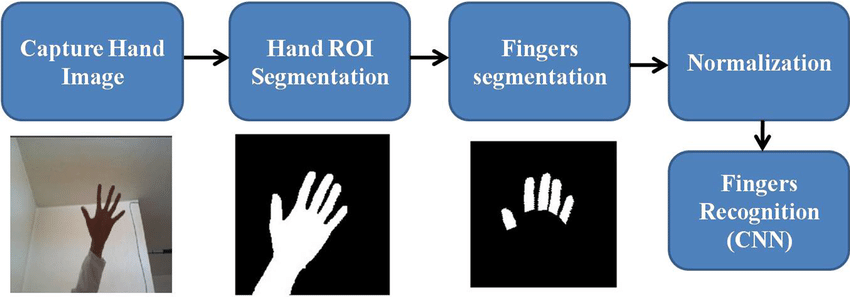
**Human hand Gesture Recognition in Human–computer interaction**

**1, Introduction**

Gesture recognition is a computing process that attempts to recognize and interpret human gestures through the use of mathematical algorithms. Gesture recognition is not limited to just human hand gestures, but rather can be used to recognize everything from head nods to different walking gaits.

A gesture recognition system starts with a camera pointed at a specific three-dimensional zone within the vehicle, capturing frame-by-frame images of hand positions and motions. This camera is typically mounted in the roof module or other vantage point that is unlikely to be obstructed. The system illuminates the area with infrared LEDs or lasers for a clear image even when there is not much natural light.



**Figure 1.** Proposed hand gesture recognition system

Proposed hand gesture

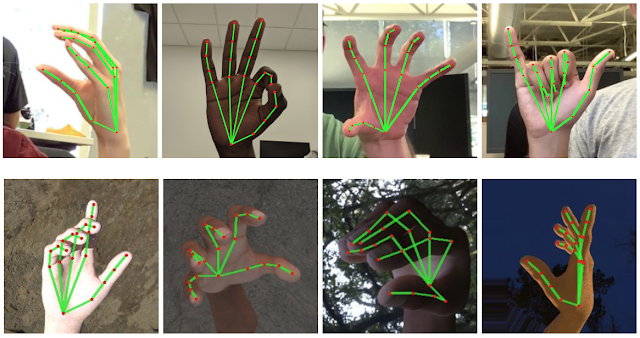
recognition syste

A camera feeds image data into a sensing device that is connected to a computer. The sensing device typically uses an infrared sensor or projector for the purpose of calculating depth,

Specially designed software identifies meaningful gestures from a predetermined gesture library where each gesture is matched to a computer command.

The software then correlates each registered real-time gesture, interprets the gesture and uses the library to identify meaningful gestures that match the library.

Once the gesture has been interpreted by computer vision and machine learning technologies, which translate the hand motions into commands, based on a predetermined library of signs, the computer executes the command correlated to that specific gesture.



**Figure 2.** Gesture Recognition in Machine Learning

Commands generated by the gesture recognition software become just another type of input, similar to turning a dial, pressing a button or touching a screen. Additionally, as the quantity and quality of cabin cameras improves, other passengers in the vehicle could eventually get in on the act.

**2, Design principles of touch-free gesture-based interfaces**

The next generation of UI can not blindly inherit the old principles of interaction design. Designers can take into consideration existing concepts, but they need to adapt them according to the new type of interaction. Here are a few important rules that you need to remember when working on gesture recognition UX:

**2.1, Avoid using the WIMP or touch-based models**

One of the common pitfalls that many UX designers fall into is using a mouse + keyboard model for the gesture-based interface. Designers often rely on WIMP (Window, icons, Menus, Pointers)  — a standard model for desktop apps—and replace the mouse pointer with a human finger. This model does not apply to gesture-based interfaces because it fails to account for natural human motions. Since it’s based on using cursors, it will make the interaction uncomfortable for a person.

It’s also not recommended to apply touch-based paradigms to hands-free design. What works for touch may not necessarily work hands-free design.

**2.2, Make interaction comfortable**

Making interaction with UI comfortable should be a top priority for designers. Since users will interact with their arms, you need to ensure that users’ arms don’t tire quickly from having to interact with UI. Here are a few things to remember:

* **Make individual gestures comfortable for the user.**Consider the human body ergonomics when creating UI. If a gesture is uncomfortable or too repetitive, the experience won’t be great for users, and there’s a high chance that the user will abandon the product.
* **Avoid gestures that require a lot of physical work.**When interacting with UI, a user has to do a lot of movements (especially ones that involve gesturing with hands above your heart), that can quickly become annoying. Games and physical exercises are an exception to this rule. When users interact with Nintendo Wii a lot of movement can be really positive.
* **Take user session into account.** Gesture control interfaces are great for short periods, but they quickly fail under long timelines.

**2.3, Design intuitive gestures**

Gestures are hidden controls, and this can cause problems for UI designers. Back in 2010, Don Norman drew attention for the problems : “Because gestures are ephemeral, they do not leave behind any record of their path, which means that if one makes a gesture and either gets no response or the wrong response, there is little information available.” Still, in 2019, we don’t have a universally-accepted language of gestures that we can rely on when designing interfaces.

The learning curve for interactions can be problematic for gesture-driven interfaces. That’s why it’s recommended to only use intuitive gestures so users don’t have to learn a special gesture language.

Here are a few simple tips for you:

* **Borrow gestures from real life.** Try to emulate sign language or borrow gestures from it to perform actions. Observe how users naturally move their arms and hands and introduce these patterns in your UI.
* **Avoid using complex gestures.** Users don’t want to memorize combinations to make an action.

**3, Virtual Environments**

A virtual environment is a computer generated environment aimed to provide a userwith the experience of being in a simulated place. This is usually achieved by blockingstimuli from the “real” world and replacing them with synthetic ones. It can be described as the feeling of being partof the virtual environment or becoming unaware of the physical surroundings. Sensoryfeed back allows a presentation based on a user’s position and depends on input devicesas well as output hardware. Interactivity refers to the ability of a user to interact insome way with the artiﬁcial surroundings or the objects in this virtual world.

Typical examples would be navigating through the environment, moving objects or initializing simulations. All of those elements eventually have an eﬀect on the presence, the senseof actually being in the virtual environment. Obviously, ﬁdelity of sensory feedback and the level of interactivity can increase the feeling of immersion and the sense of presence. On the other hand, inaccurate, cumbersome or unfamiliar devices and displays can also distract a user from the environment which inﬂuences immersion and presence negatively.The term virtual reality (VR) is commonly used synonymously to virtual envi-ronments. However the deﬁnition of VR diﬀers since it is relatively new. It can mean the ﬁeld of study which deals with creating and improving the synthetic experience described earlier, but also the technology or medium used to create this experience or even the experience itself. At ﬁrst, this chapter introduces the basic components of virtual environments contributing to the key elements mentioned above. Afterwards, virtual reality soft-ware frameworks are discussed, because they are extensively used to design virtual environments and the interaction within it.

**4, Drawbacks and possible solutions**

Gesture-based interfaces have many advantages and provide the user with a completely new form of interaction. However, this kind of input also raises issues that are not relevant with traditional input. On the user's side, these problems are to learn, to remember and to accurately execute gestures. The developer has to provide a system that correctly recognizes these gestures. Therefore, the developer not only has to ensure that gestures are quickly and correctly recognized, but also has to provide a guide that allows a rapid and easy learning of these gestures.

The teaching of multi-touch and mid-air gestures is more difficult than that of single-touch gestures. In the case of the latter, the hand posture is irrelevant - users only need to follow a path correctly to perform a command. But with an extension to multi-touch and mid-air gestures, the position and movement of several fingers or even the whole hand becomes relevant. Teaching systems usually instruct the user about the necessary hand movement and path for a gesture rather than the posture and form of contact, focusing on commands that can also be performed with a single-touch input device like a mouse or a pen.

**4.1, Discoverability**

A disadvantage with gestures, is the fact that they are neither self-revealing nor self-explanatory. A named button on a toolbar has an explicit purpose and is also easy to find, gestures, however, may be arbitrary and are usually more difficult to discover.

In order to solve this problem, Bau and Mackay (2008) proposed OctoPocus, a dynamic guide that combines feedforward and feedback mechanisms. After a press-and-wait gesture, a map of all possible gestures, visualized through coloured templates, is displayed around the current cursor position. As the user begins to follow a path, the other paths become progressively thinner, indicating that they're less likely to be recognized, until they disappear.

A solution that is also suitable for multi-touch input is the ShadowGuides system. A so-called user-shadow visualizes the user's input, giving feedback on what parts of the hand are in contact with the surface. The user shadow annotations demonstrate possible gestures available from the current hand pose, the registration pose guide informs the user about alternative registration poses.

Another, a little different approach is GestureBar. While the aforementioned learning guides employ the "learning-by-doing" technique, GestureBar separates the learning area from the user's document and discloses information about a gesture only if needed. The system works like a traditional toolbar - the user can click an item to find details about the execution of the command and to test it in an experimental area.

**4.2 Memorability**

While conventional commands only have to be recognized, gestures need to be known and remembered before executing them.

One possibility to create memorable gestures is to make them as intuitive as possible, as they are more likely to be remembered that way. Wobbrock et al. researched these natural gestures and found that although there are common features used by nearly all of the participants, gestures are far from being "obvious" and that it is difficult to design a gesture set that feels natural for every user. People often used reversible gestures to achieve two opposing effects and used more fingers for moving larger objects, mirroring their experiences in the real world. They were also strongly influenced by their knowledge of traditional computers, using gestures that could also be performed with a mouse (even tapping their fingers as if clicking it) and locating the "Close" gesture at the top-right corner of objects as if they were using a Windows PC.

Another aspect to remember is the fact that the concept of intuitiveness strongly depends on culture and experience. Many mid-air gestures used in everyday life strongly differ from country to country - a nod, for example, will be commonly interpreted as an indication of agreement, but there are some countries, like Greece for instance, where it stands for the exact opposite. Another example is the pinch-to-zoom gesture that will come natural to every regular smartphone user, but not to someone who has never seen a touchscreen.

An alternative to intuitive gestures are the so-called Marking Menus that combine named commands in a pie menu and gestures. That way, they ease the transition from novice- to expert-mode usage. A further development of Marking Menus are the Augmented Letters by Roy et al., where the user activates a pie menu by drawing the letter the elements start with. After that, he can choose from the menu by extending the gesture.

**4.3 Fatigue**

Gestures normally involve more muscles than other interaction techniques (Baudel & Beaudouin-Lafon, 1993), and especially mid-air gestures, but also gestures that require muscle tension and complex movements over a long period of time can be very exhausting. Therefore, developers should design gestures that are quick and comfortable to execute.

One approach are the so-called Microgestures - tiny gestures that can even be executed during other activities and therefore allow true multitasking. A possible area of usage is driving, where small tasks like changing the volume of the radio can thus be performed without the potential risk of releasing the steering wheel. This idea has of course been already partially implemented with the integration of additional control elements in the steering wheel. But microinteractions could also be incorporated in other fields of our everyday life, like when writing with a pen or holding a cash card.

Another aspect, addressed by Forlines and others that needs to be considered is the fact that many bimanual interactions in the real world are asymmetric, with the non-dominant hand being slower and less precise than the dominant hand. Therefore, gestures need to be equally easy for left- and right-handed users and should not demand too much of a person's non-dominant hand. (Forlines, Wigdor, Shen & Balakrishnan, 2007)

**5, Usability of gesture recognition system**

USABILITY FACTORS:

Usability is a term that has been defined by different authors and in different standards, but to classify the usability factors found in the literature three attributes included in the ISO 9241-11, the international standard on Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs):

* Effectiveness refers to task performance; how accurately and completely did the user achieve the goals?
* Efficiency is the amount of effort that is required to achieve the level of effectiveness when achieving the goals. Efficiency is the relationship between effectiveness level and resource consumption.
* Satisfaction refers to how comfortable the user feels while using the system.

When designing a gesture for an interface, it is important to consider **two different aspects**.

On the one hand the application’s context and the user’s requirements.

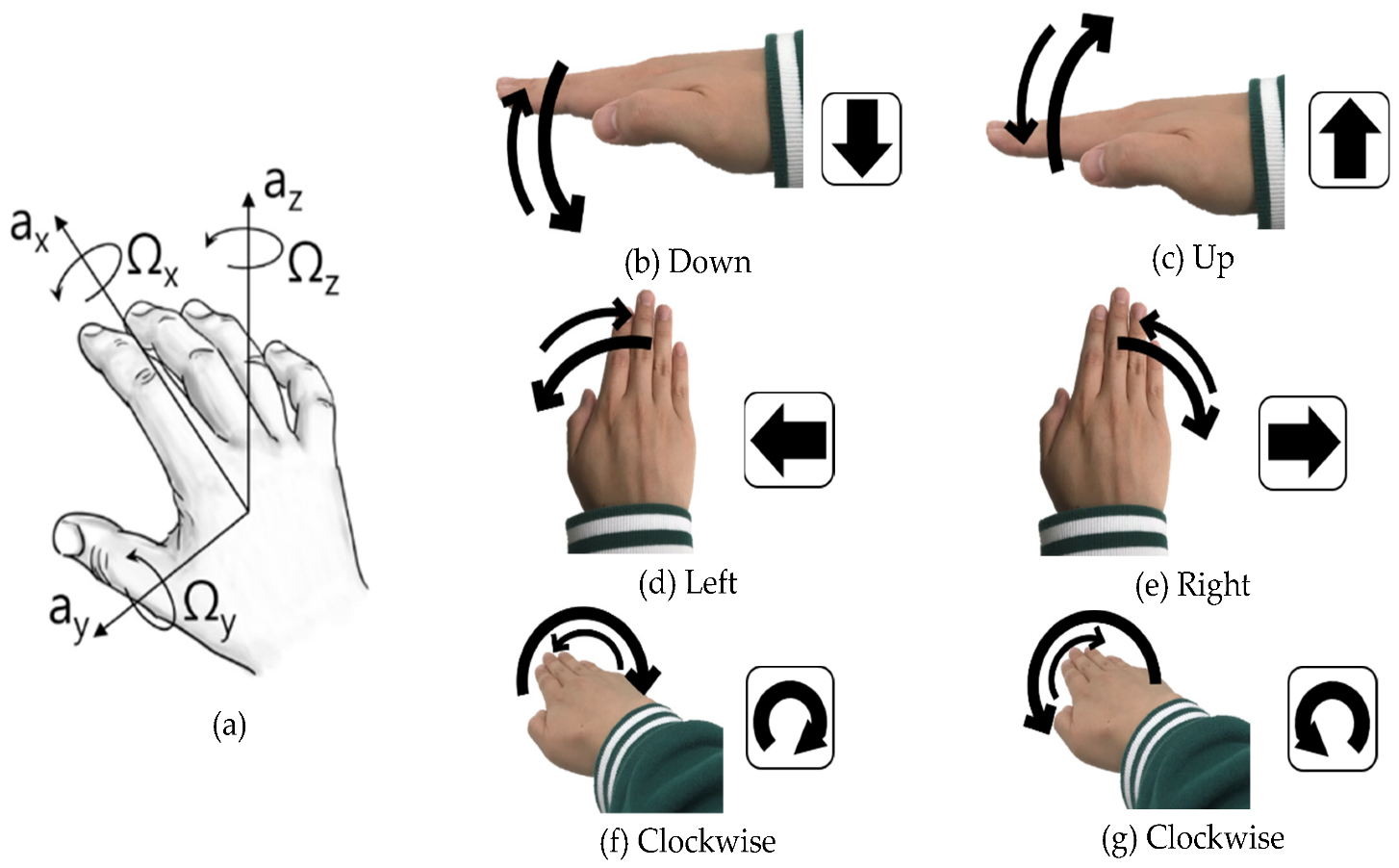
And on the other hand, Lenman characterize gestures along three dimensions: cognitive, technological and articulatory:

* Cognitive aspects are related with how easy a command can be learnt and recalled.
* Technological aspect refer to need of taking into account the state-of-the art of technology, now and in the near future when designing the command set for gestural interaction based on computer vision. To exploit the use of gestures in HCI it is necessary to provide the means by which they can be interpreted by computers [18].
* Articulatory aspects take into consideration that a gesture has to be comfortable to ensure that a physically stressing gesture is avoided taking into account the human anthropometrics .

These evaluation will be classified into effectiveness, efficiency and satisfaction and the metrics used to measure them.

**5.1 Accuracy and Error rate**

Accuracy is the correctness in recognizing the gestures performed by the user. This factor is related with the robustness and precision of the computer vision techniques. This factor is frequently tested and is an indicator of the gesture uniqueness, that is, if a gesture is similar to another one, the system can misinterpret it and trigger a wrong action.



**Figure 3.** Six unit hand gestures selected

Errors are related with the accuracy and we can distinguish two kinds of errors:

* Misinterpreted gestures within the set of possible gestures, which would be related with the uniqueness.
* Gestures that are not understood, which could be related with the robustness of the computer vision techniques.
* A gesture that is not recognized requires repetition, but a gesture which is misrecognized needs to be corrected. In this case, both parameters could be evaluated individually.

A metric frequently used to evaluate accuracy is to control the number of correct recognized gestures regarding the total number of performed gestures and the accuracy for each gesture was measured using:

**5.2 Efficiency**

When assessing efficiency, different measurements are used such as the user’s physical and mental effort, duration of the gesture or memorability/learnability.

5.2.1 Physical fatigue

When interacting with body movements, fatigue or tiredness can appear, especially if the gesture is physically demanding. Gestural commands must be concise, fast and avoid gestures that require a high precision over a long period of time in order to minimize effort.

Fatigue is a difficult attribute to measure because is userdependent. The metrics that have been used are usually questionnaires using Likert-scales or Borg CR10 scale (which is a method of rating perceived exertion.

5.2.2 Duration

The gesture duration is how long the user needs to perform the gesture. A gesture involves a preparation, an execution and retraction phase. The execution phase sometimes requires maintaining the gesture for a predefined time for robustness.

Duration is strongly related with the computer vision techniques, as a minimum or maximum duration may have been set or can be configured to recognize a gesture. This factor affects the system’s efficiency and the user’s fatigue: the longer the user needs to perform the gesture, the fatigue can increase and lesser input commands to the system can be performed.

This factor is evaluated by measuring the time between the start and the end of a gesture. In a gesture interface to control a media centre is tested, and to assess the duration they manually computed the time between the start and the end of a gesture. Moreover, many works evaluate this factor by computing the duration of carrying out a task, instead of the actual duration of the gesture, especially when comparing different interfaces.

5.2.3 Cognitive load

Cognitive load refers to the total amount of mental activity imposed on working memory.

Another approach to evaluate cognitive load is by analyzing the quantity of information that an individual can remember while using that interface. This latter test is based on the idea that when individuals are forced to use working memory or other cognitive resources, information is lost or displaced.

5.2.4 Learnability and Memorability

Learnability, or time to learn, is the time and effort required reaching a specific level of use performance. Memorability, or retention over time, is the ease of system intermittently for casual users. The closer the syntax of the operations match the user's understanding the easier it will be to remember how to operate the interface. If the time to learn is fast, then the retention will be less important .

**5.3 Satisfaction**

Most of the factors classified under the satisfaction attribute are

evaluated by using user questionnaires to capture the subjective

users’ feelings towards the interface.

5.3.1 Comfort

Comfort is defined as a pleasant feeling of being relaxed and free from pain.

This factor is frequently assessed by subjective questionnaires Many works focused on improving this attribute by identifying comfort zones.

*(****For example****, a method for objective assessment of postural comfort, where postural comfort is defined as a posture that does not elicit compensating motion of other body parts, is presented. The authors analyze the user’s posture to define a comfort zone.)*

5.3.2 Ease of use

Ease of use, easy-to-use or easiness, means that the user needs little effort to operate with the system.

The easier the interface is, the fastest the user will bring out a profit. In order to improve the easiness of a VBI, the design should take into account the previous user experience and the design should be familiar and consistent with the users expectations. This factor can be evaluated by means of subjective questionnaires.

5.3.4 User experience and Satisfaction of use

The user experience includes both pragmatic (efficiency and effectiveness attributes) and hedonic aspects of the system, measured through subjective indicators such as user satisfaction and hedonic quality (fun, aesthetics). Hedonic quality is the extent to which a system allows for stimulation by its challenging and novel character or identification by communicating important personal values.

**REFERENCES**

1, Graetzel, C., Grange, S., Fong, T., and Baur, C. 2003. A Non-Contact Mouse for Surgeon-Computer Interaction. NCCR-COME Research Networking Workshop, Brauwald, Switzerland.

2, Tollmar, K., Demirdjian, D. and Darrell, T. 2004. Navigating in virtual environments using a vision-based interface. In Proceedings of the third Nordic conference on Humancomputer interaction (NordiCHI '04). ACM, New York, NY, USA, 113-120.

3, Norman, D.A. and Nielsen, J. 2010. Gestural interfaces: a step backward in usability. Interactions 17(5) (September 2010), 46-49.

4, Manresa-Yee, Cristina, Esperança Amengual, and Pere Ponsa Asensio. "Usability of vision-based interfaces." In *Actas del XIV Congreso Internacional Interacción Persona Ordenador*, pp. 113-118. 2013.

5, Manresa-Yee, C., Varona, J., Mas R. and Perales, F.J. 2005. Hand Tracking and Gesture Recognition for HumanComputer Interaction. Electronic Letters on Computer Vision and Image Analysis 5(3), 96-104.

6, Wickeroth, D., Benölken, P. and Lang, U. 2009. Manipulating 3D Content using Gestures in Design Review Scenarios. International Journal of Information Studies, 1(4), 242-250.

7,<https://www.researchgate.net/publication/340104477_An_efficient_method_for_human_hand_gesture_detection_and_recognition_using_deep_learning_convolutional_neural_networks>

8, <https://how2electronics.com/gesture-recognition-application-machine-learning/>