

## Research Article

# 3D Assembly Group Analysis for Cognitive Automation

**Christian Brecher,<sup>1</sup> Thomas Breitbach,<sup>1</sup> Simon Müller,<sup>1</sup> Marcel Ph. Mayer,<sup>2</sup>  
Barbara Odenthal,<sup>2</sup> Christopher M. Schlick,<sup>2</sup> and Werner Herfs<sup>1</sup>**

<sup>1</sup>Laboratory for Machine Tools and Production Engineering (WZL), RWTH Aachen University, 52074 Aachen, Germany

<sup>2</sup>Institute for Industrial Engineering and Ergonomics, RWTH Aachen University, 52062 Aachen, Germany

Correspondence should be addressed to Thomas Breitbach, th.breitbach@wzl.rwth-aachen.de

Received 15 July 2011; Accepted 8 November 2011

Academic Editor: Ivo Bukovsky

Copyright © 2012 Christian Brecher et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A concept that allows the cognitive automation of robotic assembly processes is introduced. An assembly cell comprised of two robots was designed to verify the concept. For the purpose of validation a customer-defined part group consisting of Hubelino bricks is assembled. One of the key aspects for this process is the verification of the assembly group. Hence a software component was designed that utilizes the Microsoft Kinect to perceive both depth and color data in the assembly area. This information is used to determine the current state of the assembly group and is compared to a CAD model for validation purposes. In order to efficiently resolve erroneous situations, the results are interactively accessible to a human expert. The implications for an industrial application are demonstrated by transferring the developed concepts to an assembly scenario for switch-cabinet systems.

## 1. Introduction

One of the effects of globalization in public view is the reduction of production in high-wage countries especially due to job relocation abroad to low-wage countries, for example, towards Eastern Europe or Asia [1–3]. Based on this, a competition between manufacturing companies in high-wage and low-wage countries typically occurs within two dimensions: value-orientation and planning-orientation. Possible disadvantages of production in low-wage countries concerning process times, factor consumption and process mastering are compensated by low productive factor costs.

In contrast, companies in high-wage countries try to utilize the relatively expensive productivity factors by maximizing the output (economies of scale). Another way to compensate the arising unit cost disadvantages is customization or fast adaptation to market needs (economies of scope), even though the escape into sophisticated niche markets does not seem to be a promising way for the future anymore.

Within the dimension planning-orientation companies in high-wage countries try to optimize processes with sophisticated, investment-intensive planning approaches, and production systems while value-orientation offers the benefit of

shop floor-oriented production with little planning effort. Since processes and production systems do not exceed the limits of an optimal operating range, additional competitive disadvantages for high-wage countries emerge.

In order to achieve a sustainable competitive advantage for manufacturing companies in high-wage countries with their highly skilled workers, it is therefore not promising to further increase the planning orientation of the manufacturing systems and simultaneously improve the economies of scale. The primary goal should be to wholly resolve the so-called polylemma of production, which is analyzed in detail by Klocke [4]. Economies of scale and economies of scope must be maximized at the same time, while additionally the share of added-value activities must be further maximized without neglecting the planning quality. Therefore, according to the “law of diminishing returns” a naive increase in automation will likely not lead to a significant increase in productivity but can also have adverse effects. According to Kinkel et al. [5], the amount of process errors is in average significantly reduced by automation, but the severity of potential consequences of a single error increases disproportionately. These “Ironies of Automation” [6] which were identified by Lisanne Bainbridge as early

as 1987 can be considered as a vicious circle [7], where a function that was allocated to a human operator due to poor human reliability is automated. This automation results in higher function complexity, finally increasing the demands on the human operator for planning, teaching, and monitoring, and hence leading to a more error-prone system.

In order to break the cited vicious circle, one essential step is the application of cognitive control mechanisms by means of simulation of human cognition within the technical system. Such cognitive production cells can generally be understood as a further development of autonomous production cells. Admittedly, autonomous production cells only possess limited abilities in self-optimization and self-adaptation to changing production tasks. These abilities are the fundamental approach of cognitive production cells and are currently one challenge in research and development [8]. Based on these functions, the concept of cognitive automation was introduced by Onken and Schulte in 2010 [7]. However, their original concept was strongly influenced by the research field of unmanned vehicles. The corresponding concept of the “cognitive plant” by Zaeh et al. [9] transfers the cognitive approach onto production systems. This concept successfully integrates cognitive mechanism into manufacturing systems, but the superior subject of using cognitive modules including the “human factor” as an operator and surveillant, however, still remains unexplored. Based on artificial cognition, technical systems shall not only be able to (semi)autonomously perform process planning, adapt to changing manufacturing environments or objectives, and be able to learn from experience, but also to simulate goal-directed human behavior and therefore significantly increase the conformity with operator expectations. Within this focus, a highly debated issue is the software architecture of a cognitive system. For this purpose, various architectures were proposed as a basic framework for the simulation of cognitive functions [10, 11]. Herein, a popular approach is the three-layer model with a cognitive, an associative, and a reactive layer of regulation [12, 13]. Comparative structures can be found within the Collaborative Research Centre 614 “Self-optimizing concepts and structures in mechanical engineering” (CRC 614) [14] as well as within the “cognitive controller” at the Technical University of Munich [15, 16]. Further broad researches within the field of cognitive technical systems can be found in Onken and Schulte [7] and, with special focus on the production environment, in Ding et al. [17]. Herein, the implementation of cognitive abilities within security systems for plant control focused. In this context, especially the safety of human-machine interaction and safety at work are taken into account. Additional concepts and methods can be found within the automotive sector as well as within space and aeronautic research [18, 19].

Within the Cluster of Excellence “Integrative Production Technology for High-Wage Countries” at RWTH Aachen University, a Cognitive Control Unit (CCU) and its ergonomic user-centered human machine interface are developed for a robotized production unit [20–23] which partially transfers non-value-adding planning and implementation tasks of the skilled worker to the technical cognitive system. The validation and interaction with the human expert

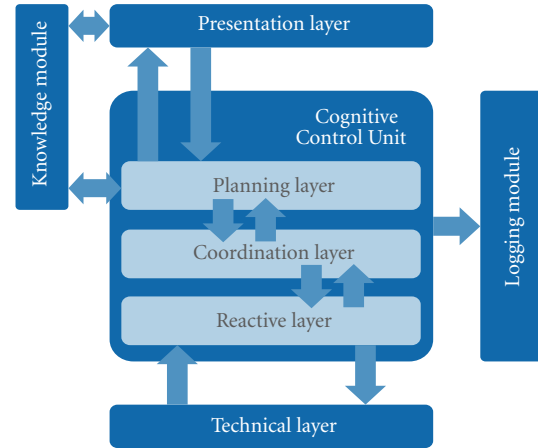


FIGURE 1: Concept of the cognitive architecture.

requires a technical recognition system which is able to measure the current state of the environment and provide feedback upon that information. This system needs to be integrated into the software architecture of the CCU.

## 2. Cognitive Control Unit and Evaluation Scenario

Due to the fact that the CCU is a system, which needs to be comprised of several different software components, the design required a modular structure. It allows for an integration and enhancement through distributed software modules.

**2.1. Cognitive Architecture.** The software architecture of the CCU is separated into five layers and shown in Figure 1. The presentation layer incorporates the human machine interface and an interface for editing the knowledge base. The planning layer is the deliberative layer in which the actual decision for the next action in the assembly process is made. The services that the coordination layer provides can be invoked by the planning layer to start action execution. The reactive layer is responsible for a low response time reaction of the whole system, for example, in order to efficiently respond to emergency situations. The knowledge module contains the necessary domain knowledge of the system in terms of production rules.

**2.2. Human Operator.** In regard to the role that humans play in standard automated production, the main task involves managing and monitoring the manufacturing system. In the advent of malfunction, they must be able to take over manual control and return the system to a safe, productive state. This concept, termed “supervisory control” by Sheridan [24], involves five typical, separate subtasks that exist in a cascading relationship to one another: plan, teach, monitor, intervene, and learn.

After receiving an (assembly) order, the human operator’s first task usually involves planning the assembly process.

To do so, he or she must first understand the functions in the relevant machine and the physical actions involved to be able to construct a mental model of the process. Using this basic understanding, the operator then develops a concrete plan that contains all specific subtargets and tasks necessary. “Teaching” involves translating these targets and tasks into machine-readable format—for example, NC or RC programs—which allow for a (partially) automated process. The resulting automation must be monitored to ensure that it runs properly and generates products of the desired quality. The expectations for the process are drawn from the mental model the operator created at the start. In cases where reality significantly deviates from this model or where there are anomalies, the human operator can intervene for example, by modifying the NC or RC program or by manually optimizing the process parameters. Ultimately, every intervention involves the human operator continually adapting his/her mental model, while existing process information, characteristic values, and trend analyses help the operator to better understand the process and develop a more detailed mental model.

With a cognitively automated system, the tasks change gradually, but in a conceptually relevant way. In this system, the human operator defines the assembly tasks based on the status of the subproduct or end product, carries out adaptations or sets priorities as needed, compiles rough process plans, and sets initial and boundary conditions. The information-related stress on the human operator is considerably reduced in the areas of detailed planning and teaching, since they are handled by the cognitive system. But shifting this load from the human to the machine can result in the human operator forming an insufficient mental model of the state variables and state transition functions in the assembly process. In order to ensure the conformity with the operator’s expectations during the supervision of the assembly process [25], the first step is the use of motion descriptors to plan and execute the assembly process, since motions are familiar to the human operator from manually performed assembly tasks [26]. Therefore, the Methods-Time Measurement (MTM) system as a library of fundamental movements was chosen [25, 27]. Even though the sequence of fundamental movements (e.g., reach, grasp, move, position, release) is explainable a posteriori, the sequence of parts positioned after another is not predictable a priori due to a lack of elaboration knowledge [22]. Odenthal et al. [28] and Mayer et al. [29] identified human assembly strategies that were formulated as production rules. When the reasoning component is enriched with these human heuristics, a significant increase of the robot’s predictability when assembling the products can be achieved.

Further, if an error occurs which the system cannot identify or solve, the human operator must receive all information relevant to the situation in an easily understandable form so that he/she can intervene correctly and enable system recovery.

**2.3. Assembly Scenario.** To test and develop a CCU in a near-reality production environment in a variety of different assembly operations, a robotic assembly cell was set up [21].

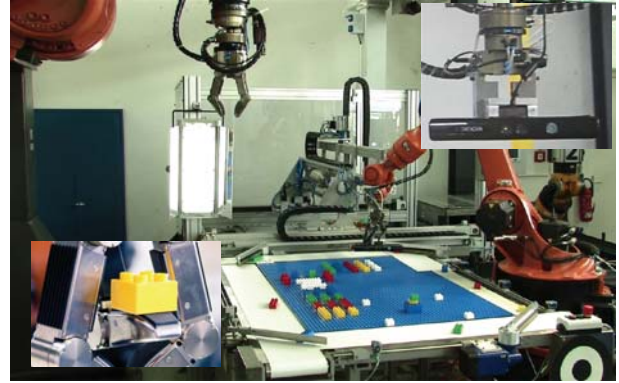


FIGURE 2: Assembly cell.

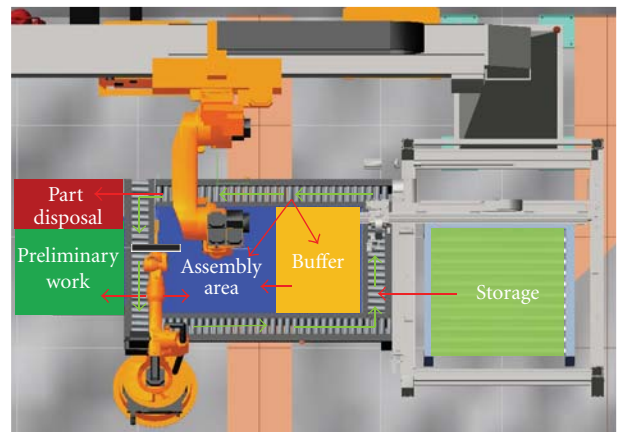


FIGURE 3: Assembly and storage areas in the assembly cell.

The layout of this cell is shown in Figure 2. The scenario was selected to address major aspects of an industrial application (“relevance”) and at the same time to easily illustrate the potential of a cognitive control system (“transparency”) [30].

**2.3.1. General Setup of the Assembly Cell.** The main function of the demonstrator cell is the assembly of predefined objects. Part of the cell is made up of a circulating conveyor system comprising six individually controllable linear belt sections. Several photoelectric sensors are arranged along the conveyor route for detection of components. Furthermore, two switches allow components to be diverted onto and from the conveyor route. Two robots are provided, with one robot travelling on a linear axis and carrying a tool (a flexible multifinger gripper) and a color camera. Several areas were provided alongside the conveyor for demand-driven storage of components and as a defined location for the assembly (see Figure 3). One area is provided for possible preliminary work by a human operator. This is currently separated from the working area by an optical safety barrier. The workstation has a multimodal human-machine interface that displays process information ergonomically, allowing it to provide information on the system state as well as help for solving problems, if necessary. To simultaneously achieve

a high level of transparency, variability, and scalability in an (approximate) abstraction of an industrial assembly process, building an assembly of Hubelino bricks was selected as the assembly task. These are in size and shape very similar to LEGO Duplo bricks. To take into account the criterion of flexibility for changing boundary conditions, the bricks are delivered at random. In terms of automation components, the system consists of two robot controllers, a motion controller, and a higher-ranking sequencer.

The initial state provides for a random delivery of required and nonrequired components on a pallet. A FESTO handling system successively places the components onto the conveyor. The automatic-control task now consists of coordinating and executing the material flow, using all the technical components in a way such that only the assembled product is in the assembly area at the end.

*2.3.2. Actions and Sequences.* The assembly scenario is as follows. An engineer has designed a mechanical assembly of medium complexity by composing it, for example, with a CAD system containing any number of subcomponents. The human operator assigns the desired assembly goal to the cognitive system via the presentation layer (see Figure 1). The desired goal is transferred to the planning layer where the SOAR-based reasoning component derives the next action based on the actual environmental state (current state on the conveyor, the assembly area, and the buffer) and the desired goal. The environmental state is based on the measured vector from the sensors in the technical application system (TAS). In the coordination layer the raw sensor data is aggregated to an environmental state. The next best action derived in the planning layer is sent back to the coordination layer, where the abstract description of that action is translated into a sequence of actor commands which are sent to the TAS. There, the sequence of commands is executed and the changed environmental state is measured again by the sensors.

*2.3.3. Motivation for Assembly Group Analysis.* The last step, an image-based recognition process of the assembly object's state in the assembly area, is focused on this contribution and is described in detail later on. If the current state differs from the target state, the human operator is informed so that he/she can detect and correct occurred errors.

Generally four types of errors are possible, when positioning a brick in the assembly group:

- (1) It might occur during assembly that a brick is not placed in the assembly group at all. This is the case, for example, when the gripper loses the brick during the transportation from conveyor to assembly area.
- (2) A generated assembly sequence might not be correct and a brick is placed at a false location. In practice, this error has never occurred, but its existence needs to be considered.
- (3) The brick is placed at the correct position, but not fitted properly. This error case refers to possible tolerances for both the brick and the position of the

robot. For example, in situations where a brick has to be positioned between two other bricks, the accuracy of the robot's position might not be sufficient. For the most part this leads to a part lying on top of the assembly board rather than being assembled onto that board.

- (4) At last it is also possible that a brick has correctly been placed and fitted at the right location—but it was the wrong brick, for example, a blue brick instead of a green one. These errors are mostly related to the image processing system, which detects the moving part on the conveyor, failing at high conveyor speeds.

It can be stated that in practice most of the errors that occur during the course of the assembly are related to the failure of some sort of component. In order to efficiently interact with a human expert, the type of error needs to be identified. Since error type 1 should be reflected in the data as a whole set of missing data points and for error type 3 only as a small shift at a specific location, it becomes apparent that the reliability of this classification varies for the different error types.

The human operator must be supported with more information in case of an assembly error during the operation, for example, if the image-based control of the assembly step leads to a deviation between the current and the target state. Under this assumption, a first prototype of a supporting system was developed dealing with the task of error identification in an assembly object (incorrect construction of the assembly object). More precisely in this case, a prototype of an augmented vision system (AVS) was developed and implemented with the focus on the presentation of the assembly information. The aim of using this system was to place the human operator or skilled worker, respectively, in a position to detect the construction errors in a fast and adequate way. Therefore, a laboratory test was carried out in order to investigate different display types and different modes of visualizable assembly information from an ergonomic point of view [23, 31, 32]. Within a second step, the AVS was extended to assist the human operator in the disassembly of the erroneous object in order to correct the detected error in cooperation with the robot [33].

While a detailed overview of the cognitive control can be found in Kempf [30], there was still the need for an automated verification of the assembly group. In the following, the technical recognition of an assembly group within the assembly area of the cognitively automated assembly cell is described in detail.

### 3. Recognition Process

In this scenario, individual assembly groups are established which consist of an arbitrary combination of different basic elements. For demonstration, the actual assembly groups consist of Hubelino parts which differ in color (yellow, orange, red, light green, dark green, and blue) and length. The length of each part varies from 32 mm to 192 mm in steps of 32 mm. Each element has a width of 32 mm and a height of 16 mm.

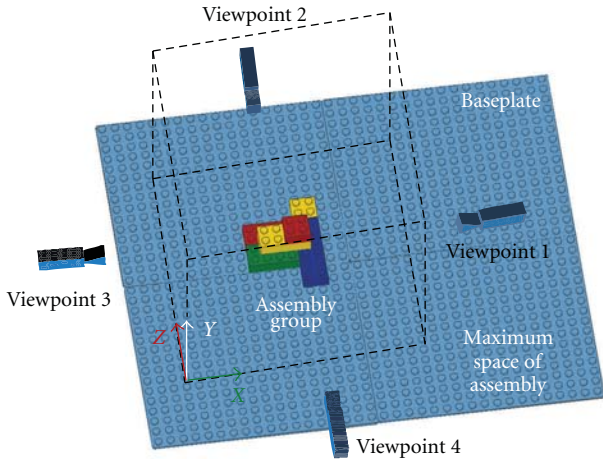


FIGURE 4: General design of the recognition process.

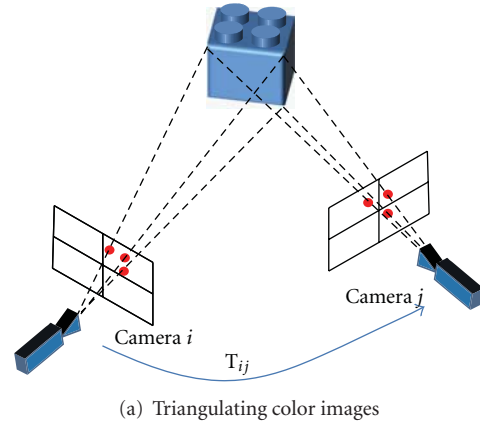
Since the Hubelino parts are plugged onto the baseplate, respectively, onto each other, their positions are defined by the round shapes on top of each Hubelino part (see Figure 4). Each Hubelino part is laminated with a glossy surface that has a high light-reflecting coefficient.

**3.1. Technical Recognition Systems.** Within the recognition process, the assembly has to be checked if it is constructed correctly. Therefore, the assembly group is analyzed from four different viewpoints using a contactless 3D measurement system. Figure 4 shows the general design of the scenario. The dashed cuboid defines the maximum space of assembly.

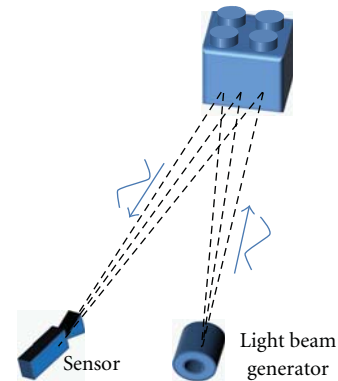
At each position the Hubelino brick, that is present in the real assembly group, potentially differs in color as well as in size from the corresponding part defined in the virtual model. Thus information about these two properties needs to be measured for all of the possible assembly positions.

**3.1.1. Contactless Measurement Methods.** In the past few years, contactless 3D measurement systems have become an important tool for quality control. The two most used contactless measurement methods [34] are (i) triangulating the image data with several color images, (ii) time-of-flight method.

Within the first method, a scene is inspected from several viewpoints and identical “landmarks” inside each color images are identified (e.g., edges or corners). The pixel position of those landmarks combined with the camera’s intrinsic calibration (relation between a pixel position and the directions relative to the camera center) yields in the vectors pointing from the camera center to the direction of each landmark. Determining this vector in at least two camera images and considering the relative position between these cameras, the special position of this landmark can be triangulated (compare Figure 5(a)). For this measurement method, at least two RGB cameras are necessary as well as an accurate spatial transformation  $T_{ij}$  between the cameras. The disadvantages of this technology are the huge computing



(a) Triangulating color images



(b) Time-of-flight method

FIGURE 5: Principle of both contactless 3D measurement techniques.

power necessary to calculate several high-resolution color images as well as finding concurrent and unique landmarks in each image. Consequently, this measurement method fails in image regions with a structured surface. Details concerning this technology are described in [34].

The second widespread contactless measurement method is the time-of-flight technique (see Figure 5(b)). Here, the time that a frequency-modulated beam of light needs to reach an object and to get back to the sensor is measured indirectly by comparing the phase of the emitted light with the phase of the received signal. For that reason, this method requires a device emitting the light beam as well as a light sensitive sensor detecting the reflected light beam. Accordingly, this method depends on the light-reflecting coefficient of the measured surface. Hence, objects with a low reflectivity coefficient (i.e., windows) are not detected as well as objects with a very high reflectivity coefficient (i.e., bright surfaces).

Widely used types of application for this technology are 2D laserscanners. Within 2D-laserscanners, a single light beam is diverted by a mirror. Concerning the angular position of the mirror and the time of flight of the single light beam, the depth values of a line can be measured. By panning the 2D-laserscanner, a complete 3D depth image of the environment can be computed. Recently, time of flight sensors are available measuring the depth values not only of

TABLE 1: Sensor requirements.

Requested attribute for assembly group analysis	Available for the 3D time-of-flight method	Available for triangulating color images
<i>Depth image and color image available</i>	Only depth information available	Both supported
<i>High accuracy of the depth image</i>	About 1 cm	Depends on resolution and calibration accuracy; recently ca. 1 mm
<i>Depth image independent of the surface</i>	Depends on light-reflecting coefficient	Fails in region with structured surfaces
<i>Mapping between color and depth image</i>	Not available	Available for each camera
<i>High resolution of the depth image</i>	Up to $320 \times 240$ Pixels	Equals the number of identified landmarks
<i>Dependency on ambient light</i>	Independent	Fails if scene is under/overexposed

a single point or line, but also of a complete matrix. Hereby, a 3D depth image is measured directly without requiring mechanical parts for panning the sensor [35, 36].

Concerning the recognition process, both described measurement methods have disadvantages. For the recognition both the color and the exact spatial dimensions of the assembly are considered. Therefore a color image as well as an accurate 3D depth image and the mapping between the two are required. In Table 1, the requirements for analyzing an assembly are faced with the characteristics of both contactless measurement methods.

According to Table 1 the main disadvantages of the 3D time-of-flight method are that it does not meet the requirements in terms of accuracy for depth data and that it does not provide a color image. However, compared to the triangulation method, it provides a higher measurement frequency and is independent of ambient light.

Still, both measurement methods are not completely independent of the surface of the measured objects. On the one hand, the triangulation method cannot detect the depth values of structured surfaces, while on the other hand the time-of-flight method relies on the reflectivity of the surface.

However, the actual measurement task requires the detection of the color and the exact spatial dimensions of small devices with a completely structured and glossy surface. Hence a combination of both measurement methods is required.

**3.1.2. The Kinect as a Recognition Device.** Within the field of 3D measuring, the Kinect sensor, developed by Microsoft and PrimeSense, was presented to the public. Initially developed for game consoles, this device combines a 3D depth sensor with a resolution of  $640 \times 480$  pixels and a standard color camera with a resolution up to  $1280 \times 1024$  pixels. Additionally, a microphone array, a position sensor which measures the vector of gravitation, and an electrical motor for tilting the unit are integrated. Table 2 gives a short overview of the relevant specifications [37]. Combining the measured data of those devices, the Kinect sensor provides multiple innovative opportunities for research and development within the field of environment recognition as well as in the field of man-machine interaction [38].

The Kinect's 3D depth sensor combines aspects of both contactless 3D measurement methods described above, whereby the scene is continuously illuminated with infrared structured light [39]. Structured light measures a 3D scene

TABLE 2: Specifications of the Kinect.

Device	Specifications
<i>Color sensor</i>	32 bit RGB $1280 \times 1024$ at 15 fps
<i>Depth sensor</i>	16 bit Depth $640 \times 480$ at 30 fps
<i>Depth sensor range</i>	650 mm–3500 mm
<i>Spatial Resolution (at 2 m distance)</i>	$x/y$ resolution: 2 mm, $z$ resolution: 1 cm
<i>Field of view</i>	Horizontal 57 degree, vertical 43 degree
<i>Tilt range</i>	27 degree
<i>Microphones</i>	2 microphones, each 16 bit audio at 16 kHz

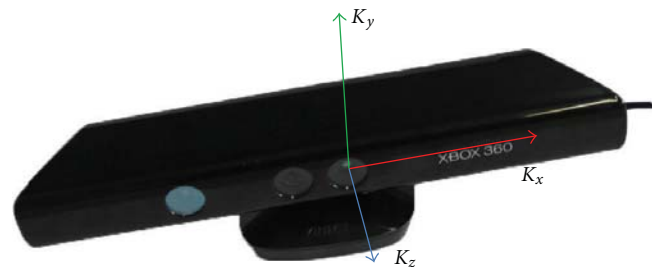


FIGURE 6: Coordinate system defined by OpenNI.

by projecting a known pattern of light onto the environment and recording it with a standard camera. The way this pattern deforms when striking surfaces allows calculating the depth information of the objects in the environment. Song [40] provides a detailed description of the computations. The Kinect projects a matrix of single IR dots onto the scene and provides a depth image with an accuracy of 1 cm at 2 m distance.

In order to accomplish the assembly group analysis, only depth and color information of the Kinect are merged. Communication and data exchange with the Kinect sensor are realized with the official driver from PrimeSense and its modifications for OpenNI. OpenNI uses a right-hand coordinate system for processing the depth data (see Figure 6).

As the Kinect depth sensor illuminates the room with an “IR Light Coding” and evaluates the reflected image,

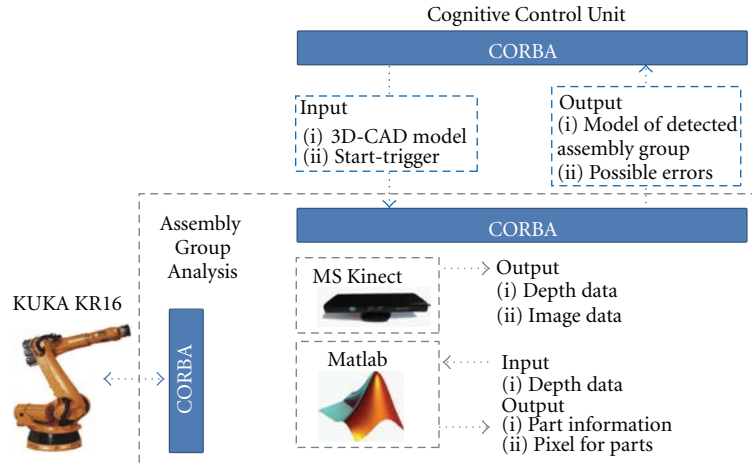


FIGURE 7: Software structure of the AGA.

the efficiency depends on the reflectivity coefficient of the measured surface. Additionally, the correct operation of the Kinect depth sensor relies on the incidence angle between the emitted light beam and the measured object. If the incidence angle is undersized, the light beam cannot be reflected back to the sensor with a sufficient intensity to be detected. The other way round, an incidence angle, which is nearly 90 degrees on an object with a high reflectivity coefficient, reflects the light beam with a very high intensity. This combination results in a cross-talk between adjacent pixels, whereby a blind spot within the Kinect's depth data occurs. As the blind spot occurs only under the described conditions, these need to be taken into consideration when the Kinect is used for analyzing an object. However, since the Kinect allows for a seamless integration of both depth and color data, it presents an appropriate device for the analysis of an assembly group.

**3.2. Assembly Group Analysis (AGA).** In order to create a perceived model, the assembly group is observed from multiple perspectives. For the generation of this model two approaches will be presented. In the first approach only general assumptions regarding the structure of a Hubelino model will be used. This approach is able to create a perceived model for any Hubelino model without previous knowledge about the bricks that should have been assembled, but is only valid in Hubelino scenarios where the general assumptions can be applied. A second approach takes advantage of the given customer-defined CAD data to derive recognizable patterns. On the one hand this approach is not able to construct a perceived model, if no CAD data is present. On the other hand it eliminates the general assumptions and thus is applicable not only in Hubelino scenarios, but also in any scenario, provided the condition that CAD data exists. Both approaches include the perception of both depth and color data and perform error detection. The superior goal of verifying a real assembly group of Hubelino bricks is achieved by comparing the perceived model with the virtual model from the customer-defined CAD file. The AGA is implemented as

a separate software component within the framework of the Cognitive Control Unit.

**3.2.1. Software Structure of the AGA.** In order to communicate with multiple components without having to implement a new interface for every new component, an abstract interface was used, which is able to transparently connect multiple devices. This abstract interface uses the CORBA Middleware and thus provides a real-time connection between devices and the CCU. This leads to an architecture, where the commands that the CCU issues are translated into CORBA calls that are directed to a specific device, in this case the AGA component. Hence the AGA component needs to provide interfaces to the CCU, which allow for a comprehensive communication. Since Matlab is a common tool for data visualization and the computation of complex algorithms, it was used to perform the computationally intensive tasks. A complete overview of the resulting architecture and their connections is provided in Figure 7.

In order to be able to verify the correct composition of an assembly group, the AGA component needs to know what the correct model looks like. Thus the CCU needs to submit a CAD file of the assembly group's expected state to the analysis component. This communication takes place for every verification, since it is possible for a model to change over time. The CAD model is given as an LDRAW file, which is an appropriate file format for the assembly of Hubelino parts [41]. After the analysis is completed, the AGA provides feedback to the presentation layer of the cognitive control framework (see Figure 1). This feedback contains the detected parts within the assembly group as well as possible errors. This allows for the CCU to present the results in the presentation layer, hence enabling a human operator to take action based on the received results.

In order to calculate a 3D model with a single sensor, the sensor needs to change its position relative to the object to examine all sides. Consequently, the Kinect sensor is attached to the flange of the KUKA KR 16 which can explore the assembly from an arbitrary position. Once the CCU requests a new

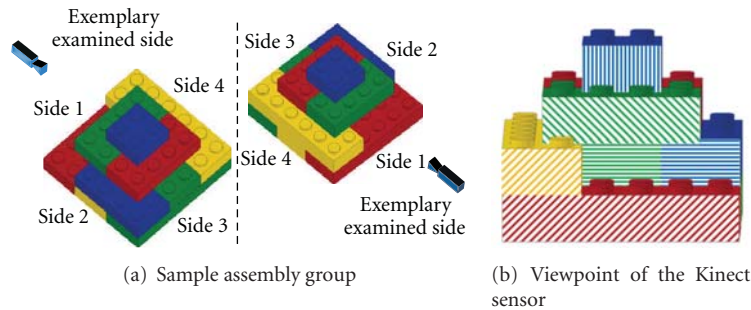


FIGURE 8: Assembly group.

analysis, the AGA component creates a connection to the Kinect as well as to the KUKA KR16 via CORBA. Within the analysis process, the assembly is examined from four different positions (compare Figure 4). Each viewpoint is aligned perpendicular to the particular side of the assembly with the intention of avoiding perspective masking of single Hubelino parts. For each viewpoint, the assembly group is analyzed and a discrete model for each side is calculated. Afterwards, these four discrete models are combined into one complete model of the assembly group.

**3.2.2. Data Acquisition.** The following description of the recognition process is based on the assembly group as shown in Figure 8. For demonstration purposes, side 1 is examined.

Figure 8(a) displays the exemplarily used assembly from two different points of view. Figure 8(b) represents the same assembly but presented from the viewpoint of the Kinect sensor. Concerning the Kinect's viewpoint, the Hubelino parts constitute planes parallel to the Kinect's  $x$ - $y$ -layer. Those planes are marked by different striped patterns.

For the data acquisition, the positioning of the viewpoints is essential, thus the sensor characteristic needs to be taken into account. As described above, the Kinect sensor fails if the emitted light beam is reflected perpendicularly by an object with a high reflectivity coefficient. Within the given task, all objects to be recognized have a high reflectivity coefficient. Therefore, each viewpoint needs to fulfill the following conditions:

- (1) The distance to the measured object has to be greater than 650 mm.
- (2) The light beams should not be reflected perpendicularly.
- (3) The particular side of the assembly group needs to be examined perpendicularly to avoid perspective masking of single Hubelino bricks.

Consequently, each viewpoint is at least 700 mm away from the assembly group and the assembly group itself is about 200 mm beneath the center of the depth image. Figure 4 shows their position and chronology. This choice of viewpoints has another important effect: in those viewpoints, the baseplate does not reflect the emitted light beams, according to the undersized angle of incidence. Hence the

baseplate will not be recognized by the Kinect, whereby the effort of interpreting the measured depth values is reduced.

After defining the viewpoints, the depth values need to be acquired. Thus, the maximum dimensions of the assembly group are limited by a virtual cube and only depth values within this cube are used for identifying the assembly (see Figure 4 dashed cuboid).

**3.2.3. Prefiltering.** Before starting to fit Hubelino parts, the depth data needs to be filtered. The first step is to identify the assembly group within the scatter plot. Within the defined cube, some parts of the baseplate or some object not belonging to the assembly, respectively, may be perceived by the Kinect sensor. Those objects disturb the recognition of the assembly group and need to be filtered.

Therefore, the data points are classified by their distance to each other. If the distance between two points is larger than a specified value, both points get a different classification. Otherwise they will be sorted into the same class. The distance for separating different scatter plots is 100 mm.

After passing this filter step, the data points are separated into their particular scatter plots. Based on the notion that the assembly group is the largest object within the defined cube, only the class with the most members is used for fitting the Hubelino parts. The result is shown in Figure 9. From the incoming measurement data (Figure 9(a)) only the red marked points are used for fitting the Hubelino parts (Figure 9(b)).

As described, the Hubelino parts can only be positioned in discrete places and the viewpoint of the sensor is perpendicular to each side of the assembly group. Hence the measured depth values are positioned on planes parallel to the sensors  $x$ - $y$ -plane (compare Figures 8(b) and 9(b)). Since all sensors are disturbed by noise, the depth values are not measured exactly and the planes of the assembly group need to be reconstructed.

**3.2.4. Identifying Planes.** For this task the RANSAC algorithm is used and optimized for finding planes. The RANSAC algorithm is an iterative method to estimate parameters of a mathematical model from a set of sensor data [42, 43]. Within the scope of reconstructing planes out of disturbed depth data, the mathematical model of a plane is described by three 3D points.



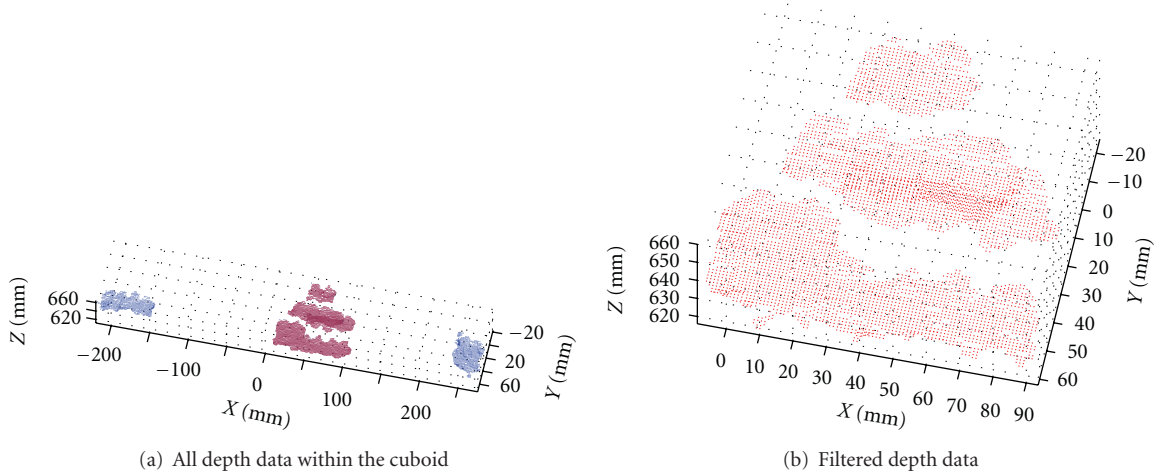


FIGURE 9: Prefiltering measured data.

The algorithm finds the plane as follows:

- (1) Define a plane by randomly taking three points from the depth data.
- (2) If this plane is parallel to the sensor's  $x$ - $y$ -plane then continue, otherwise choose different points.

- (3) Calculate the distance between each 3D point within the depth data and the randomly generated plane.

- (4) Determine the score  $S$  of the plane as follows:

$$S = \sum_{i=0}^{i=n} p(\text{distance}) \quad \text{with } p = \begin{cases} 1 - \frac{\text{distance}}{\text{maximum distance}}, & \text{if distance} \leq \text{maximum distance,} \\ 0, & \text{if distance} > \text{maximum distance.} \end{cases} \quad (1)$$

- (5) Repeat steps 1 to 4 for a predefined number of iterations.
- (6) The plane with the highest score is the best estimation of a plane within the depth data. If the highest plane score is beneath 80% of the theoretical minimum plane score, no plane is found.

The value assigned to the maximum distance results from the discrete positions of the Hubelino parts. As those positions are defined by the round shapes on top of each part, it is obvious that the minimum distance between two planes within the depth image is 16 mm (compare Figure 8). In order to allocate each depth value to a plane, the maximum distance between a measured point and the corresponding plane is 8 mm.

The decision, if a found plane is valid, depends on the minimum theoretical plane score. This parameter is equal to the theoretical number of measurements on the surface of the smallest possible plane (see Figure 10).

The theoretical number of depth data within a certain area yields from the sensor characteristics. As the Kinect depth sensor has a resolution of  $640 \times 480$  pixels and the lens

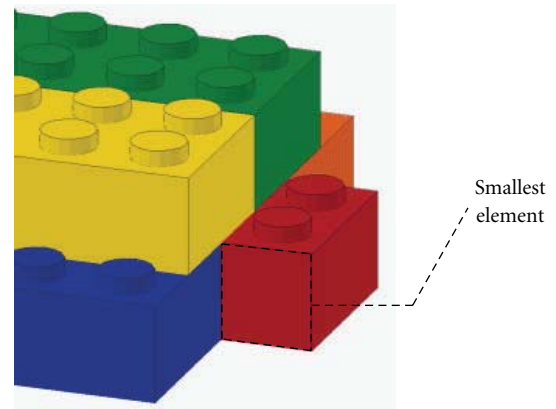


FIGURE 10: Definition of the smallest element.

has an opening angle of 57 degree horizontally, 43 degree vertically, respectively, the spatial resolution decreases with increasing distance. This interrelationship is given in Figure 11.

As Figure 11 shows, the theoretical minimum plane score depends on the distance of the plane and thus needs to be determined individually for each plane. Considering the fact

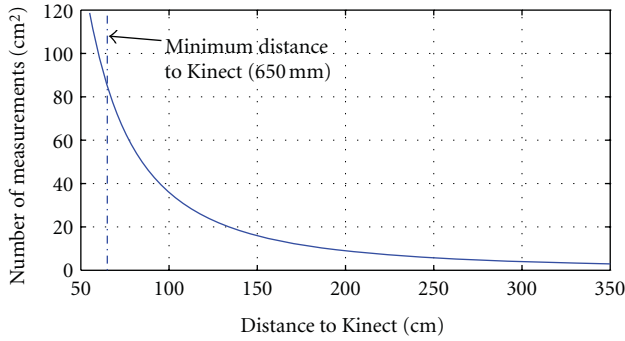


FIGURE 11: Interrelationship of spatial resolution and distance.

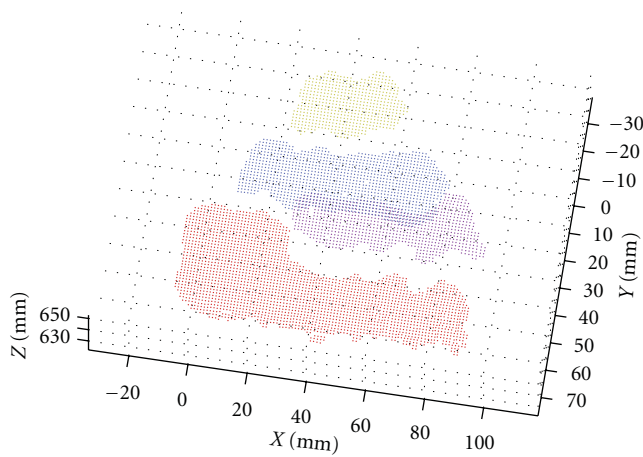


FIGURE 12: Found planes.

that the Kinect sensor is disturbed by noise, the found plane is valid, if its score is at least 80% of the theoretical score.

As the assembly group generally consists of more than one plane, the RANSAC algorithm has to be executed several times. Therefore the depth values belonging to the already found planes are deleted, and within the remaining depth data another plane is searched. This procedure is repeated until no further plane can be found. The result is shown in Figure 12.

**3.2.5. Fitting of Virtual Hubelino Bricks.** After finding the planes within the depth data, the single Hubelino parts can be fitted. Since the Hubelino parts can only be placed at discrete positions they can occlude each other. Therefore, the smallest visible element is half of one side of a  $32\text{ mm} \times 32\text{ mm}$  Hubelino brick (see Figure 10). Hence, only rectangles with a height of 32 mm and a width of 16 mm are positioned.

The fitting process itself is based on the assumption that size and shape of the found planes can only consist of multiples of the smallest elements (compare Figure 13). Hence for each plane the surrounding rectangle is calculated (light blue line). Regarding the sensor disturbance, the dimensions of this rectangle are not necessarily a multiple of the smallest element. Consequently, the size of the rectangle

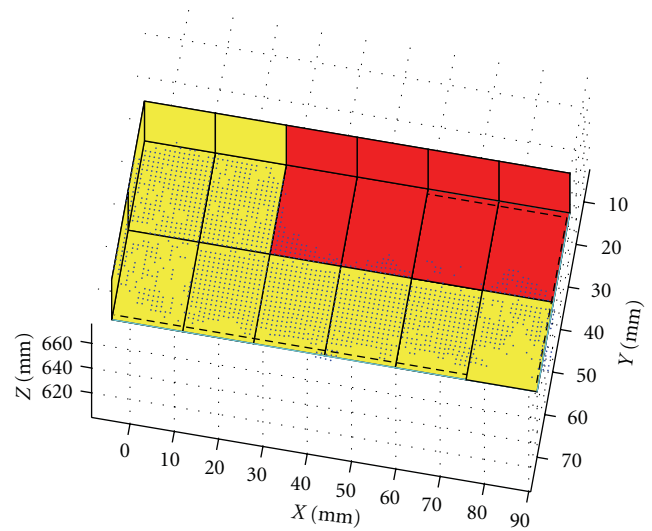


FIGURE 13: Fitting elements into plane number 4.

is approximated towards the nearest multiple of the smallest element's size (black dashed line). Afterwards, the resulting rectangle is filled with smaller rectangles that have the shape of the smallest elements (red and yellow cuboids). This procedure is repeated for each found plane.

Since not all of those fitted elements really exist, the next step is to decide whether an element belongs to the assembly group or not. This decision is based on the quotient of the actual number of measured 3D depth points on the element and the theoretical maximum number of data points on the element. This maximum is given by Figure 11. Since the edge line of the found planes is irregular, the quotient is generally not equal to one. Thus a threshold of 90% is defined. Concerning this threshold, the decision whether a fitted element belongs to the assembly group can be evaluated by the following steps:

- (1) Determine the number of measured depth data within the element.
- (2) Based on the distance of the element calculate how many data points should be on the element.
- (3) If the result of step 1 is at least 90% of the result of step 2, then the element belongs to the assembly group.

This procedure is repeated for each element on each found plane and results in a model for the examined side. Within Figure 13, the red marked cuboids do not belong to the assembly group.

**3.2.6. Error Correction for a Single Side.** After the Hubelino parts are positioned according to the established planes, the resulting model is checked for errors. This step proved to be necessary, after several initial evaluation tests failed due to frayed shapes of the identified planes (compare Figure 13).

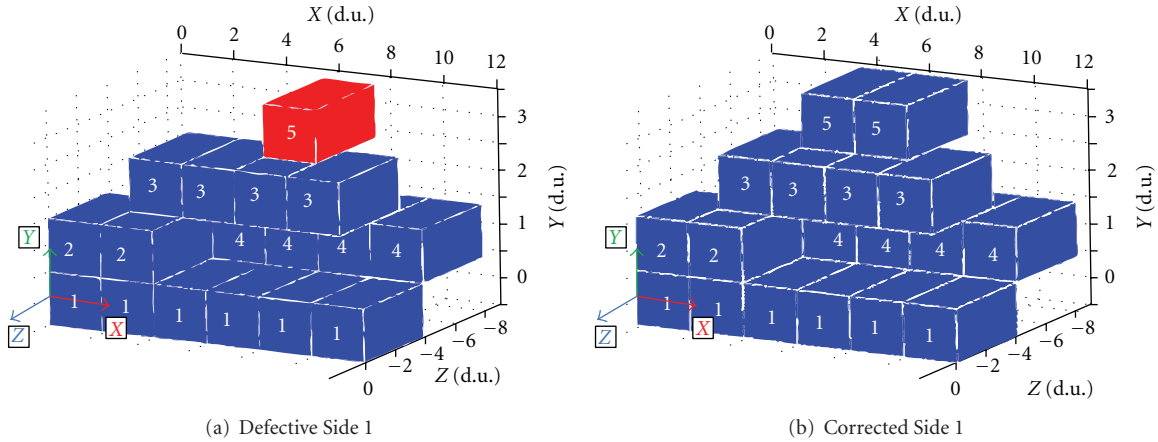


FIGURE 14: Correction of side 1.

Within the correction process, the found model has to meet the following requirements:

- (1) A single smallest element can only appear if the rest of the Hubelino part is masked (compare Figure 10).
- (2) Each line of the model must consist of an even number of smallest elements.
- (3) All found elements have to be placed at a discrete position.

In order to check these conditions, first the coordinate system is changed. Since the presented calculations contain the measured data, they are based on the coordinate system defined by OpenNI. This measured data is represented by the fitted elements which are placed in discrete positions. Hence, the unit of the new coordinate system is given in dimension units (d.u.). Thereby, the size of two dimension units equals the width of the smallest element. The new coordinate system is placed on the lower left element of the scanned side (see Figure 14). Within this coordinate system the values of the  $x$ - and  $y$ -axis are positive and those for the  $z$ -axis are negative. In Matlab the Hubelino model is contained in an array according to discrete positions of smallest element's multiples (see Figure 10).

The error correction process is illustrated in Figure 14. First, the found elements are grouped (numbers on the elements in Figure 14) based on their position within the model. Therefore, all elements within the same coordinates in  $y$  and  $z$  are grouped together. For each group with a noneven number of members (red marked element) the fitting of the element inside the plane is rechecked. Within this check, the unused elements of a plane are analyzed. Of special interest are the elements at the left and right of the defective group, respectively. Those elements meet only one condition listed in Table 3 and result in the corresponding error correction.

Each increase as well as decrease refers to the width of an element. For each correction process only the dimension of the surrounding rectangle changes. The position of its center remains the same. The advantage of this error correction process is that it can be calculated straightforward without iteratively searching the best solution.

TABLE 3: Possible errors and corresponding corrections.

Error	Correction
Both elements were not chosen	Increase the width of the surrounding rectangle
Both elements were chosen	Decrease the width of the surrounding rectangle
One element was chosen	Decrease the width of the surrounding rectangle

Figure 15 demonstrates this error correction process for the defectively fitted group presented in Figure 14(a). For this figure it is obvious that the error correction finds a better mapping between the surrounding rectangle (light blue line) and the measured data.

Since the calculated model is based on disturbed sensor data, it is generally not the exact model of the actual assembly group. Hence, the last step is to reconstruct the complete 3D model of the assembly group and to check if all four sides match.

**3.2.7. Reconstruction of the Complete Model.** In order to reconstruct the complete model of the assembly group out of the models for each side, first the transformations between adjacent sides have to be calculated. Considering the anti-clockwise chronology of the scan positions (see Figure 4), the values of the right-hand rotation matrix to rotate side B into the coordinate system of side A are given as follows:

- (i) Rotation around the  $x$ -axis:  $0^\circ$ .
- (ii) Rotation around the  $y$ -axis:  $90^\circ$ .
- (iii) Rotation around the  $z$ -axis:  $0^\circ$ .

The translation vector between both sides is calculated as follows:

- (a) Starting from the assumption that model A is correct, calculate those parts within model A that are expected to be part of side B.
- (b) Within side B, calculate those elements, which side A would expect to be part of side B.

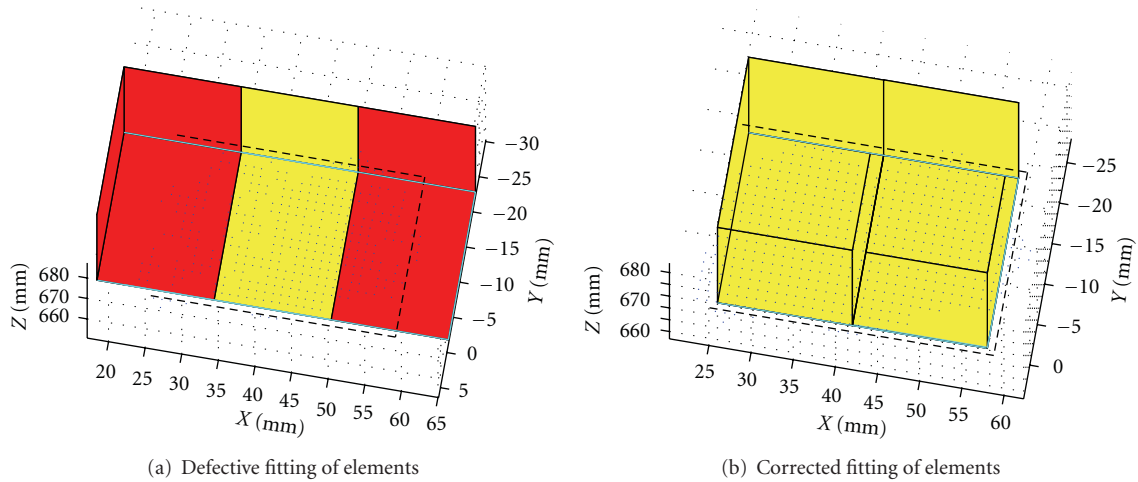


FIGURE 15: Failure correction for plane 1 within side 1.

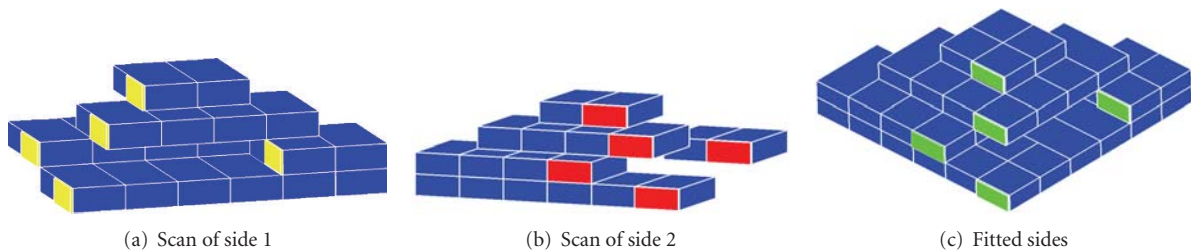


FIGURE 16: Fitting of side 1 and 2.

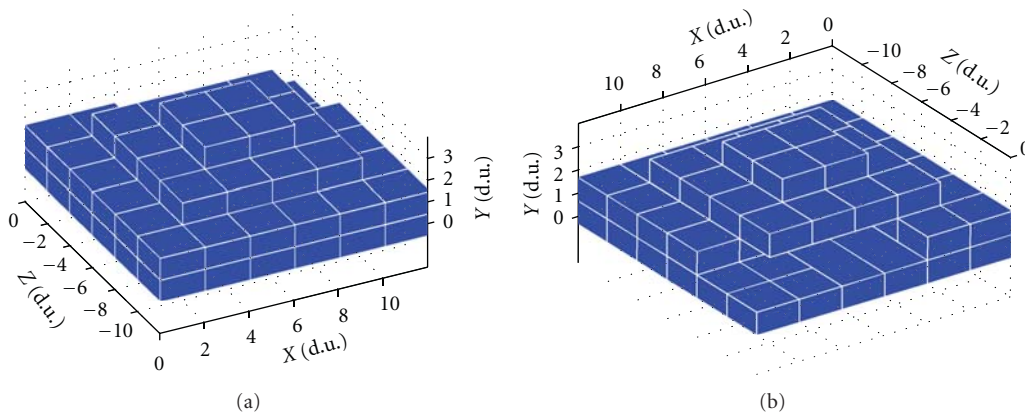


FIGURE 17: Reconstructed assembly group.

- (c) Rotate side B into the coordinate system of side A.
- (d) Find the translation vector mapping most elements of step 1 to those of step 2.

If not all elements of step (b) are mapped to those of step (a), at least one of the involved sides contains errors. Figure 16 shows this procedure for side 1 and 2 (compare Figure 8). The calculated elements of steps (a) and (b) are marked in yellow and red, respectively. Afterwards those elements are mapped onto each other by the found transformation (green elements).

Those transformations are calculated between all adjacent sides. If none of the sides contains errors, the 3D model of the complete assembly group can be calculated. The 3D model of the analyzed assembly group is shown in Figure 17.

**3.2.8. Color Detection.** In order to recreate an exact virtual model of the assembly group, information about the color of the detected parts needs to be acquired. In case of a scenario, that involves a color camera as the sole source of information, usually very complex algorithms have to be applied in order to efficiently detect regions of certain colors. In the given

TABLE 4: RGB values for the main colors used in the scenario.

	<i>R</i>	<i>G</i>	<i>B</i>
<i>Red</i>	196	40	27
<i>Green</i>	40	127	70
<i>Blue</i>	13	105	171
<i>Yellow</i>	245	205	47

case, however, this task is extremely simplified, since it is already known which pixels belong to a certain part. This information can be used to identify the color for a detected part. As an example, it can be assumed that an identified pixel region  $x_m \cdots x_n$  contains  $j$  pixels for one side of a detected part. For the assumption that all of those pixels belong to the same part and should thus roughly contain the same color information, the basic approach of calculating the mean RGB value should suffice. Thus the resulting RGB value for the detected part can easily be calculated as:

$$\begin{aligned} cd.R &= \frac{\sum_{k=m}^n x_k.R}{j} & cd.G &= \frac{\sum_{k=m}^n x_k.G}{j} \\ cd.B &= \frac{\sum_{k=m}^n x_k.B}{j}. \end{aligned} \quad (2)$$

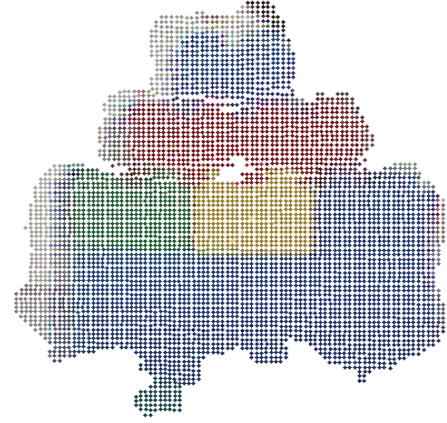
Since the RGB values for most Hubelino parts are publicly available, see for example, [44], these can be used to identify the corresponding part color. The RGB values for the parts used in this example are presented in Table 4.

Given the fact that the different colors are saved in an array  $z$ , the resulting part color can be calculated:

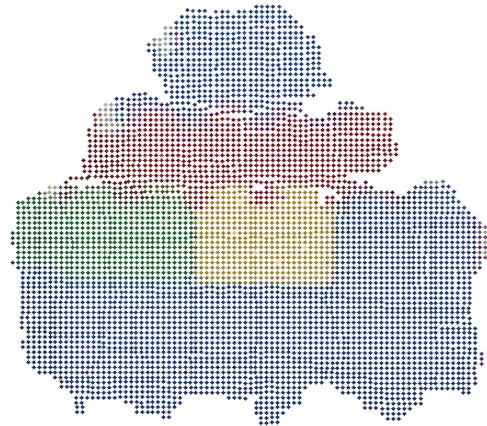
$$\begin{aligned} \|(z_i.R - cd.R)\| + \|(z_i.G - cd.G)\| \\ + \|(z_i.B - cd.B)\| = \min. \end{aligned} \quad (3)$$

However, since only four different part colors are possible, the significant distance between the different RGB values partly contributes to the efficiency of this approach. Since the RGB values are determined by the manufactured parts and thus do not relate to the specific conditions (i.e., lighting, reflection), even better results can be achieved if these factors are taken into consideration. Hence, sample values for the different colors have to be recorded manually. Since lighting is a significant factor, the best results can be achieved by recording different color values for each of the four viewpoints.

In order to achieve even better results, the calculation of the RGB values for a detected part can be improved. In the previous approach, all of the points that were considered as belonging to a specific part were used for the determination of that part's color. Inasmuch as all of those color values are roughly the same, this approach is sufficient. However, the approach can be improved if the mean RGB value for a part is calculated using only the ten most central points for a specific brick. On the one hand this approach eliminates the problem that the part's color tends to change at the part's border. On the other hand this might lead to false detections in case that local reflection takes place near a part's center.



(a) Mapping without calibration



(b) Final mapping

FIGURE 18: Mapped sensor data.

Given the situation that a limited solution space of colors is present, the problem of reflection can be reduced by comparing each of a part's pixels with the possible colors. A color is assigned to a pixel if

$$\begin{aligned} \|(z_i.R - cd.R)\| + \|(z_i.G - cd.G)\| + \|(z_i.B - cd.B)\| = \min \cap \\ \|(z_i.R - cd.R)\| + \|(z_i.G - cd.G)\| + \|(z_i.B - cd.B)\| < \delta. \end{aligned} \quad (4)$$

Thus it is possible to dynamically apply different  $\delta$  for different scenarios and lighting conditions. Subsequently, the color for a part is computed by averaging over the pixels that were assigned a specific color. This computation reduces the problem of reflection, since colors that are out of range do not contribute to the resulting part color.

As shown in Figure 18(a) another problem, that needed to be resolved, is based on the fact that there is a partial displacement between the color information and the depth information. Since the color detection algorithms rely on the assumption that for each pixel in the depth information the image data corresponds to that exact same spot in the real world, this displacement leads to miscalculations of a part's color. As the Kinect is attached to the flange of a robot,

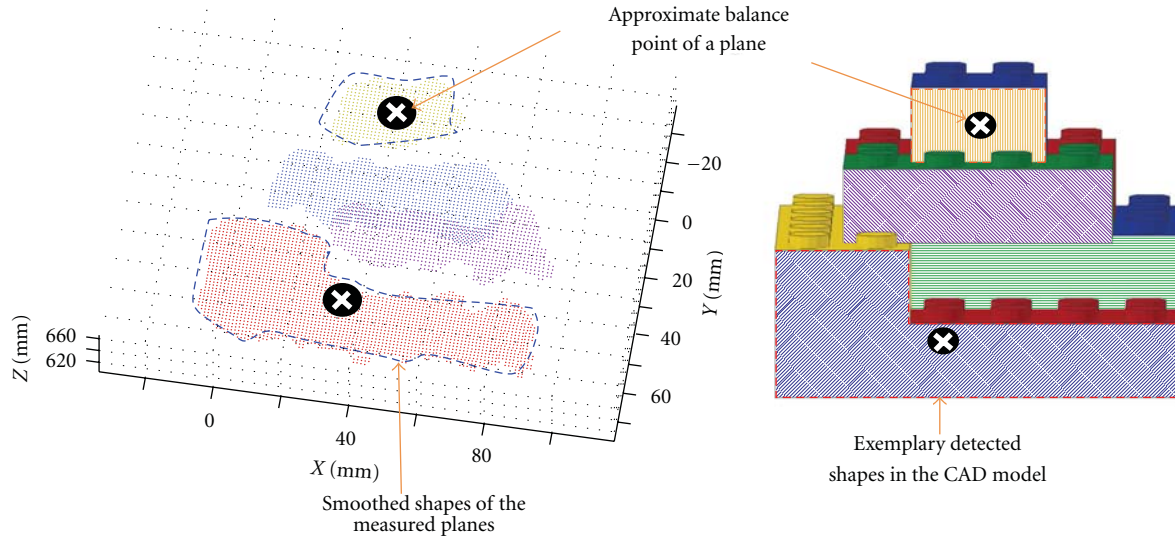


FIGURE 19: Model-based approach.

however, the information about the current position of the robot's tool center point (TCP) can be utilized for the exact determination of a viewpoint's displacement within the assembly area. The resulting image containing the mapping of both depth and color data is shown in Figure 18(b).

**3.3. Generalized, Model-Based Approach.** As a modification to the presented approach, the elimination of assumptions for the matching process offers to broaden the range of detected parts without additional engineering efforts. Due to the fact that the model needs to be verified, it is apparent that there is already a CAD model present for the desired assembly group. The superior goal is to use this model in order to match CAD and Kinect data without the need for additional assumptions regarding the presence of an object. This task is often achieved by photogrammetry methods [45, 46], but normally relies on different conditions regarding time and accuracy [47–49].

As a first step to realize this approach, the CAD model has to be analyzed in regard to different layers that can be perceived from different viewpoints. The result of this step is a “virtual model” of the assembly group that should be built. The model that is recorded with the Kinect forms the “perceived model”. For this generalized approach it can be stated, that the point of view from which an assembly group is looked at in the virtual and the perceived perspective have to match to achieve the best results. But since it is always possible to correct the perspective for both the virtual and perceived model, there is no need for an exact calibration between these two perspectives. The only requirement is that the basic “sides,” from which an assembly group is looked at, match. As previously stated, the four viewpoints correspond to the four sides of the rectangle that defines the maximum assembly space (see Figure 4). Thus for each of those sides the visible layers need to be extracted from the virtual model. The fact that the sides are either parallel to the  $x$ - $y$ -layer or the  $y$ - $z$ -layer of the modeling environment simplifies

this task, inasmuch as that for the different viewpoints the polygons with the minimum or maximum  $x$  or  $z$  value, respectively, form the foremost layer for each height, which is given in the  $y$ -coordinate. As an example for viewpoint 1 the maximum  $x$  values for a given height  $y$  define the visible planes. The polygons themselves are extracted from the underlying CAD model. All of the different polygons that are adjacent to each other comprise a region. In order to utilize these regions for further processing, their shape and balance point are determined.

Once the different layers have been identified in the virtual model, the same steps as in the approach mentioned above—from recording the sensor data to the RANSAC algorithm—are performed. Hence the results of these steps are the identified regions for the perceived data. For these regions the different shapes and balance points are extracted from those results. The shape herein is defined by the outmost data points for a specific region. Figure 19 shows this process. The process is illustrated exemplarily for two out of four planes within the assembly. In the left of Figure 19 the planes, found by the RANSAC algorithm within the measured data, are presented. For both planes their smoothed shapes and balance points are shown. The right side illustrates both planes within the underlying virtual model as well as their ideal shapes and ideal balance points.

As a last step, this information about shapes and balance points is used to match the regions of the virtual and the perceived model. This is realized by first comparing the number of balance points for the virtual and perceived model. If there is a difference between those numbers then it becomes apparent that there is an error. After that the Gauss algorithm (cp. [50]) is applied to match the balance points of the different models. The algorithm completes when either the distance between the balance points is small enough or the Gauss algorithm exceeds a certain number of iterations. Next the balance points of the two models can be mapped by taking the smallest difference between two points as the main

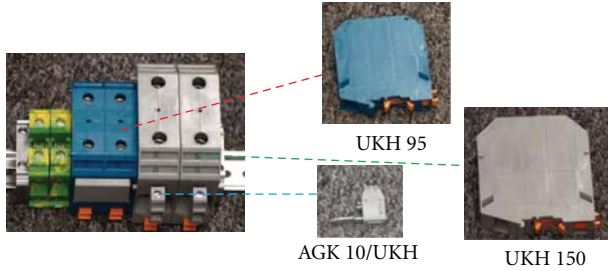


FIGURE 20: Final assembly.

criteria. If the numbers of balance points were different, only points that have a minimum distance are considered. In the following the regions that belong to a balance point in the perceived model are projected onto the region of the corresponding balance point in the virtual model. By calculating the overlapping areas it is possible to determine if the two models match or if it is more likely that there is a difference between them.

#### 4. Industrial Application

In order to apply the achieved results in an industrial application, the concepts already tested with the demonstrator cell were simulated, in collaboration with automation technology manufacturer Phoenix Contact, on a scenario for the assembly of switch cabinets. This approach addresses the customer-driven switch cabinet assembly system, which Phoenix Contact operates in addition to the final assembly line in its mass-production-style manufacturing system. The assembly of switch cabinets consists of components (control units, terminals, etc.) in configurations predefined by the customer that are mounted onto top-hat sections and passed on as completed modules to switch-cabinet production (module assembly). Additionally some of these components need to be equipped with specific clip combinations. Figure 20 shows an example of this system.

A model of each customer-specific and mass-production-style switch cabinet is available in a CAD format. Currently this data is used to derive assembly instructions which are printed out and handed to a human worker. Finally the manually assembled switch cabinet is analyzed by an image recognition system and the results are compared to the mounting requirements.

Since the target assembly is available electronically, it is feasible to cognitively automate the process, hence achieving economic advantages. The main challenges involved are as follows:

- (i) Construction of a continuous flow of information from the CAD system to the manufacturing system.
- (ii) Robust system components and joining processes.
- (iii) Logistical concepts and components for efficient stock placement during the assembly process.
- (iv) Automated verification of the final assembly and man-machine interaction in case of errors.



FIGURE 21: Mapped depth and color data for the switch-cabinet scenario (one viewpoint).

The switch-cabinet scenario bears strong similarities to the demonstrator-cell scenario, that is, in regard to the customer-defined CAD models and requirements regarding an automated verification of the final assembly.

Hence, in order to verify the capabilities of the Assembly Group Analysis, the perception and verification process using the Kinect was simulated for a given scenario. Mapped depth and color data for the final assembly indicating the different parts that it consists of is given in Figure 21.

For the verification process, three different cases are taken into consideration. In the first case, the assembly was built correctly. In the second case, one of the larger feed-through modular terminal blocks UKH 95 has not been placed onto the mounting rail. In the third case only one of the smaller pick-off terminal blocks AGK 10/UKH has not been connected to the UKH 150. Figure 22 presents the resulting Kinect data for the different cases. For the first erroneous case the missing UKH 95 results in a smaller region of the matching depth data. In comparison with the Hubelino scenario, this might be the case, if two adjacent bricks that have the same color are present in the desired CAD model, but only one of them has been placed in the real scenario. For the second error case the AGK 10/UKH defines its own region, since it is not directly connected to any other component. A comparable Hubelino scenario would be that a single brick, that is, one that does not comprise a layer with any other brick, has not been placed into the assembly group.

Through transferring the described approaches in cognition as well as in 3D Assembly Group Analysis onto this scenario, an autonomous assembly process can be established. Particularly, within the focus of customer-defined switch-cabinet assemblies, a cognitive production cell is useful as it easily adapts the production process on changing assemblies. Additionally, an automated analysis process relieves the human worker and is less error-prone.

#### 5. Conclusion

In order to resolve the problem that the commissioning and programming of complex robotized assembly takes considerable planning efforts which do not directly contribute to

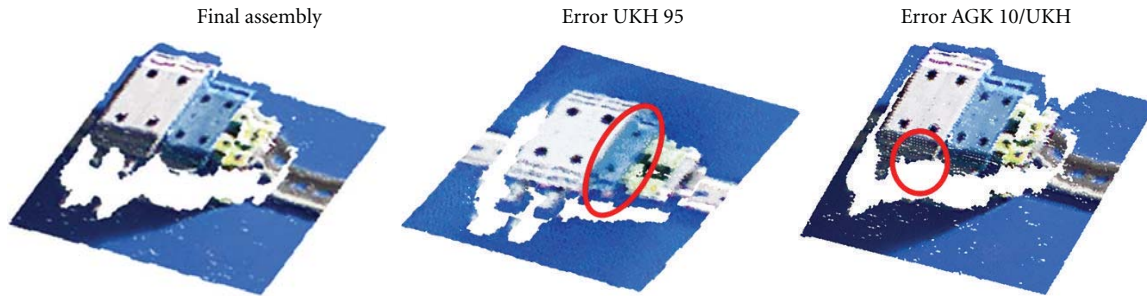


FIGURE 22: Resulting depth data for switch-cabinet scenario.

the added value, the overall approach was to shift planning tasks to the level of execution. Hence, within the Cluster of Excellence “Integrative Production Technology for High-Wage Countries” at RWTH Aachen University, a Cognitive Control Unit had been created, which is able to plan and execute action flows autonomously for a given goal state. The combination with an ergonomic user-centered human machine interface allows for an interaction with a human expert in order to partially transfer planning and implementation tasks to the technical cognitive system as well as to present him/her the opportunity to intervene in case of assembly errors. To test and further enhance the CCU in a near-reality production environment in a variety of different assembly operations, a robotic assembly cell had been set up. The scenario comprises major aspects of an industrial application, while at the same time easily illustrating the potential of a cognitive control system. As a demonstration scenario, the building of an assembly group comprised of Hubelino bricks, which is given to the CCU in the form of a CAD file, was chosen.

In order to verify an assembled group, the CCU requires a recognition process. Additionally, the human operator must be supported in case of assembly errors. This verification process including the feedback to a human expert was realized using the Microsoft Kinect as a perception device. The Kinect was chosen since it merges the capability of a depth sensor and a color camera in a single device, hence providing an innovative approach for research and development for the field of environment recognition.

In the scenario, the Kinect was attached to the flange of an industrial robot, allowing for a positioning at multiple viewpoints in order to create a virtual image of the current assembly group. For each viewpoint the depth data as well as color information is recorded and analyzed to create a discrete model for each side. Afterwards, the discrete models are combined into a complete model of the assembly group. Finally, the superior goal of verifying a real assembly group of Hubelino bricks was achieved by comparing the resulting perceived model with the virtual model from the customer-defined CAD file.

In order to demonstrate the implications in an industrial application, the concepts that were tested with the demonstrator cell were transferred to a scenario for the assembly of switch-cabinets. This scenario bears strong similarities to the demonstrator cell, that is, in regard to the customer-defined

CAD models and requirements regarding an automated verification of the final assembly.

## Acknowledgment

The authors would like to thank the German Research Foundation (DFG) for its kind support of the research on cognitive automation within the Cluster of Excellence Integrative Production Technology for High-Wage-Countries. No other sponsorship or economic interests were involved which ensured free evaluation of the experiments.

## References

- [1] E. von Weizsäcker, *Globalisierung der Weltwirtschaft - Herausforderungen und Antworten*, vol. 5 of *Politik und Zeitgeschichte*, 2003, [http://www.bpb.de/publikationen/U27MV5,0,Globalisierung\\_der\\_Weltwirtschaft\\_Herausforderungen\\_und\\_Antworten.html](http://www.bpb.de/publikationen/U27MV5,0,Globalisierung_der_Weltwirtschaft_Herausforderungen_und_Antworten.html).
- [2] DIHK, “Produktionsverlagerung als Element der Globalisierungsstrategie von Unternehmen: Ergebnisse einer Unternehmensbefragung,” 2011, <http://www.dihk.de/ressourcen/downloads/produktionsverlagerung.pdf>.
- [3] DIHK, “Auslandsinvestitionen in der Industrie,” 2011, [http://www.muenchen.ihk.de/mike/ihk\\_geschaeftsfelder/starthilfe/Anhaenge/DIHK-Auslandsinvestitionen-2010.pdf](http://www.muenchen.ihk.de/mike/ihk_geschaeftsfelder/starthilfe/Anhaenge/DIHK-Auslandsinvestitionen-2010.pdf).
- [4] F. Klocke, “Production technology in high-wage countries: from ideas of today to products of tomorrow,” in *Industrial Engineering and Ergonomics: Visions, Concepts, Methods and Tools*, C. Schlick, Ed., Festschrift in Honor of Professor Holger Luczak, Springer, New York, NY, USA, 2009.
- [5] S. Kinkel, *Arbeiten in der Zukunft: Strukturen und Trends der Industriearbeit*, Ed. Sigma, Berlin, Germany, 2008.
- [6] L. Bainbridge, “Ironies of automation,” in *New Technology and Human Error*, J. Rasmussen, K. Duncan, and J. Leplat, Eds., pp. 775–779, John Wiley & Sons, Chichester, UK, 1987.
- [7] R. Onken and A. Schulte, *System-Ergonomic Design of Cognitive Automation: Dual-Mode Cognitive Design of Vehicle Guidance and Control Work Systems*, Springer, Berlin, Germany, 2010.
- [8] C. Brecher, Ed., *Integrative Produktionstechnik für Hochlohnländer*, Springer, Berlin, Germany, 2011.
- [9] M. F. Zäh, M. Beetz, K. Shea et al., “The cognitive factory,” in *Changeable and Reconfigurable Manufacturing Systems*, H. A. El-Maraghy, Ed., Springer, Berlin, Germany, 2009.
- [10] S. Karim, L. Sonenberg, and A. H. Tan, “A hybrid architecture combining reactive plan execution and reactive learning,” in



- Proceedings of the 9th Biennial Pacific Rim International Conference on Artificial Intelligence (PRICAI '06)*, China, 2006.
- [11] E. Gat, "On three-layer architectures," in *Artificial Intelligence and Mobile Robots*, D. Kortenkamp, R. Bonasso, and R. Murphy, Eds., pp. 195–211, AAAI Press, Menlo Park, Calif, USA, 1998.
  - [12] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Pearson Education, Upper Saddle River, NJ, USA, 2003.
  - [13] K. Paetzold, "On the importance of a functional description for the development of cognitive technical systems," in *International Design Conference*, Dubrovnik, Croatia, 2006.
  - [14] P. Adelt, J. Donath, J. Gausemeier et al., "Selbstoptimierende Systeme des Maschinenbaus," in *HNI-Verlagsschriftenreihe*, J. Gausemeier, F. Rammig, and W. Schäfer, Eds., Westfalia Druck GmbH, Paderborn, Germany, 2009.
  - [15] M. Zäh and M. Wiesbeck, "A model for adaptively generating assembly instructions using state-based graphs," in *Manufacturing Systems and Technologies for the New Frontier*, The 41st CIRP Conference on Manufacturing Systems, Tokyo, Japan, 2008.
  - [16] M. Zäh, M. Wiesbeck, F. Engstler et al., "Kognitive Assistenzsysteme in der manuellen Montage," in *wt Werkstatttechnik*, vol. 97, no. 9, pp. 644–650, 2007.
  - [17] H. Ding, S. Kain, F. Schiller, and O. A. Stursberg, "Control architecture for safe cognitive systems," in *10. Fachtagung Entwurf komplexer Automatisierungssysteme*, Magdeburg, Germany, 2008.
  - [18] S. Kammel, J. Ziegler, B. Pitzer et al., "Team AnnieWAY's autonomous system for the 2007 DARPA Urban Challenge," *Journal of Field Robotics*, vol. 25, no. 9, pp. 615–639, 2008.
  - [19] H. J. Putzer, *Ein uniformer Architekturansatz für Kognitive Systeme und seine Umsetzung in ein operatives Framework*, Dr. Köster, Berlin, Germany, 2004.
  - [20] D. Ewert, E. Hauck, A. Gramatke, and S. Jeschke, "Cognitive assembly planning using state graphs," in *Proceedings of the 3rd International Conference on Applied Human Factors and Ergonomics*, Miami, Fla, USA, 2010.
  - [21] T. Kempf, W. Herfs, and C. Brecher, "Cognitive Control Technology for a Self-optimizing Robot Based Assembly Cell," in *ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conferences (DETC '08)*, New York, NY, USA, August 2008.
  - [22] M. Mayer, B. Odenthal, M. Faber et al., "Cognitive engineering for direct human-robot cooperation in self-optimizing assembly cells," in *First International Conference on Human Centered Design (HCD '09)*, M. Kurosu, Ed., San Diego, Calif, USA, July 2009.
  - [23] C. Schlick, B. Odenthal, M. Mayer et al., "Design and evaluation of an augmented vision system for self-optimizing assembly cells," in *Industrial Engineering and Ergonomics: Visions, Concepts, Methods and Tools*, C. Schlick, Ed., Festschrift in Honor of Professor Holger Luczak, Springer, New York, NY, USA, 2009.
  - [24] T. B. Sheridan, *Humans and Automation: System Design and Research Issues*, Human Factors and Ergonomics Soc, Santa Monica, Calif, USA, 2001.
  - [25] M. Mayer, B. Odenthal, and M. Grandt, "Task-oriented process planning for cognitive production systems using MTM," in *Proceedings of the 2nd International Conference on Applied Human Factors and Ergonomic*, Louisville, Ky, USA, 2008.
  - [26] V. Gazzola, G. Rizzolatti, B. Wicker, and C. Keysers, "The anthropomorphic brain: the mirror neuron system responds to human and robotic actions," *NeuroImage*, vol. 35, no. 4, pp. 1674–1684, 2007.
  - [27] E. Drumwright, "Toward a vocabulary of primitive task programs for humanoid robots," in *Proceedings of the International Conference on Development and Learning (ICDL '06)*, Bloomington, Ind, USA, 2006.
  - [28] B. Odenthal, M. Mayer, N. Jochems, and C. Schlick, "Cognitive engineering for human-robot interaction—the effect of subassemblies on assembly strategies," in *Advances in Human Factors, Ergonomics, and Safety in Manufacturing and Service Industries*, pp. 1–10, CRC Press, Boca Raton, Fla, USA, 2011.
  - [29] M. Mayer, C. Schlick, D. Ewert et al., "Automation of robotic assembly processes on the basis of an architecture of human cognition," *Production Engineering Research and Development*, vol. 5, no. 4, pp. 423–431, 2011.
  - [30] T. Kempf, *Ein kognitives Steuerungsframework für robotergestützte Handhabungsaufgaben*, Apprimus, Aachen, Germany, 1st edition, 2010.
  - [31] B. Odenthal, M. Mayer, W. Kabuß, B. Kausch, and C. Schlick, "Investigation of error detection in assembled workpieces using an augmented vision system," in *Proceedings of the 17th World Congress on Ergonomics (IEA '09)*, Beijing, China, August 2009.
  - [32] B. Odenthal, M. Mayer, W. Kabuß, and C. Schlick, "Einfluss der Bildschirmposition auf die Fehlererkennung in einem Montagebauteil," in *Neue Arbeits- und Lebenswelten gestalten: Vom 24. - 26. März*, M. Schütte, Ed., pp. 203–206, GfA-Press, Dortmund, Germany, 2010.
  - [33] B. Odenthal, M. Mayer, W. Kabuß, B. Kausch, and C. Schlick, "An empirical study of disassembling using an augmented vision system," in *3rd International Conference on Digital Human Modeling (ICDHM '11)*, Orlando, Fla, USA, 2011.
  - [34] C. Beder, B. Bartczak, and R. Koch, "A comparison of PMD-cameras and stereo-vision for the task of surface reconstruction using patchlets," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '07)*, June 2007.
  - [35] C. Sa, "Time of Flight Camera Technology," *Technology*, 2009.
  - [36] D. Schauer, *Integration of a 3D time of flight camera system into a robot system: integration, validation and comparison*, VDM Verlag Dr. Müller, Saarbrücken, Germany, 2011.
  - [37] Microsoft, Ed, "Programming Guide: Getting Started with the Kinect for Windows SDK Beta from Microsoft Research," 2011.
  - [38] D. Laing, "Kinect - Could this technology have a future in industrial automation?" <http://blog.vdcresearch.com/industrial-automation/2010/11/kinect-could-this-technology-have-a-future-in-industrial-automation.html>.
  - [39] D. L. Andrews, *Structured light and its applications: an introduction to phase-structured beams and nanoscale optical forces*, Academic Press, Amsterdam, The Netherlands, 2008.
  - [40] Z. Song, *Use of structured light for 3D reconstruction*, Hong Kong, 2008.
  - [41] K. Clague and M. Agullo, *LEGO software power tools*, Syngress, Rockland, Mass, USA, 2002.
  - [42] Z. Yaniv, "Random Sample Consensus (RANSAC) Algorithm, A Generic Implementation," *Information Systems Journal*, 2010, <http://isiswiki.georgetown.edu/zivy/writtenMaterial/RANSAC.pdf>.
  - [43] R. Szeliski, "Computer vision: Algorithms and applications," *Computer Vision*, 2011, <http://www.worldcat.org/oclc/700473658>.

- [44] C. Stephens, “Lego Color List - official names, numbers, CYMK, RGB, Pantone values,” 2011, <http://isodomos.com/Color-Tree/Lego-List.html>.
- [45] C. Zhang, N. Xi, and Q. Shi, “Object-orientated registration method for surface inspection of automotive windshields,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '08)*, pp. 3553–3558, September 2008.
- [46] X. Liang, J. Liang, J. Liu, and C. Guo, “A rapid inspection method for large water turbine blade,” in *IEEE International Conference on Mechatronics and Automation (ICMA '10)*, pp. 712–716, August 2010.
- [47] F. Bosché, “Automated recognition of 3D CAD model objects in laser scans and calculation of as-built dimensions for dimensional compliance control in construction,” *Advanced Engineering Informatics*, vol. 24, no. 1, pp. 107–118, 2010.
- [48] L. Yue and X. Liu, “Application of 3D optical measurement system on quality inspection of turbine blade,” in *16th IEEE International Conference on Industrial Engineering and Engineering Management (IE and EM '09)*, pp. 1089–1092, October 2009.
- [49] A. Tellaèche, R. Arana, A. Ibarguren, and J. Martínez-Otzeta, “Automatic quality inspection of percussion cap mass production by means of 3D machine vision and machine learning techniques,” in *Lecture Notes in Computer Science, Hybrid Artificial Intelligence Systems*, M. Graña Romay, E. Corchado, and M. Garcia Sebastian, Eds., pp. 270–277, Springer, Berlin, Germany, 2010.
- [50] B. Muralikrishnan and J. Raja, *Computational Surface and Roundness Metrology*, Springer, 2009.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

