worn and covered by a sterile gown or attached to an object for unobtrusive hand and finger gesture recognition while holding surgical tools.
2. Hand and finger gesture detection when either one or both hands are busy using only two features: zero-crossings and magnitude difference of the signal. We show that our system can detect gestures with a precision of 94.5% using a K-Nearest Neighbor classifier with low false positives.
3. An evaluation of our system in a simulated OR using four gestures: circle, double tap, swipe and finger click.

2. Related work

Touch-less gesture recognition systems help facilitate with the sterile interaction between surgeons and the digital mediums they want to control de la Barre, Chojecki, Leiner, Muhlbach, and Ruschin (2009). In practice surgeons interact with two types of intra-operative imaging systems: a main display that shows the anatomy of the human body for operating, and a secondary display for pre-operative images Mentis (2017). For medical image interaction with these systems, several gesture recognition systems have been proposed and studied in literature with the focus in two main categories. This includes vision-based and wearable solutions. Each solution has its own set of limitations and advantages which we describe next.

2.1. Vision-based approaches

Vast majority of research uses vision-based gesture recognition systems like the Kinect O'hara et al. (2014a, 2014b); Yusoff, Basori, and Mohamed (2013); Lopes et al. (2017); Gallo, Placitelli, and Ciampi (2011) to capture in-air arm, hand and body motions, skeletal information, and voice commands for interacting with medical images. The Kinect uses depth, IR, color, and sound sensing. Fields such as radiology Tan et al. (2013); Jacob et al. (2013); Ebert et al. (2012, pp. 301–307); Feng et al. (2018) and urology Ruppert et al. (2012) have successfully used the Kinect to interact with 2D and 3D medical image data using hand and arm gestures. For example, Kenton O'hara et al. (2014a, 2014b) developed a gesture system for manipulating a 3D model in vascular surgery using the Kinect. In their study, they showed that surgeons could collaborate, communicate, and interact with the system in a sterile way using one hand, two hands, and voice controls. The surgeon however, is unable to hold a tool and track a finger, as the Kinect is more suited for hand, arm, and body tracking. For more fine grained gestures, researches have turned to devices for finger tracking like the Leap motion Bizzotto et al. (2014, pp. 655–656); Cho, Lee, Park, Ko, and Kim (2018). The Leap motion can accurately track objects while being held in the hand. A touch-free medical interface, using Leap motion, was developed in Nathaniel Rossol (2014), tracking a pen while being held in the hand for gesture recognition. When compared to the Kinect, Leap motion has a smaller field of view (FoV) and shorter range detection. We argue that positioning the device near the FoV of the surgeons hands, while operating can therefore be difficult. It was shown in P. Hughes (2015) that surgeons prefer the Kinect over the Leap motion due to its wider Fov and range, but found the Kinect to be tiresome, as surgeons were physically exhausted from holding their hands and arms up. Although useful, vision-based systems are limited by their range of detection, require line-of-sight sensing for interaction, are affected by lighting conditions [14], and fail when obscured by sterile drapes and sheets. Unlike vision based approaches, our system is instrumented with a radar, capable of detecting hand and fine grained finger motion while holding surgical tools. Our system is not as sensitive to lighting conditions and can travel through material, with the ability to be placed behind a sterile gown or attached to an overhead medical light for interaction.

2.2. Wearable approaches

Devices that can be worn on the human body, often requiring attachment to the skin and body parts, known as wearables, have been used for gesture interaction with medical images. For example, Shahram Jalaliniya et al. Jalaliniya et al. (2013) used inertial measurement units (IMUs) worn on the wrist and pressure sensitive floors (non-wearable) for detecting hand and foot gestures to interact with medical images. In addition, the use of inertial sensors Schwarz et al. (2011, pp. 129–136) worn on the body have been used for multiple user-defined gestures by tracking relative pose within a performed gesture. While others have used commercial products like the Myo arm band Hettig et al. (2015) for exploring 3D medical image data. We argue that these systems are uncomfortable, having to touch the skin, and are obtrusive, getting in the way of surgeon, not allowing the use of gestures while holding surgical tools. Our radar gesture interface can also be worn, but does not require attachment to arm, hands, or skin of the surgeon. Instead our radar can be attached on the chest via Velcro. We argue that this allows for an unobtrusive way to recognize gestures that may be more comfortable. Further more, we can attach our radar to other objects, such as a overhead medical light, for gesture interaction. In both cases, our system can be used while holding surgical tools.

3. The RadSense system

The *RadSense* system (see Fig. 2) allows a surgeon to perform gestures, while holding surgical tools, to interact with secondary intraoperative imaging systems. The system allows the surgeon to remain at the operating table, not having to release tools or scrub out in order to interact with images. The system works by using directional radar, to minimize background noise, and the Doppler effect to capture hand and finger motion. This motion is captured from the radar using a custom built data collection module. The data collection module sends the captured motion to a data relay unit which transmits the motion to a computer via BLE. The computer then receives the motion detects and classifies gestures, which are then used as commands to control a secondary intraoperative imaging system. In

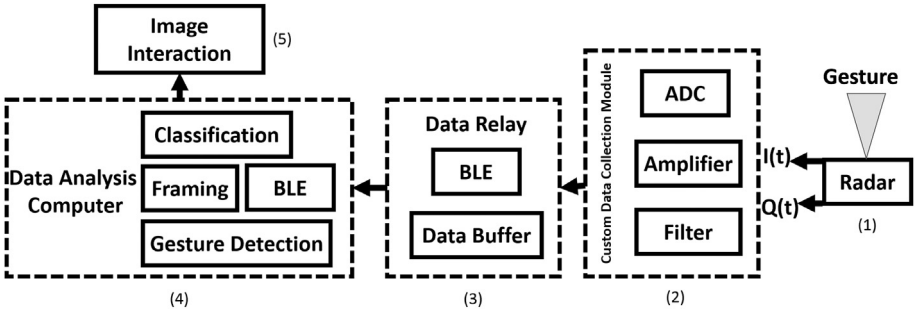

Fig. 2. System architecture design: from right to left, radar (1), custom data collection module (2), data relay (3), data analysis computer (4), image interaction (5).

the following sections we describe the details of how to interact with our system through gestures, how our system captures hand and finger motion for gesture detection, the hardware components of the system, an algorithm we developed to detect gestures with high precision, and the features we use to reliably distinguish gestures for classification.

3.1. Gesture set and interaction

We created four initial gestures that can be used for interaction with the secondary intraoperative imaging system shown in Fig. 3(a–d). Our system is not restricted to this gesture set as any gesture can be created, by following a set of rules. The rules include: varying the length of a gesture, making one motion longer than the other, varying the size of the gesture, making a large motion with the hand compared to a small gesture with the finger, and the location of the gesture, far or close to the radar sensor. We used these rules to develop four initial gestures that can be used while holding surgical tools in two cases described next.

3.1.1. One hand-free interaction

One hand interaction involves the surgeon having only one hand free to perform hand gestures and interact with images. In this scenario the surgeon can hold a surgical tool in one hand while still performing gestures with the second hand, shown in Fig. 4. The radar is worn on the chest of the surgeon for gesture detection. In this position, the surgeon can perform the following three gestures while holding one tool: hand swipe to scroll through images, a hand circle switch between commands, and a hand double tap to flip an image left or right (see Fig. 3(a–c)). The hand swipe occurs in front of the chest, the arm extended half the length of the arm, with a swinging motion either up or down. The circle is performed with the arm extended in front of the chest, the hand then moves either clockwise or counter clockwise, and performed five times, named circle 5 gesture, in order to create a long distinguishable gesture. The double tap occurs with the arm extended as far as possible in front of the chest. The four fingers of the hand are then moved towards the palm and then away from the palm twice.

3.1.2. No hands-free interaction

No hands free interaction involves the surgeon having no hands free, but only the index finger available to perform gestures. In this scenario the surgeon can hold two surgical tools, one in each hand while still being able to interact with medical images, shown in Fig. 5. For example, if using a laparoscopic grasper or scissors, the index finger used to manipulate the endefector is free. For this configuration, the radar is attached to an overhead medical light, so as to be in view of the finger. In this position, the surgeon can perform a finger click gesture, similar to pulling a trigger and then releasing, to flip an image left or right. Performing the same click can also be used to toggle between flipping left or right. In this case the radar is attached to an overhead medical light, angled towards the finger for gesture detection.

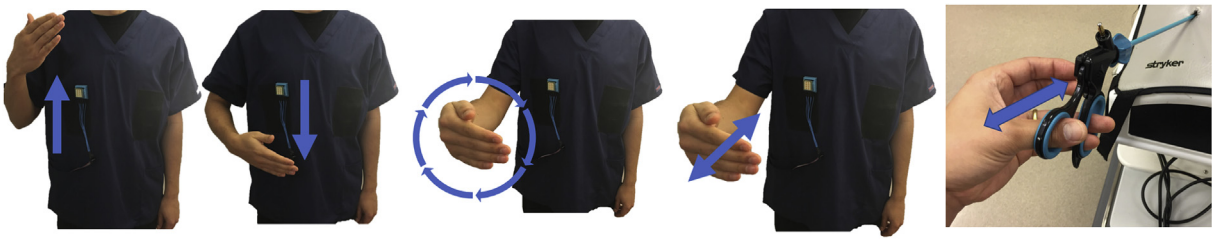

Fig. 3. Gesture Set: swipe (a), circle 5 (b), double tap (c), and finger click (d).

Fig. 4. One hand interaction: viewing the image (a), performing a gesture (b), RadSense worn on chest (c) busy hand holding tool (d).

Fig. 5. No hands interaction: RadSense attached to medical light (a), viewing image (b), performing finger gesture while holding a tool (c) busy hand holding tool (d).

3.2. Doppler effect primer

Our system uses the well known Doppler effect or Doppler Shift principal to sense hand and finger motion. The Doppler Shift phenomenon is observed when a source emits a signal that is reflected back by a moving object and the received signal is observed to have a change in frequency. In our approach, we chose a radar that has the source and receiver co-located. When a user moves their hand or fingers toward or away from the radar, it reflects the emitted radio waves causing a shift in frequency. The received signal captured by the radar can be described by the following equation:

$$f_r = f_t\left(\frac{c+v}{c-v}\right) \tag{1}$$

where, f_r is the frequency of the received wave, f_t is the frequency of the transmitted wave, v is the velocity of the object (moving hand/

fingers), and c is the speed of light. The shift in frequency is defined as $f_d = f_r - f_t$ and can be used to determine gestures.

3.3. Quadrature signal processing

Detecting the Doppler shift from hand and finger motion is done using quadrature signal processing. First, our radar system demodulates f_r from equation (1) into the baseband in-phase I and quadrature-phase Q. Demodulation is a process of detecting a received signal represented as a quadrature signal with real part I and complex part Q. In radar applications we can represent quadrature signals as complex numbers with in-phase I referring to the momentary amplitude of the real-signal and the quadrature-phase Q referring to the momentary amplitude of the real signal shifted by 90°. Once we obtain the demodulated I and Q components we can calculate the phase of the signal to determine direction of the hands and fingers. To do this we use the differentiated and cross-multiply algorithm (DACM) proposed in Wang et al. (2014) to compute the instantaneous phase of the quadrature signal to obtain f_d from equation (1). A positive phase value describes an object moving toward the radar and a negative phase value describes an object moving away from the radar. We use this principal to detect direction of the hands and fingers moving towards and away from the radar. The DACM algorithm given in discrete form Wang et al. (2014), i.e:

$$\Theta(n) = \sum_{k=2}^{n} \frac{I(k)*[Q(k) - Q(k-1)] - Q(k)*[I(k) - I(k-1)]}{I(k)^2 + Q(k)^2} \tag{2}$$

where $\Theta(n)$ represents the instantaneous phase and $I(k)$ and $Q(k)$ represent the k^{th} I and Q channel sample from the frame of size n. We chose the DACM algorithm as it has advantages over the widely used arctangent demodulation method that can handle the phase discontinuity problem by automatically phase unwrapping for phase reconstruction without ambiguities Gu (2016).

3.4. Hardware design

The hardware for our system (see Fig. 6) uses the K-LC2 25-GHz K-band Doppler radar transceiver RFBeam (2016), a custom built data collection module, and a RFduino micro-controller to obtain the demodulated I and Q signal channels for gesture recognition. Our hardware design is simple, small, and low cost; all components total less than $80. In the following sections, we introduce the major subsystems of our hardware platform.

3.4.1. Radar

To sense hand and finger motion, we used the K-LC2 continuous wave (CW) radar, capable of sending Doppler shifts. The K-LC2 cannot be used to detect the range or distance of an object, but only displacement of movement of the object due to lack of modulated spectral information Gu (2016). Two other types of radar technology exist known as frequency modulated continuous wave (FMCW) radar and pulse or impulse radar. Theses two types of radar are capable of sensing range information. We chose CW radar over FMCW and impulse radar as CW radar has higher accuracy in detecting movement, can operate in low power, and requires simpler hardware and signal processing techniques to condition the input signal Gu (2016).

3.4.2. Custom data collection module

To capture hand and finger motion, we designed a custom data collection module which connects to our radar, by leveraging the design in Li et al. (2015). To do this we created a printed circuit board (PCB), that amplifies and applies a low-pass filter on the captured I and Q signals, from the radar, for short-range movement detection. We define short range as the distance from the chest of a subject to end of their arm fully extended. The data collection module includes a 12-bit ADC that is used to transmit the I and Q signals to a Simblee RFduino micro-controller via a serial peripheral interface bus (SPI). We housed the data collection module inside a 3D printed case with Velcro attached to the back side. The Velcro allows our system to be worn on the human chest or attached to an object such as an

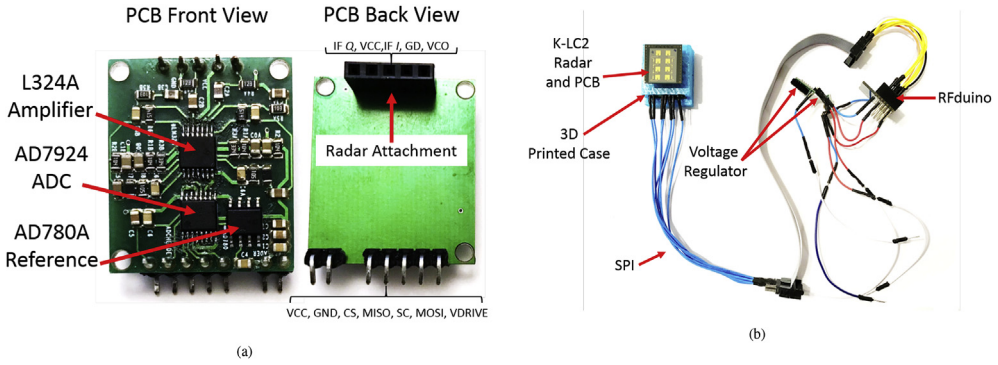

Fig. 6. Custom PCB design (a), including the front view of the circuitry (left), and back view of the radar attachment (right), and the Wearable Radar Interface design (b).

overhead medical light for gesture detection.

3.4.3. Data relay

To transmit the hand and finger motion from our custom data collection module to a computer, we designed a data relay module. Our data relay module uses the Simblee RFduino micro-controller. This micro-controller houses an ARM Cortex-M0, running at 16 MHz, and a BLE module used to transmit the digital I and Q signals sampled every 3 ms to a central computer.

3.4.4. Hardware power and memory requirements

The radar module consumes 35 mA at 5 V (0.175 W), the RFduino consumes 10 mA at 3.3 V (0.033 W), L324 consumes 0.8 mA at 3 V (0.0024 W), the AD7924 consumes 2 mA at 3.0 V (0.006 W), and the AD780 consumes 1 mA at 3.0 V (0.003 W). The total power consumption of the wearable radar interface is 0.2194 W while transmitting data. A 3.7 V lithium-ion rechargeable battery is used to power our system with appropriate step up/down regulators. The RFduino micro-controller program uses less than 24KB of memory.

3.5. Gesture detection

To reliably classify gestures, we developed a highly precise real-time gesture detection algorithm shown in Fig. 7. Our detection algorithm is used to detect the start and end of a gesture. Once the start and end of a gesture has been determined we calculate a set of features to classify gestures. We also, create a simple threshold to determine if there is no movement. This is explained in the following three sections.

3.5.1. Step 1

In step 1, We capture a frame size of 2080 samples of raw I and Q channel values from the radar shown in Fig. 7 (a). We chose a frame size of 2080 by observing the average window size used during labeling for data collection which was about 1040 samples. We wanted a larger frame size, double in our case, to allow for our frame to include additional motion information for easier detection as our gestures create an impulse in the signal and then flat-lines. This window size corresponds to about 6.23 s of motion for 2080 samples, sampled every 3 ms.

3.5.2. Step 2

We find the start and end points within the captured frame shown in Fig. 7 (b). To do this we calculate the velocity of the I and Q quadrature signal by taking the forward difference of the instantaneous phase using the DACM algorithm shown in equation (2). The forward difference is then calculated as:

$$\Theta_d(n) = \Theta(n+1) - \Theta(n) \tag{3}$$

where $\Theta_d(n)$ represents the phase difference corresponding to the gesture velocity. To de-noise the velocity signal while maintaining the shape of its original curve we apply a Savitzky-Golay (SG) filter with polynomial order equal to 21 and frame length equal to 89. The filtered velocity signal is a good indicator for detecting hand and finger motion as little motion happens between $+0.2$ m/s and -0.2 m/s and hand and finger motion happens outside this range. In Fig. 7 (b) it is clear that there is a larger variation in velocity where the gesture happens compared to where there is no movement. We use this observation to calculate the change points by observing where the root-mean-square (RMS) level of the velocity signal changes most significantly. The inflexion points help us identify the start and end of a gesture. For our purposes we want to identify two inflexion points that divide our gesture frame into three sections. We use the *changepoint* detection method Lavielle (2005); Killick, Fearnhead, and Eckley (2012) to find two change points $K = 2$, by minimizing the function:

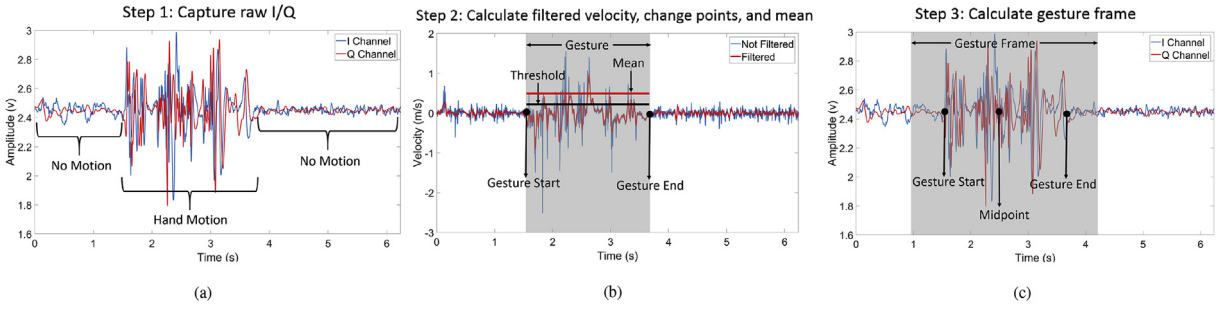

Fig. 7. Gesture detection algorithm: step 1, capture the raw I and Q with 2080 samples (a), step 2, calculate the velocity of motion, filter the velocity with Savitzky-Golay filter, calculate the change points that indicate the start and end of the gesture, and compare the gesture mean with a predetermined threshold (b). If the mean is greater than the threshold we continue to step 3 and calculate the I/Q gesture frame by finding the midpoint between change points and extending the frame 520 samples to the left and right (c).

$$J(K) = \sum_{r=0}^{K-1} \sum_{i=K_r}^{K_{r+1}-1} \Delta(x_i; \chi([x_{K_r} \ldots x_{k_r+1} - 1])) + \beta K \tag{4}$$

where $J(K)$ is the total residual error from the three sections for two change points $K = 2$, for a given signal $x_{K_r}, \ldots, x_{K_r+1}$, given the empirical estimate χ and the deviation measurement Δ, where β represents the fixed penalty added for each change point. We use the RMS empirical estimate statistic and find two change points indicating the start and end of a gesture. After finding the two change points we calculate the mean of the velocity signal between the two change points. We created a threshold $+0.2$ m/s by averaging a total of 60 s worth of collected velocity values with no movement. If the mean of the velocity signal, defined by the change points, falls below this threshold we ignore the gesture and consider it as no movement. In the other case we continue to step 3.

3.5.3. Step 3

We calculate the midpoint between the two change points and extend our frame 520 samples to the right and to the left of the midpoint Fig. 7 (c). We use these left and right points for the start and end of our raw I and Q channel gesture frame. If the gesture is off center within the 2080 sample frame and we can not extend our frame to the right or to the left by 512 samples, we truncate the frame to the max sample less than 512. At this point our gesture is still unknown, but we are more confident that a gesture happened. Our gesture frame is now created and we can continue to calculate features to classify the gesture.

3.6. Feature extraction

We experimentally developed two features to reliably classify our four gestures. To do this we extract two features from our captured gesture frame of 1040 samples. Previous works have used zero-crossing Gao et al. (2016) and the magnitude differences between the highest crest and the lowest crest through each signal Wan, Li, Li, and Pal (2014) achieving above 90% accuracy for classification. Also, the set of measurable properties using the Doppler effect were described in Gupta, Morris, Patel, and Tan (2012) for sound waves that are also applicable to radar. We base our feature set off this work and show how we can obtain higher than 94% accuracy for classification explained in our evaluation section. The two features are described next.

3.6.1. Sum of the zero-crossings

The first feature we calculated was the sum of the zero-crossings of the quadrature signal. We normalize each channel so that the signal is centered at zero along the x-axis. The zero-crossings is proportional to the phase of the signal. To find the zero crossings for each channel, I and Q, we look for the following conditions: $X_i < 0$ and $X_i - 1 > 0$ and $X_i > 0$ and $X_i - 1 < 0$ where X_i represents the i*th* sample for channel I or Q. We then take the sum of all zero-crossings found for both I and Q as a feature.

3.6.2. Magnitude difference

The second feature we calculate is the difference of the maximum, minus the minimum, magnitude. The magnitude of the signal is proportional to the amplitude of the signal. The amplitude is higher when the hand and fingers approach the radar and lower when they recede. The magnitude difference is calculated as:

$$\widehat{M} = max(M(t)) - min(M(t)) \tag{5}$$

where $\widehat{M}$ is the magnitude difference, and $M(t)$ is the magnitude defined as:

$$M(t) = \sqrt{I(t)^2 + Q(t)^2} \tag{6}$$

where t is the tth sample of the signal.

3.7. Importance of features

When the two features, discussed previously, are used together, they cluster the feature set into 4 categories (see Fig. 8 (a)). This is important as it makes our machine learning model more accurate. The magnitude difference helps distinguish a large gesture, hand performing circle 5 gesture, from a small gesture, finger performing finger click. The sum of the zero-crossings help distinguish between the length of the gesture, short, finger click and double tap, or longer, swipe and circle 5. The two features we chose may make our system less robust if additional gestures are added. To retain high accuracy of our system, we suggest to use additional sensors, each with their own trained models of just a few gestures. Thus, the gesture set can be extended by adding more sensors.

4. Evaluation

We evaluated our system in a simulated OR, by training a K-Nearest-Neighbor (KNN) multi-class classifier using a 200 labeled gesture data set composed of the sum of the zero-crossings and magnitude difference features described previously. We collected our data from five subjects, with Institutional Review Board (IRB) approval, two male, and three female. Each subject was asked to perform our four gestures 10 times each. We trained each subject on how to perform gestures, and then collected the gesture data. We used a

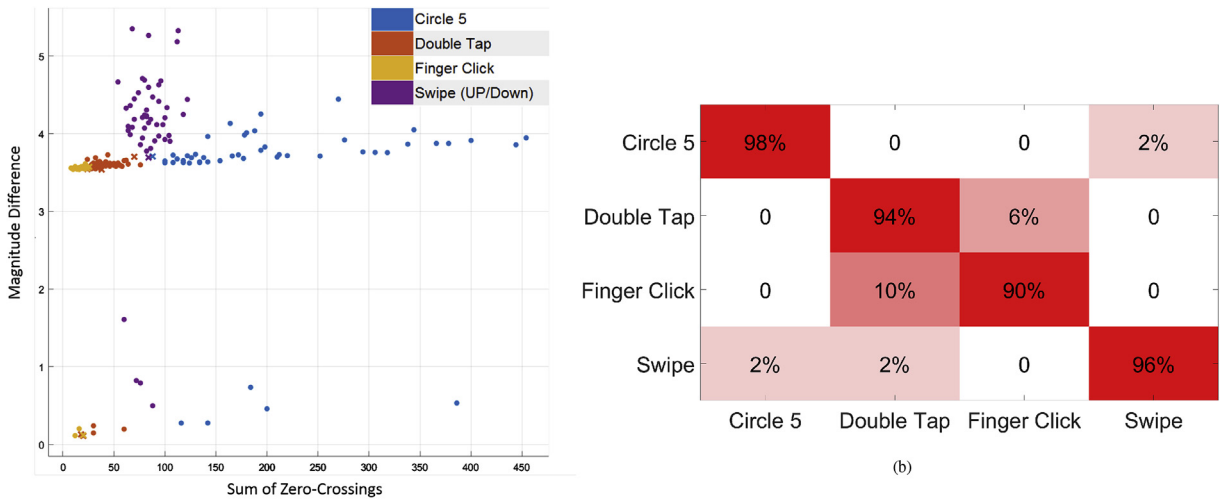

Fig. 8. Sum of zero crossing features, and magnitude difference, the circle indicates correct prediction and x indicates incorrect prediction (a), and Confusion matrix classification accuracy for radar training data (b).

camera to record video of the gestures for accurate labeling of each gesture. The magnitude difference (y-axis) and sum of zero-crossings (x-axis) features are shown in Fig. 8 (a).

4.1. KNN evaluation

For the KNN model we used the euclidean distance and squared inverse distance weight with K = 10 for the number of nearest neighbors. We evaluated our trained KNN model using 15-fold cross validation achieving 94.5% accuracy calculated from the confusion matrix (see Fig. 8 (b)). In detail we take the 200 examples divide them into 15 equal folds (13 examples per fold), then randomly shuffle the folds. Then K-1 folds are used for training (15-1 = 14) and the last fold (1) is used for validation. This process is repeated K (15) times, leaving one different fold for validation each time. After each K-iteration we calculate the classification accuracy and then take the average of all iterations for mean classification accuracy (see Fig. 9 (a)). In addition we varied the number of folds and found 15-fold cross validation to have the highest accuracy (see Fig. 9 (b)). For comparison, we also trained three additional models using 15-fold cross validation: cubic support vector machine (SVM) achieving 94.5% accuracy, Ensemble Bagged Trees (EBT) achieving 91% accuracy, and Fine Decision Tree (FDT) achieving 90% accuracy (see Fig. 10). In addition, we varied the number of folds for cross validation for the KNN: 5-fold achieving 93% accuracy and 10-fold achieving 93.4% accuracy. We also varied the features and found that using both features resulted in the highest accuracy of 94.5%.

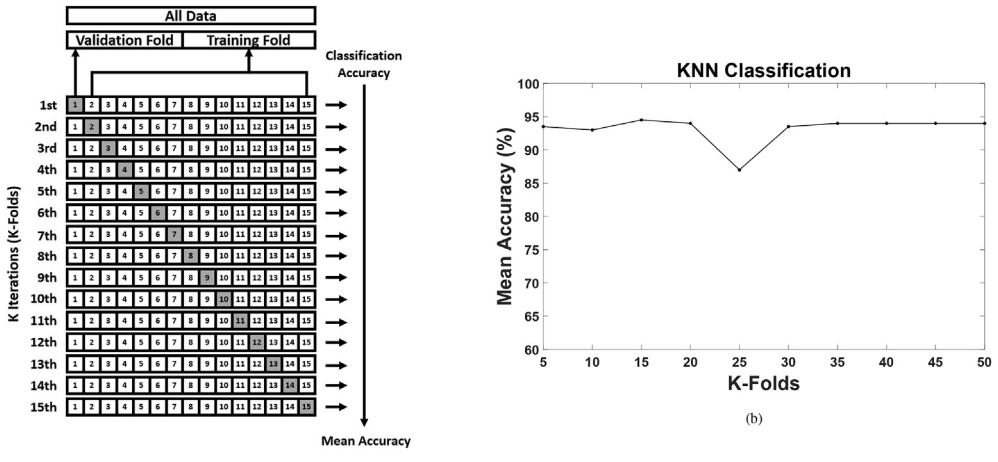

Fig. 9. 15 K-Fold cross validation process (a), and Evaluation of K-Fold cross validation (b).

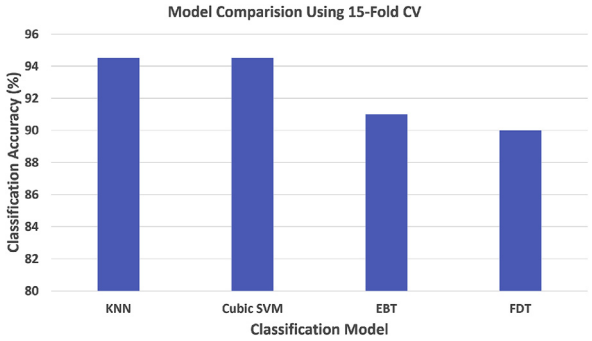

Fig. 10. Model comparison using 15-fold cross validation.

Table 1
Training precision and recall.

Gesture	Circle 5	Double Tap	Finger Click	Swipe
Precision	0.9800	0.9400	0.9000	0.9600
Recall	0.9800	0.8668	0.9375	0.9796

4.2. System precision and recall

To evaluate our trained KNN model we analyzed the precision and recall of our confusion matrix Table 1. Precision measures the proportion of positive identifications of gestures that were actually correct. Recall measures the proportion of actual positive gestures that were correctly identified. We can see in Table 1 that the precision for circle 5 is 0.98, with 2% classified as swipe, with recall 0.98. The precision and recall for double tap is 0.94 with recall of 0.8668 with 6% classified as finger click. The precision and recall for finger click is 0.90 with recall 0.9375 with 10% being classified as double tap. The swipe gesture precision is 0.96 and recall is 0.9796, with 2% of the gestures classified as circle 5 and another 2% classified as double tap. The biggest discrepancies were between the double tap and finger click gestures as they were similar in movement (two motions) and distance (arm length similar to distance between overhead light and finger position).

4.3. Detection through material

We also found that our system can reliably classify gestures when covered by material (see Fig. 11 (a)). This is particularly useful for meeting requirements of sterility, as surgeons can wear the *RadSense* system underneath their sterile gown. We performed a preliminary evaluation and trained a new, supervised KNN model and removed the finger click gesture. In this experiment, we had one subject wear the radar on the left side of the chest and then wear a sterile gown over the radar. The subject performed each gesture 10 times each. We were able to achieve an accuracy of 97% when covered by a sterile gown (see Fig. 11 (b)). The precision and recall are shown in Table 2.

5. Discussion and results

We tested our system in a simulated OR (see Fig. 1). We used a monitor to display CT scan images using the MicroDicom application. Images were retrieved from the national institute of health (NIH) national biomedical imaging archive (NBIA). The MicroDicom

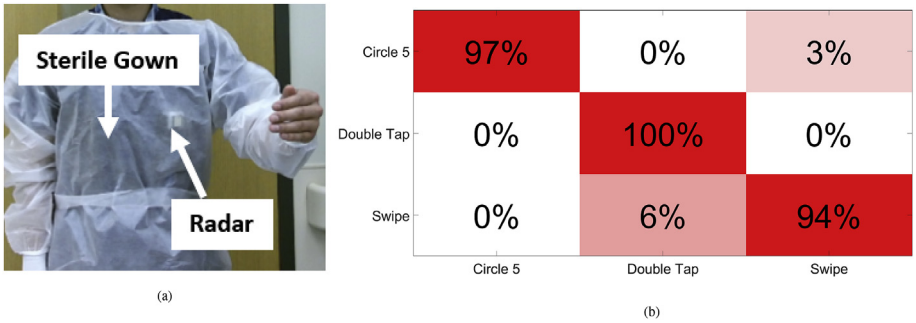

Fig. 11. The radar worn on the chest of a subject behind a sterile gown (a), and the confusion matrix results when the radar is covered by the sterile gown (b).

Table 2
Training precision and recall.

Gesture	Circle 5	Double Tap	Swipe
Precision	0.9700	1	0.94
Recall	0.9691	1	0.9434

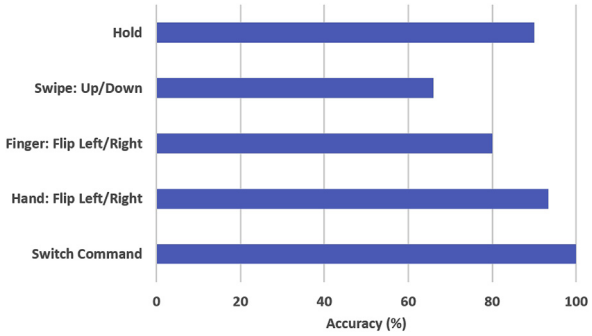

Fig. 12. Accuracy of commands when using real-time system.

application allows for scrolling, zooming, rotating, panning, measuring, and drawing on a CT scan image. Finally, we mapped our gesture set to a set of commands to control the CT scan images. We mapped the following gestures to the following commands: swipe to scrolling up and down, double tap to flip the image left and right with the hand, finger click to flip the image left and right, done while holding a surgical tool, and circle 5 to switch between commands, up or down for swipe, and left or right for double tap and finger click.

We tested our commands by having five subjects perform the gestures 10 times each then took the average of correctly executed commands. The system was able to detect gestures and control the CT scan images shown in Fig. 12, but future work is required to analyze the usability and failure rate of our system. The real-time system was particularly bad at detecting the swipe gestures as we observed that users swiped to close to the radar, or shifted the radar up and down on the shirt when performing a swipe. Also, the finger click gesture, often was classified as a double tap, but since they were mapped to the same command users did not notice. This miss-classification between the finger click and double tap, was not a big issue for users, as the two gestures were used in two different configurations: one hand interaction (using the hand) or no hands interaction (using the finger). The hold gesture was not explicitly trained, but rather, any movement below our threshold was assumed to be a hold.

6. Conclusion

In this paper, we presented an end-to-end and unobtrusive system that uses Doppler radar sensing to recognize hand and finger gestures when either one hand or both hands are busy. Our system permits the following important capabilities: (1) touch-less input for sterile interaction with connected health applications, (2) hand and finger gesture recognition when either one or both hands are busy holding tools, extending multitasking capabilities for health professionals, and (3) wearable, mobile, and networked, allowing for custom configurations. We evaluated or system in a simulated OR, using five subjects, by training a KNN multi-class classifier achieving 94.5% gesture recognition accuracy. We also tested our real-time system, capable of detecting the start and end of a gesture for classification. We mapped four gestures: double tap, circle, swipe, and finger click to commands to control a CT scan image application. We found that future work is required to evaluate the usability of our system as well as ways to more accurately process the radar signal in noisy conditions. To the best of our knowledge, this is the first system to successfully use radar for image interaction in the OR.

Acknowledgments

This work is supported in part by National Science Foundation (NSF), USA, grants CNS-1539047 and CNS-1652669. This work is also supported in part by the James and Sylvia Earl Simulation to Advance Innovation and Learning Center (SAIL). We acknowledge Rashmi Prava Patro for helping with data collection.

References

de la Barre, R., Chojecki, P., Leiner, U., Muhlbach, L., & Ruschin, D. (2009). Touchless interaction-novel chances and challenges. *International Conference on HCI*, 161–169.

Bizzotto, N., Costanzo, A., Bizzotto, L., Regis, D., Sandri, A., & Magnan, B. (2014). Leap motion gesture control with osirix in the operating room to control imaging: First experiences during live surgery. *Surgical Innovation*.

Chi, Z., Yao, Y., Xie, T., Huang, Z., Hammond, M., & Zhu, T. (2016). *Harmony: Exploiting coarse-grained received signal strength from iot devices for human activity recognition* (pp. 1–10). ICNP.

Chi, Z., Yao, Y., Xie, T., Liu, X., Huang, Z., Wang, W., et al. (2018). Ear: Exploiting uncontrollable ambient rf signals in heterogeneous networks for gesture recognition. In *Proceedings of the 16th ACM conference on embedded networked sensor systems* (pp. 237–249).

Cho, Y., Lee, A., Park, J., Ko, B., & Kim, N. (2018). Enhancement of gesture recognition for contactless interface using a personalized classifier in the operating room. *Computer Methods and Programs in Biomedicine*, 39–44.
Ebert, L. C., Hatch, G., Ampanozi, G., Thali, M. J., & Ross, S. (2012). You can't touch this: Touch-free navigation through radiological images. *Surgical Innovation*.
Feng, Y., McGowan, H., Semsar, A., Zahiri, H. R., George, I. M., Turner, T., et al. (2018). A virtual pointer to support the adoption of professional vision in laparoscopic training. *International Journal of Computer Assisted Radiology and Surgery*, 1463–1472.
Gallo, L., Placitelli, A. P., & Ciampi, M. (2011). Controller-free exploration of medical image data: Experiencing the kinect. In *2011 24th international symposium on computer-based medical systems CBMS* (pp. 1–6).
Gao, X., Xu, J., Rahman, A., Yavari, E., Lee, A., Lubecke, V., et al. (2016). Barcode based hand gesture classification using ac coupled quadrature Doppler radar. In *2016 IEEE MTT-S international microwave symposium (IMS)* (pp. 1–4).
Gu, C. (2016). Short-range noncontact sensors for healthcare and other emerging applications: A review. *Sensors*, 1169.
Gupta, S., Morris, D., Patel, S., & Tan, D. (2012). Soundwave: Using the Doppler effect to sense gestures. In *Proceedings of the SIGCHI conference on human factors in computing systems CHI '12* (pp. 1911–1914).
Hettig, J., Mewes, A., Riabikin, O., Skalej, M., Preim, B., & Hansen, C. (2015). Exploration of 3d medical image data for interventional radiology using myoelectric gesture control. In *Proceedings of the eurographics workshop on visual computing for biology and medicine VCBM '15* (pp. 177–185).
Hughes, P., Healy, N., Sheehy, N., O'Hare, N., & Nestorov, N. (2015). Comparing the utility and usability of the microsoft kinect and leap motion sensor devices in the context of their application for gesture control of biomedical images. *European Society of Radiology*.
Jacob, M. G., Wachs, J. P., & Packer, R. A. (2013). Hand-gesture-based sterile interface for the operating room using contextual cues for the navigation of radiological images. *Journal of the American Medical Informatics Association*, e183–e186.
Jalaliniya, S., Smith, J., Sousa, M., Büthe, L., & Pederson, T. (2013). Touch-less interaction with medical images using hand and foot gestures. In *Proceedings of the 2013 ACM conference on pervasive and ubiquitous computing adjunct publication UbiComp '13 adjunct* (pp. 1265–1274).
Khan, M. A. A. H., Kukkapalli, R., Waradpande, P., Kulandaivel, S., Banerjee, N., Roy, N., et al. (2016). Ram: Radar-based activity monitor. In *IEEE INFOCOM 2016 - the 35th annual IEEE international conference on computer communications*.
Killick, R., Fearnhead, P., & Eckley, I. (2012). Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 1590–1598.
Lavielle, M. (2005). Using penalized contrasts for the change-point problem. *Signal Processing*, 1501–1510.
Li, Z., Robucci, R., Banerjee, N., & Patel, C. (2015). *Tongue-n-cheek: Non-contact tongue gesture recognition* (pp. 95–105).
Li, Y., & Zhu, T. (2016). Gait-based wi-fi signatures for privacy-preserving. In *Proceedings of the 11th ACM on asia conference on computer and communications security* (pp. 571–582).
Lopes, D. S., de Figueiredo Parreira, P. D., Paulo, S. F., Nunes, V., Rego, P. A., Neves, M. C., et al. (2017). On the utility of 3d hand cursors to explore medical volume dataasets with a touchless interface. *Journal of Biomedical Informatics*, 140–149.
Mentis, H. M. (2017). Collocated use of imaging systems in coordinated surgical practice. In *Proc. ACM hum.-comput. Interact* (pp. 78:1–78:17).
Mentis, H. M., Chellali, A., & Schwaitzberg, S. (2014). Learning to see the body: Supporting instructional practices in laparoscopic surgical procedures. In *Proceedings of the SIGCHI conference on human factors in computing systems CHI '14* (pp. 2113–2122).
Mondada, L. (2003). Working with video: How surgeons produce video records of their actions. *Visual Studies*, 58–73.
Nathaniel Rossol, R. S. A. B., & Cheng, I. (2014). Touchfree medical interfaces. In *International conference of the IEEE engineering in medicine and biology society* (pp. 6597–6600).
O'hara, K., Gonzalez, G., Penney, G., Sellen, A., Corish, R., Mentis, H., et al. (2014a). Interactional order and constructed ways of seeing with touchless imaging systems in surgery. *Computer Supported Cooperative Work*, 299–337.
O'Hara, K., Gonzalez, G., Sellen, A., Penney, G., Varnavas, A., Mentis, H., et al. (2014b). Touchless interaction in surgery. *Communications of the ACM*, 70–77.
RFBeam. (2016). Klc2 radar. URL: http://www.rfbeam.ch/.
Ruppert, G. C. S., Reis, L. O., Amorim, P. H. J., de Moraes, T. F., & da Silva, J. V. L. (2012). Touchless gesture user interface for interactive image visualization in urological surgery. *World Journal of Urology*, 687–691.
Schwarz, L. A., Bigdelou, A., & Navab, N. (2011). *Learning gestures for customizable human-computer interaction in the operating room*.
Tan, J. H., Chao, C., Zawaideh, M., Roberts, A. C., & Kinney, T. B. (2013). Informatics in radiology: Developing a touchless user interface for intraoperative image control during interventional radiology procedures. *RadioGraphics*, E61–E70.
Wang, J., Wang, X., Chen, L., Huangfu, J., Li, C., & Ran, L. (2014). Noncontact distance and amplitude-independent vibration measurement based on an extended dacm algorithm. *IEEE Trans. Instrum. Meas*, 2666–2674.
Wan, Q., Li, Y., Li, C., & Pal, R. (2014). Gesture recognition for smart home applications using portable radar sensors. In *2014 36th annual international conference of the IEEE engineering in medicine and biology society* (pp. 6414–6417).
Yao, Y., Li, Y., Liu, X., Chi, Z., Wang, W., Xie, T., et al. (2018). Aegis: An interference-negligible rf sensing shield. In *Proceedings of the 37th annual IEEE international conference on computer communications* (pp. 1718–1726).
Yusoff, Y. A., Basori, A. H., & Mohamed, F. (2013). Interactive hand and arm gesture control for 2d medical image and 3d volumetric medical visualization. *Procedia - Social and Behavioral Sciences*.