# cfHMI: A Novel Contact-Free Human-Machine Interface

Tobias Rehrl, Alexander Bannat, Jürgen Gast, Gerhard Rigoll,
and Frank Wallhoff

Department of Electrical Engineering and Information Technology,
Institute for Human-Machine Communication
Technische Universität München
80290 Munich, Germany

**Abstract.** In this paper we present our approach for a new contact-free Human-Machine Interface (cfHMI). This cfHMI is designed for controlling applications – instruction presentation, robot control – in the so-called "Cognitive Factory Scenario", introduced in [1]. However, the interface can be applied in other environments and areas of application as well. Due to its generic approach, this low-cost contact-free interface can be easily adapted to several independent applications, featuring individual menu-structures, etc. In addition, the modular software architecture facilitates the upgrades and improvements of the different software modules embedded in the cfHMI.[1]

## 1 Motivation

Our objective for the implementation of this contact-free Human-Machine Interface is the following: Due to appearing demands and constraints as a consequence of the industrial surroundings in the "Cognitive Factory Scenario", a new form of a HMI was required. The current state-of-the-art of HMI [2,3,4] is not well suited for the factory design encountered in the "Cognitive Factory Scenario".

In general, the present and in particular the future forms of industrial assembly require a holistic approach to tackle the challenges appearing in the interaction between humans and developing (cognitive) robots. Therefore, the cfHMI can be a module for a more pleasant interaction, because of its flexibility and adjustability to the production task as well as to the current working step. Additionally, it provides for the certain amount of safety and secure handling needed in industrial factory surroundings. A modular architecture concept is a further objective of the cfHMI, because this kind of concept relieves the software-module development and improvement. Besides, the in this paper introduced cfHMI features three major underlying conceptual foundations: flexibility in presentation, secure handling of system operations and adaptability in the displayed menu-structure.

---

[1] All authors contributed equally to the work presented in this paper.

J.A. Jacko (Ed.): Human-Computer Interaction, Part II, HCII 2009, LNCS 5611, pp. 246–254, 2009.

## 2 Hardware Setup

As mentioned above, the here introduced cfHMI requires only a small amount of hardware, this characteristic facilitates the integration of this system in various surroundings. A typical standard PC is sufficient for executing all vision-based computations for the cfHMI. A simple webcam delivers the raw video data as input for the image-processing steps. For this purpose, a webcam is mounted above the workbench to survey the menu interaction scene, besides a projection unit is attached in the vicinity of the webcam. Via this projection unit the interaction menu is displayed on the workbench. The borders of the projected interaction menu are constituted via two black markers. These markers have a rectangular form, which is beneficial for the vision-based processing steps.

## 3 System Overview

Our approach for the cfHMI is based on a modular concept, where different software modules process different tasks. Following this concept, it is necessary to define a clear and simple interface for these software modules. However, in turn, a huge advantage of this concept is that these software modules can be readily replaced with a more sophisticated and improved version, without any further modifications at the remaining software modules. In addition, with this approach a high-degree of flexibility and portability of the system to other areas of application comes along.

We have subdivided the system into two major components: menu-design and image-processing.

### 3.1 Menu-Design

The design of the interaction menu of the cfHMI is splited into two domains. In the first domain, the structure of the menu and basic depiction features are defined. The second domain is concerned with the underlying logic of the functions incorporated in the interaction menu.

#### Menu Structure

The menu structure of the cfHMI can be defined with an XML-based configuration file. Generally, the main characteristics of the menu are set up in the XML-file: color, amount of interaction buttons, label-text and associated function calls.

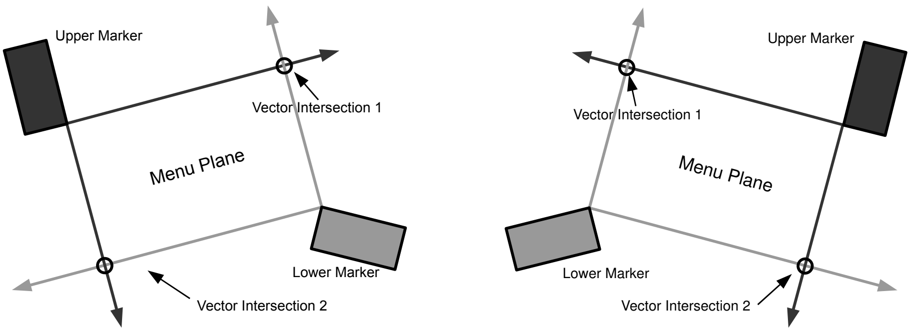
This choice of XML to design the structure of the menu has the following reasons: First, XML has a simple and coherent structure, and thus the creation and adaption of the menu structure can be easily implemented. Second, the syntax underlying in the XML is coming up to meet a menu structure with its different menu points and sub-menus.

**Menu Functions**
Different menu functions (e.g. timers) are implemented in the computer programming language python. Because of its simplicity and reduced syntax, the multi-paradigm programming language python is used to implement varying menu functions. A further point for the implementation of the menu functions in python is the reason, that python supports OpenCV [5] constituting the foundation for the implementation of the image-processing operations.

## 3.2 Image-Processing

Many tasks for the cfHMI are situated in the visual domain, therefore, OpenCV is used for the image-processing, because many standard and high-level computer vision techniques are implemented in this open source library. As a starting point for the image-processing, the two black markers have to be located. Besides, the hand of the user has to be detected and the interactions with the projected menu have to be interpreted to trigger the corresponding menu commands. Another problem treated with OpenCV is the transformation of the positions of the two black markers into suitable coordinates for the projection unit.

**Marker Localization**
As mentioned above, two black rectangular markers constitute the indicators for the boundaries of the display region of the menu. In Figure 1, a schematic setting of the markers and the resulting menu plane is depicted. The before mentioned markers are indicated as "Upper Marker" and "Lower Marker". As a starting point, a simple thresholding operation is performed to exclude image elements, having a brightness above a certain limit. After this pre-segmentation step, a typical filter operation embedded in OpenCV, the famous Canny Filter [6] is applied to detect the edges and corners occurring in the image, delivered of the top-down-view camera monitoring the interaction scene from above. As the subsequent step, the obtained contours from the filter operation are evaluated. A-priory knowledge about the characteristics of the two markers – geometric



(a) Alignment Sample 1 (b) Alignment Sample 2

**Fig. 1.** Menu Plane

features – assists in the verification process of possible marker candidates constituted by the gained contours. After completion of the pre-processing steps, only the two markers are visible in the processing image and the marker localization can begin.

At first, the two centers of gravity of the two markers are determined. Afterwards, for each marker the inner border point is extracted. If the upper marker is left of the lower marker: The upper border of the menu is the lower right corner of the upper marker, and the lower border is the upper left corner of the lower marker, see Figure 1(a). Vice-versa, if the upper marker is right to the lower marker: The upper border of the menu is the lower left corner of the upper marker, and the lower border is the upper right corner of the lower marker, see Figure 1(b). However, only two corner points of the interaction menu are determined by the markers. Thus, the remaining corners have to be computed. For this computation, the angle of the principal component of the upper marker is necessitated.

To calculate the missing two corners of the menu, two simple vector intersections need to be computed, marked as "Vector Intersection 1" and "Vector Intersection 2" in Figure 1.

### Coordinates Transformation

For a contact analog displaying of the cfHMI, it is necessary to establish a relation between the coordinates of the image obtained by the webcam and the table projector coordinates. Therefore, a serial projection of three points is used to calculate the mathematical dependencies between the two different coordinate systems to acquire the homogeneous transformation matrix $H$. The reference points are statically set up to cover the entire workspace in the webcam image. In the next step, these points are displayed with the table projection unit. Afterwards in the calibration phase of the system, the user has to select the projected points in the webcam image. The webcam coordinates are stored into an internal vector array.

After having clicked on all three points, the mathematical dependencies between the projector and the webcam coordinate systems are then calculated and as a result, the transformation matrix $H$, containing information about the rotation $R$, translation $T$ and some shearing $S$ between both systems, is stored into a text file. A fourth acquired point is afterwards reverse calculated and then displayed to be able to verify the gained resulting precision of the transformation process.

### Hand Localization and Hand Interaction

For the hand localization process, the four obtained corner points of the interaction menu are utilized to set up a region of interest (ROI) and therefore reducing the computational requirements as well as decreasing false positives. In the next step in this ROI, a skin-color filter method [7,8] is applied to extract possible hand candidates. Afterwards, non-linear filter operations are used to remove noise and little speckles in the resulting filter mask. We made the following constraint, that all hand-based interaction within the ROI have to be performed orthogonal to the horizontal borders of the cfHMI (e.g. we assume that the pointing finger is parallel aligned to the vertical borders). In practice, these constraints are neither disturbing the overall usage of the system, nor reducing the joy of use.

For reducing the complexity and enhancing the reliability in the hand gesture extraction process, we introduced a rotation invariant working image (riwi). This so-called "riwi" is obtained by performing a rotation via the principal component axis angle of the upper marker around the center of the interaction menu on the resulting image of the filter operation.

Having the hand information available, the hand gestures can be analyzed depending on the current state of the interaction menu. All available gestures are depicted in Figure 2 and are explained in the following. In the starting process and to unlock the interaction menu, a closed hand (see Figure 2(a)) has to hover above the interaction menu. For activating menu functions and menu navigation, a pointing gesture (see Figure 2(b)) has to be performed. A moving hand (see Figure 2(c)) from the right vertical border to the left vertical border will trigger the menu locking, if the gesture is performed rectangular to the verticals and has a certain velocity. The first and the second hand gesture interaction are basing on a template matching algorithm, whereas the third gesture utilizes optical flow for the detection process [9].

(a) Closed hand for unlocking the menu.

(b) Pointing gesture.

(c) Wiping gesture.

**Fig. 2.** Hand gestures to be recognized by the presented system

## 4 Menu Interaction

In general, the menu interaction process between the user and the cfHMI can be subdivided into three main operations: initialization / unlocking, navigation / function control and locking.

As a requirement for the menu interaction, two black markers are needed. Both markers constitute the limiters for the display region. To introduce the required flexibility in displaying the menu, the corner-points – set-up by the above described markers – can be adapted in real-time. With this feature, it is possible to relocate the displayed menu within the region observable by the top-down-view camera and also covered by the projection unit. As a further system-characteristic, the size of the menu can be on-line rescaled – within a certain allowed surface size, predetermined by the layout of the designed menu-structure – by simply moving the markers freely on the active surface. By analyzing the principle component of the anchor marker (”Upper Marker”), a rotational angle for the displayed menu can be set, cf. Section 3.2.

### 4.1 Initialization / Unlocking

The first step for enabling the interaction with the cfHMI is the following: The closed right hand (see Figure 2(a)) has to hover above the starting screen for a certain amount of time, i.e. two seconds, to unlock the menu. This procedure has also to be performed by the user, if the menu is in the locked state. The reasons for this initialization and unlocking procedure are the following: To avoid unintended interaction and to reduce the amount of information currently available on the workbench (cf. screen saver).

### 4.2 Navigation / Function Control

The pointing gesture (see Figure 2(b)) has to be performed by the users hand to activate the navigation in the interaction menu as well as to enable/disable menu functions. The exact position for the selection of a menu field is determined by the tip of the right index finger. The selection of the menu function incorporated by a certain field is confirmed via two feedback channels: acoustical and visual. A short audio signal provides the user a feedback for activation of a certain field, besides this field is also highlighted by color change of the borders of the field.

### 4.3 Locking

The locking operation is applied to set the menu from the active interaction mode to standby mode. Here, we use the evaluation of the optical flow above the cfHMI to trigger this function (see Figure 2(c)). In addition, this procedure is also accompanied by audio feedback. The purpose for this operation is comparable to the unlocking procedure. The user can actively choose to deactivate the system interaction and therefore avoid unintended usage of the presented cfHMI (cf. keypad locking in cellular phones).

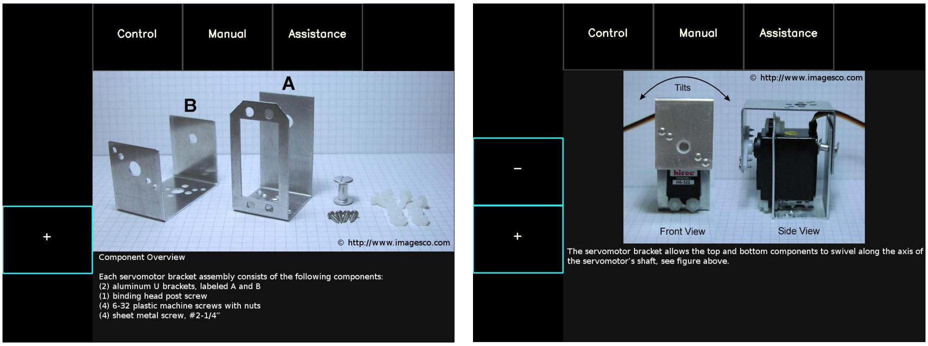
Furthermore, in context of an industrial application, a worker using this presented system could accidentally activate functions in an unsafe situation, e.g. by laying his hand on the active surface while performing his assembly task, therefore, a further safety-feature is implemented. For preventing such activations, the system is equipped with an automatic keylock-procedure: If the unlocked and operational system remains inactive for a certain amount of time, the cfHMI will switch to the initial locked state.

## 5 Area of Application

Up to now, two different areas of application have been developed for cfHMI. The first area of application is an interface for a robot control (see Figure 3(a)). With

(a) Menu Overview

(b) Robot Control Menu



Component Overview

Each servomotor bracket assembly consists of the following components:
(2) aluminum U brackets, labeled A and B
(1) binding head post screw
(4) 6-32 plastic machine screws with nuts
(4) sheet metal screw, #2-1/4"

The servomotor bracket allows the top and bottom components to swivel along the axis of the servomotor's shaft, see figure above.

(c) Instruction Menu Overview

(d) Instruction Menu 1. Step

**Fig. 3.** Overview of the Projected Interaction Menu

this unit, it is possible to communicate simple control commands to the robot. The second area of application is an assisting presentation of the instruction for a hybrid assembly, where the human manufactures in cooperation with an industrial robot a given product. This hybrid assembly resembles the setup of the CoTeSys Project *Joint Action of Humans and Industrial Robots (JAHIR)* presented in [10].

## 5.1 Robot Control

The industrial robot manipulator arm controlled by the cfHMI is a Mitsubishi robot RV-3SB. The robot manipulator arm has six degrees of freedom and can lift objects with a maximum weight of three kilograms. A radius of 0.642 m around the robot's trunk composes the accessible and manipulable workspace of the robot. Its tool point is equipped with a force-torque-sensor and a tool-change unit.

The current version of the robot control unit is capable to communicate four different commands towards the robot (see Figure 3(b)). It is possible to control the robot arm in four directions (up, down, left, right), however, this set of commands can be easily extended.

The instantaneous communication between the robot and cfHMI is implemented via a direct network access of the menu computer on the robot control unit.

### 5.2 Assembly Instruction

In the context of an industrial manual assembly station, the here presented cfHMI was also integrated. In this scenario, the purpose for the menu interaction system is to display assembly instructions, as depicted in Figures 3(c) and 3(d). The advantage of the cfHMI is the flexibility in the location and scale of the menu. With this setup, the worker can decide where to project the assembly instructions (location as well as rotation). The cfHMI is used as a browser interface towards the assembly instructions, where the instruction steps can be shifted through forwards as well as backwards. Furthermore, the user can set up the degree of displayed information, allowing him to choose the preferred amount of details for the current assembly step.

## 6 Conclusion and Future Work

A first technical realization of the cfHMI has been integrated in a scenario occurring in the "Cognitive Factory". The system has not achieved its ultimate state, and there are still some domains, where improvements can be made. Nonetheless, the presented approach offers high potential to be integrated in a new form of a multimodal human-robot interaction scenario as an additional input modality.

At the current state, the interaction with cfHMI is limited to button selection and simple gestures. As a next step, it is imaginable to incorporate a large set of hand gestures for the menu interaction. In addition, it is thinkable that to a certain degree the trajectory of the human hand can be repeated by the robot arm for "Programming by Demonstration".

A further starting point for improvement is the integration of the cfHMI software-architecture into the real-time database, see [11,12,13]. This framework, as foundation for the cfHMI, facilitates the integration in a multimodal interaction scenario.

## Acknowledgment

This ongoing work is supported by the DFG excellence initiative research cluster *Cognition for Technical Systems – CoTeSys*, see `www.cotesys.org` for further information and details.

We want to thank all our partners within our CoTeSys-Projects for the fruitful discussions and implementation work to make our visions and ideas become reality.

## References

1. Zäh, M.F., Lau, C., Wiesbeck, M., Ostgathe, M., Vogl, W.: Towards the Cognitive Factory. In: Proceedings of the 2nd International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV), Toronto, Canada (July 2007)
2. Jaimes, R., Sebe, N.: Multimodal human computer interaction: A survey, pp. 15–21 (2005)
3. Goodrich, M.A., Schultz, A.C.: Human-robot interaction: A survey. Foundations and Trends in Human-Computer Interaction, 203–275 (2007)
4. Chouvardas, V.G., Miliou, A.N., Hatalis, M.K.: Tactile display applications: A state of the art survey. In: Proceedings of the 2nd Balkan Conference in Informatics, pp. 290–303 (2005)
5. Bradski, G., Kaehler, A.: earning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Press, Sebastopol (2008)
6. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. 8(6), 679–698 (1986)
7. Soriano, M., Huovinen, S., Martinkauppi, B., Laaksonen, M.: Skin Detection in Video under Changing Illumination Conditions. In: Proc. 15th International Conference on Pattern Recognition, Barcelona, Spain, pp. 839–842 (2000)
8. Vezhnevets, V., Sazonov, V., Andreeva, A.: A survey on pixel-based skin color detection techniques. In: Proc. Graphicon 2003, pp. 85–92 (2003)
9. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision, pp. 674–679 (1981)
10. Lenz, C., Nair, S., Rickert, M., Knoll, A., Rösel, W., Bannat, A., Gast, J., Wallhoff, F.: Joint Actions for Humans and Industrial Robots: A Hybrid Assembly Concept. In: Proc. 17th IEEE International Symposium on Robot and Human Interactive Communication, Munich, Germany (2008)
11. Goebl, M., Förber, G.: A real-time-capable hard- and software architecture for joint image and knowledge processing in cognitive automobiles. In: Intelligent Vehicles Symposium, pp. 737–740 (June 2007)
12. Stiller, C., Färber, G., Kammel, S.: Cooperative cognitive automobiles. In: Intelligent Vehicles Symposium, pp. 215–220. IEEE, Los Alamitos (2007)
13. Thuy, M., Göbl, M., Rattei, F., Althoff, M., Obermeier, F., Hawe, S., Nagel, R., Kraus, S., Wang, C., Hecker, F., Russ, M., Schweitzer, M., León, F.P., Diepold, K., Eberspächer, J., Heißing, B., Wünsche, H.-J.: Kognitive automobile - neue konzepte und ideen des sonderforschungsbereiches/tr-28. In: Aktive Sicherheit durch Fahrerassistenz, Garching bei München, April 7-8 (2008)