# Design Principles and Need for a Universal Accepted language of Gestures for Touch-Free HCI


## Abstract

Since the beginning of the computer era, human-computer interaction via a human-machine interface (HMI) – using touch sensing devices, such as keyboard, mouse, and touch screens. Current touch-free technologies are gaining more and more attention, taking user experience to a new level of engagement. The design and selection of hand gestures for the touch-free interface should follow the principles of natural language. This paper presents new design principles that designers should consider for designing the gestures for a touch-free interface. Alongside, this paper highlights the need for universally accepted gesture language to rely on when designing interfaces. The unique experience of sign language interpreters can guide in designing gesture language for computer input.


## 1 Introduction

Human-computer interaction has come a long way from its inception but still requires using a keyboard, mouse, or touch as an input. HCI has become an increasingly important aspect of our daily lives due to a massive influx of computing devices in our society. Even though the current input methods are a better way to interact, the leaps in the innovation and progress of computing and display technologies demand to create a more effective and efficient way of communication which is not possible with the current devices. To perform a task, the user still has to press buttons, type on keyboards, or touch swipes on mobile devices. These methods limit the speed and naturalness of interacting with the computer. The creates the possibility of developing next-generation UI through Natural User Interface (NUI).

The Natural User Interface is a type of user interface that is designed to feel as natural as possible making the interface itself seem to disappear to create a seamless interaction between the human and machine. This type of interface requires no other tool other than human speech, hands, or face to interact with the machine. Many advancements have been made with the speech recognition interface and successfully utilizing them in commercial use. Similarly, the rapid progress of gesture recognition in recent years has allowed product designers to introduce new gesture recognition-based solutions to the market. These include a class of techniques based on the human arm and hand movement. These hand gestures provide a means of non-verbal interaction with a computer.

By using computer sensors, gesture recognition detects and understands human gestures and movements. This type of technology isn't new, as we have been using touch-based gesture devices for more than a decade or even more. The recent advancements in fast computing and real-time vision processing allowed us to focus on implementing mid-air hand gestures as a way of interacting with computers. To develop the next generation touch-less UI, the designers cannot inherit old techniques directly. They need to develop new design principles for a touch-free interaction to take users to an entirely new level of engagement.

Many challenges are associated with developing design principles of touch-free gesture-based interfaces. The designers should follow certain rules such as avoiding using standard touch-based or device input models since the standard models fail to account for touch-free gesture inputs making it difficult for a user to interact. The designers are required to design gestures that are easier to use and don't require a lot of physical work. This can cause huge problems to the UI designers as there is no universal language of gestures that they can rely on while designing interfaces. This paper focuses on the need for a universal language of gestures and methods that we can adapt to achieve it. We explore the previous implementations of mid-air gestures and associated problems and limitations of their user interfaces.

## 2 Touch-Free Interfaces in use

Many devices have been developed to utilize the mid-air hand gesture interface over the decade. With the rapid progress in gesture recognition along with the falling cost of computation and sensors, more products are being developed and made available in the market for purchase. Even though some of these products showcase incredible hardware to make hand-gesture recognition possible, they have never been a part of our daily lives due to limitations in UI design to incorporate these gestures effectively.

Microsoft Kinect is one of the devices to use a gesture recognition system to track mid-air gestures and speech. It is mainly used by Xbox users to interact and play games with their hand movements. At the time, when it was released in 2010, Kinect meant to replace the controller. But this device has limitations and not every game developer was able to incorporate it into the game. Later this device is discontinued by Microsoft.

Leap Motion controller has a motion sensor to capture and understand mid-air gestures to interact with the computer and it is meant to replace a mouse. Using this device, some researchers were able to develop sign language recognition by a proposed system that uses Multi-Layer Perceptron (MLP) neural network with Back Propagation to recognize different signs.

Fig 1: *Leap motion usage. The small object in the middle is Leap Motion Controller connecting to the Mac on the right. Hand on top of the Leap Motion is tracked and interacted with virtual objects*

Microsoft HoloLens is introduced by Microsoft in 2015 which is a mixed reality smart glasses containing windows 10. This device has a head-mounted display that is capable of voice detection, sensing mid-air gestures, and creates an augmented reality environment. The hologram can be used to display information, blend with a real word or even simulate a virtual world. Microsoft implemented various hand gestures within its UI so that the user can interact with the information on the display using their hand movements.

Google has been working on mid-air gesture recognition and they have introduced this system in pixel 4. This smartphone has a miniature radar that can detect and understand the human motion of various scales. The intention is to understand our natural body language and gestures as a form of input allowing us to connect and interact with the device without words. Even though the technology and hardware were exceptional, the lack of impressive user interface within the existing applications and limited gestures have limited this attempt to become mainstream.

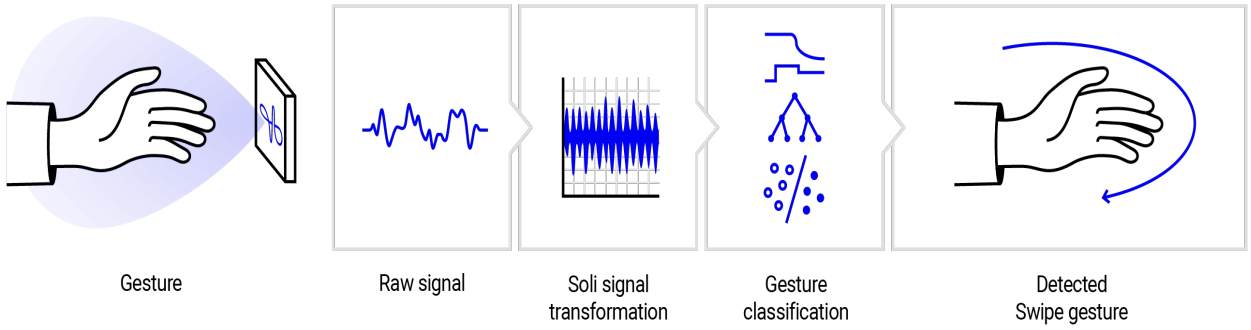


Fig.2: *Gesture recognition using soli sensor by Google*

Hands-free gesture recognition has become a huge part of virtual reality since it improves immersion and allows for natural interaction with the virtual world. Many VR products using some type of controller to track hand-movements and this might offset users from achieving complete immersion. Some VR devices such as Oculus quest 2 allow for free-hand movement by using onboard cameras to track the hand movement.

## 3 Design principles for touch-free gesture-based interfaces

The rapid progress of gesture recognition in recent years allowed product designers to develop solutions to design touch-free HCI. At the same time, to design these new input methods designers need to consider new design principles. These new rules will be vital to understand what kind of gestures needed to be introduced effectively and supported during product development.

The following presents fundamental design principles that are completely independent of technology such as:

- Visibility
- Consistency
- Feedback
- Non-destructive operations
- Discoverability
- Non- destructive operations
- Discoverability
- Scalability
- Reliability

Designers can consider existing heuristics, but they will need to adapt them based on the new type of interaction. Here are some design principles that designers need to consider while working on gestures for a touch-free interface:

**Stop considering the WIMP or touch-based models**
Designers often rely on using WIMP (Windows, Icons, Menus, Pointers) which is a default model for developing desktop applications and they can't simply replace the mouse pointer with the human finger. This is one of the common pitfalls as it fails to account for human natural motions. This would also apply to using touch-based models as it might not work for touch-free interaction.

**Make gestures comfortable for users unless that is the goal**
Designers need to make sure the interaction with UI is comfortable so that the users don't tire quickly as they interact with their arms. The designers should consider the ergonomics of a human body and too repetitive or uncomfortable gestures can make a user abandon the product. Making users do a lot of gestures to interact with UI can be annoying unless they are doing physical exercises or playing games. Due to this, the gestures requiring a lot of physical work should be avoided. The designers will need to consider user sessions since gesture controls are impressive for short periods but fail for longer sessions.

**Provide realistic and consistent feedback**
The gesture design should be aimed at being both simple and high responsiveness to express direct user control over touch interactions. The latency in the UI response should be minimal as users will notice any lag between the gesture and the actual response.

**Anticipatory design**
This is based on the idea of predicting user intent. This has been implemented in various touch-based UI predicting the user's next step. When it comes to touchless gesture-based UI, you have more tools to predict what users are trying to achieve. Based on the user's focus and recent actions, you can predict the next step.

**Gestures should be appropriate to its function**
The implementation of gestures will vary based on the product and the company's development. So, the learning curve for interactions will be a problem for gesture-driven devices. Due to this, it is recommended to use intuitive gestures which can be used by any user. The designers can utilize gestures from real life such as sign language to perform actions. Analyze the users' natural movement of their arms and hands and introduce these patterns into the UI. The designers need to avoid using complex gestures as it would become cumbersome for the users to memorize various moves to perform any actions.

**Socially appropriate gestures**
A user will not interact an interface that makes them look stupid in the public space. So the gesture implemented should be appropriate and comfortable for the user to use in any public location.

## 4 Need for a universal accepted language of gestures

For designing interface for the computer applications, designers often follow certain guidelines which doesn't vary much among organizations. Designers have been accustomed to being using WIMP – Windows, Icons, Menus, Pointers as a default style of interaction with the machine. This style has been modified or adapted but they rarely replace it with another type of user interface. This type of UI has been accepted by everyone and even allowed the developers to focus on the functionality and practicality of the application rather than wasting their resources to develop a new way of interaction. Even Users have been comfortable interacting with computers with the same input devices mouse and keyboard for decades.

Even touch devices have established gesture types and guidelines for the designers to rely on while designing the touch interface. According to Material Design, gestures are classified into different types: Navigational gestures, Action gestures, and Transform gestures. Navigational gestures help users to move through the product easily which include tap, scroll, and pan, drag, swipe, and pinch. Action gestures can perform actions through a tap, long press, and swipes. Transform gestures can transform an element's size, position, or rotation through gestures.

But for the touch-free devices, there is no established language of gestures using which a designer can develop UI for any type of user. This causes problems for the UI designers as gestures are hidden controls i.e., they do not leave behind any record of their path. This means that if one makes a gesture and either gets no response or the wrong response, there will be only a little information available. Many touch-free devices currently available have various and unique gesture designs creating a learning curve for the users to overcome to use a device comfortably.

Constructing a universally accepted language of gestures will be challenging but will be beneficial in every section:

**UI designer**
A user interface will be as good as the way it is being interacted. With an accepted language of gestures, a UI designer can focus on designing the application around these gestures, which improves the interactivity and responsiveness of an interface.

**Corporation**
Even with huge investments by corporations in developing touch-free products, many have fallen short of achieving their objective mainly due to practicality and ease of use. These organizations always need to keep sight of the usability of a touch-free interface and it would incur even more costs. Universal gestures allow organizations to give more attention to developing and delivering products touch-free devices on large scale knowing that users can use their product without any learning curve.

**Users**
Universal gestures will allow users to interact naturally with a touch-free interface without any hesitation and responding to the visual cues to perform further action.

## 5 Designing touch-free gestures with sign language interpreters

Developing touch-free gestures will be more complex than the touch inputs which involves moving the fingertips across the surfaces of touch sensing displays or touchpads with scrolling, taps, or swiping gestures to perform relatively simple functions on the device. The current technologies offer the potential for improved user experience by improving productivity and ease of learning while reducing postural constraints.

Many implements of gestures for the touch-free interface have generally followed principles of natural language while considering the technical limitations of hand posture and gesture recognition. However, performing gestures frequently throughout the day will result in both fatigue and discomfort so we should consider physical effort when designing these gestures. While natural language and cognitive usability factors are critical for gesture selection, the selection process should also consider the discomfort and pain associated with the gesture. A model for the relationship between human cognitive and motor processes and computer gesture recognition for executing an HCI task is presented in Fig. 3.

On the human side, system performance is limited by the cognitive and motor processes used for gesture formation. On the computer, side performance is limited by the rate and fidelity of gesture capture and processing. The pace of gesture formation and external stress can cause variations in the hand postures formed.

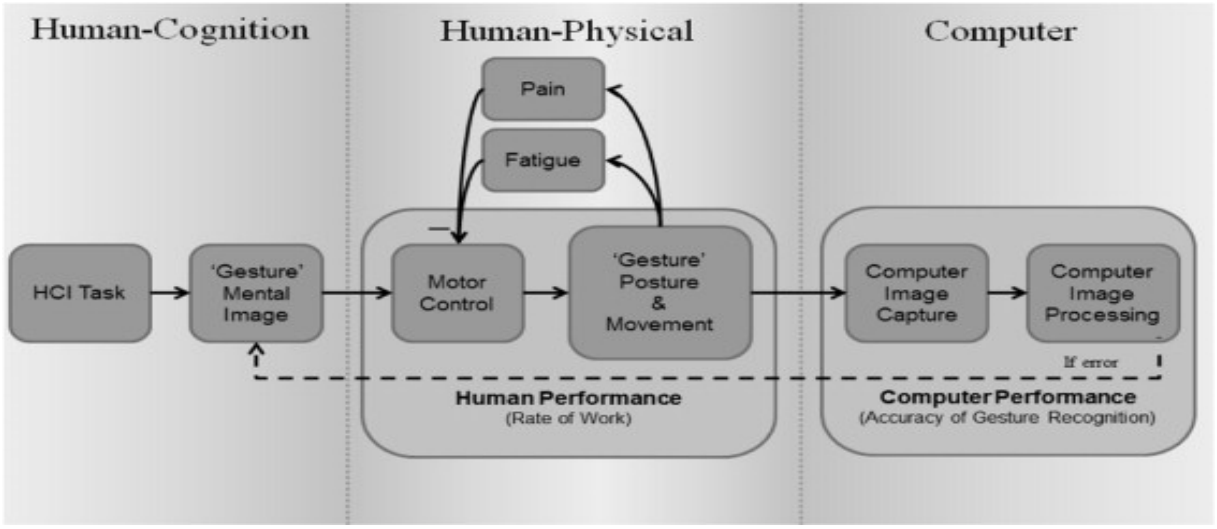


Fig. 3. *A model of the relationships between human cognitive and motor processes and computer gesture recognition for completion of an HCI task.*

The paper [3] presents a study to determine while hand gestures were comfortable to form and which ones are associated with hand pain when performed repeatedly by sign language interpreters. It showcases the ability of sign language interpreters to associate specific hand gestures to different levels of hand pain when the gestures are formed repeatedly.

Sign interpreters rate commonly used gestures, postures, and motions for discomfort level if repeated frequently during signing on a 6-point scale; 0= comfortable; 5= very uncomfortable or painful. Gestures include specific hand postures, motions, hand locations, size, and speed of movements. Mean discomfort rating for hand shapes, word signs, and postures were calculated.

UI designers can analyze the results presented in the paper to design comfortable gestures for the users. In the paper, the discomfort ratings are assessed for the location of hands during the signing, hand postures, and motions and finger postures. The most comfortable location for hands during the signing was at lower chest height and close to the body. The least comfortable area was at shoulder or face height or off to one side or the other with hands further from the centerline than the shoulders. Elbows bend to more than 90 degrees were uncomfortable

The mean ratings of discomfort/comfort for the gestures for the 37 alphanumeric characters could be grouped into 15 distinct discomfort/comfort levels and rank-ordered.

The findings from this study can be applied to design hand gestures for a touch-free interface in HCI. Tasks that are frequently executed such as next page, delete menu, return, paste, etc should be assigned to gestures that are comfortable to form repeatedly.

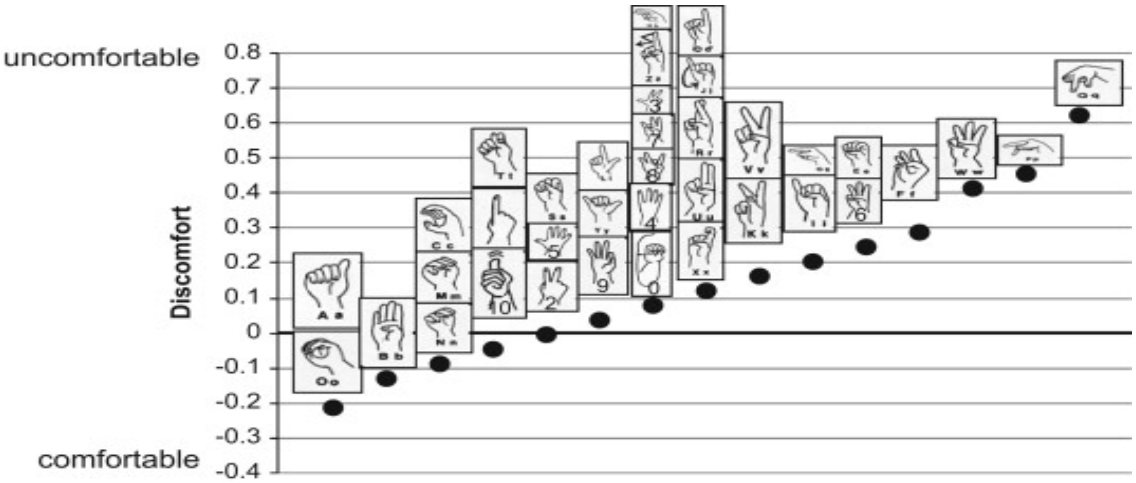


Fig. 4. *The rank order of 37 alphanumeric characters by mean discomfort (1)/comfort (–1) ratings.*

Based on this study, sign language interpreters can consistently differentiate hand gestures by the level of comfort, and designers who develop gestures should consider the experience of sign language interpreters in guiding the selection of comfortable gestures for HCI.

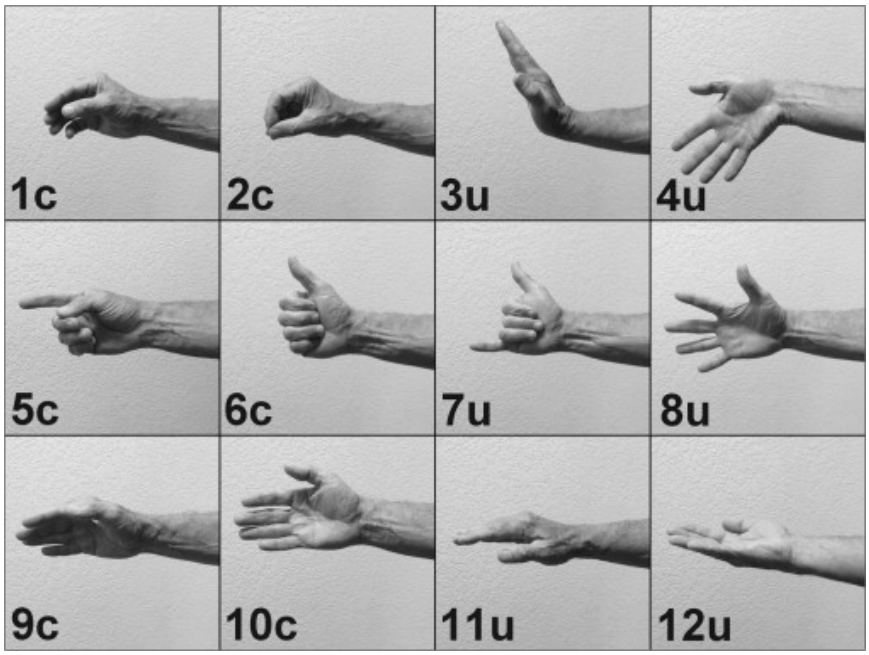


Fig. 5. *Examples of comfortable (c) and uncomfortable (u) hand postures: (1c) fingers slightly flexed; (2c) hand in a loose fist; (3u) halt sign with wrist and fingers extended; (4u) wrist in ulnar deviation and fingers extended; (5c) loose hand pointing; (6c) thumb up; (7u) shaka sign with discordant adjacent finger postures; (8u) fingers extended and abducted (spread apart); (9c) forearm rotation to 45 degrees pronation; (10c) forearm rotation in neutral; (11u) forearm rotation to full pronation; (12u) forearm rotation to full supination.*

## 6 Limitations with designing gestures

Even if we could develop a universal language of gestures, there are many limitations currently that need to be addressed to take the touch-free HCI mainstream. To be adopted, the technology has to fit in the context it is used in, and make the existing task simpler, faster, or cheaper than the current solution.

The current implementation of air gestures is not easier to use and sometimes doesn't make any sense. The air gestures implemented in phones are inconvenient to use since tapping the phone is easier than performing gestures at a distance.

Detecting smaller motion and mentally calibrating your hand motion will take time. On top of that, it can get annoying, when sensor measurements get glitchy. And it even requires a specific distance to interact which could take away the naturality of the interaction.

This shows that touch-free gestures did not match the need of the masses on any specific level and did not make a lot of tasks particularly faster or easier.

## 7 Conclusion

However, rapid progress and innovation in technology will allow the designer to create significantly better gestures and interface in the next iteration. We are only at the beginning of a new computer era where people will communicate with devices the way they do with each other. Gesture recognition UX offers fantastic opportunities to change the way we interact with computers.

The new design principles presented in this paper could help designers communicate touch interactions. Not limited by any axes on a screen, touch-free gestures allow designers to leverage depth and body movements. The case presented for a need for universal accepted gesture languages could refine the existing gesture recognition systems and usage. But certain limitations could prevent from making it happen since there isn't any regulatory body and every company has its implementations based on its hardware, software, user interface, etc.

Designers should be collaborating with the sign language interpreters for designing comfortable design gestures. But comfort is just one factor to consider when assigning hand gestures to HCI tasks. Technology is always improving and other factors such as cognitive, accuracy, reliability, and responsiveness. In conclusion, touchless interactions will bring in a fresh perspective on the human interaction paradigm and result in truly unique user experiences.

## Bibliography

[1] Vladimir I. Pavlovic, Rajeev Sharma, Thomas S. Huang. Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review 1997

[2] Davi Hirafuji Neiva, Cleber Zanchettin. Gesture recognition: A review focusing on sign language in a mobile context. 2018

[3] David Rempel, Matt J. Camilleri, David L. Lee. The design of hand gestures for human-computer-interaction: Lessons from sign language interpreters. 2014

[4] G. Manresa, J. Varona, R. Mas, F.J. Perales. Hand tracking and gesture recognition for human-computer interaction

[5] Kensy Cooperrider. " Gesture talks " from-pointing-to-nodding-is-gesture-a-universal-language.

[6] Ram Pratap Sharma, Gyanendra K. Verma. Human-Computer Interaction using Hand Gesture 2015.

[7] S. S. Rautaray and A. Agrawal, Vision-Based Hand Gesture Recognition for Human-Computer Interaction: A survey

[8] Deepak Naglot, Milind Kulkarni. Real-time sign language recognition using the leap motion controller

[9] Cao Dong, M. C. Leu, and Z. Yin, "American Sign Language alphabet recognition using Microsoft Kinect"

[10] Jacob Nielson. 10 Usability Heuristics for user interface design