



A framework for flexible and reconfigurable vision inspection systems

Francesco Lupi¹ · Michele Biancalana² · Andrea Rossi² · Michele Lanzetta²

Received: 9 May 2023 / Accepted: 15 August 2023
© The Author(s) 2023

Abstract

Reconfiguration activities remain a significant challenge for automated Vision Inspection Systems (VIS), which are characterized by hardware rigidity and time-consuming software programming tasks. This work contributes to overcoming the current gap in VIS reconfigurability by proposing a novel framework based on the design of Flexible Vision Inspection Systems (FVIS), enabling a Reconfiguration Support System (RSS). FVIS is achieved using reprogrammable hardware components that allow for easy setup based on software commands. The RSS facilitates offline software programming by extracting parameters from real images, Computer-Aided Design (CAD) data, and rendered images using Automatic Feature Recognition (AFR). The RSS offers a user-friendly interface that guides non-expert users through the reconfiguration process for new part types, eliminating the need for low-level coding. The proposed framework has been practically validated during a 4-year collaboration with a global leading automotive half shaft manufacturer. A fully automated FVIS and the related RSS have been designed following the proposed framework and are currently implemented in 7 plants of GKN global automotive supplier, checking 60 defect types on thousands of parts per day, covering more than 200 individual part types and 12 part families.

Keywords Flexible manufacturing · Reconfigurable manufacturing · Visual inspection · Variant process planning · Support systems · Feature extraction

1 Introduction

Next-generation Reconfigurable Manufacturing Systems (RMS), namely, adaptive production systems designed to quickly adjust their configurations to accommodate changes in production requirements, processes, or product designs, are being developed to achieve flexible and intelligent industrial automation [1]. According to the Fourth Industrial Revolution, significant improvements have been made in both hardware (HW), which comprises physical components such as robotic arms, conveyors, sensors, and computing devices used for product handling, data acquisition, processing, and communication; and software (SW), which includes computer programs, algorithms, and applications that control the system operation. However, despite Industry 4.0 (I4.0) envisioning the use of

modern smart technologies to automate traditional manufacturing and industrial practices, automation is still not capable of tackling challenging tasks in variable manufacturing environments, primarily executing pre-programmed jobs. Therefore, adaptability to target changes and autonomy are the next trend in the transition towards flexible industrial automation [2, 3].

The virtual programming of flexible physical systems can be framed in a broader field of research that is particularly relevant for the next generation of reconfigurable industrial robots that exhibit adaptable behavior in cyber-physical manufacturing environments [3, 4]. Cyber-Physical Systems (CPSs), which integrate heterogeneous software and hardware components, play a crucial role in reconfigurable systems, attracting researchers in various areas, particularly in manufacturing. This work focuses on a subcategory of CPS that relies on digital models in the field of industrial vision systems for inspection purposes. It includes the set of digital Computer-Aided (CA) information, assisting in reconfiguring a physical copy from the virtual version. Analogies to other flexible machines and systems are also presented [5].

Vision systems belong to the interdisciplinary fields of system engineering or, more generally, automation [6], and incrementally benefit from the fast progress in electronics

✉ Francesco Lupi
francesco.lupi@phd.unipi.it

¹ Department of Information Engineering (DII), University of Pisa, 56122 Pisa, Italy

² Department of Civil and Industrial Engineering (DICI), University of Pisa, 56122 Pisa, Italy

(e.g., sensors, processors, specialized cards) and computer science (e.g., algorithms) [7]. Despite the considerable level of automation introduced by these solutions, and the integration with I4.0 cutting-edge technologies such as the Internet of Things (IoT), enabling device connectivity [8]; autonomous robots, capable of self-operation [2]; big data and analytics, for data-driven insights [9]; Additive Manufacturing (AM), using 3D printing [10, 11]; and Augmented Reality (AR), enhancing real-world experience [12], the reconfiguration activities (i.e., hardware setup and software parametrization) are still human-intensive based [2, 13, 14].

Among the main limitations in applying Vision Inspection Systems (VIS, i.e., automated systems using HW and SW for contactless inspecting products through electromagnetic sensing) in industrial environments as an alternative to manual visual inspection is the high degree of HW adaptability required to cover different products (e.g., the wide variability of products' characteristics and defect types). Concerning the hardware side, most VIS are designed ad hoc for a given application, with low degree of flexibility, often presenting rigid architectures (i.e., not mobile components) or significant setup time. On the

software side, a major limitation is the manual low-coding activities required from skilled staff to adapt specific algorithms and thresholds via reparameterization [15]. However, owing to a lack of expertise in programming, traditional programming methods are challenging for operators. As a result, there is a growing demand to make SW programming easier for the new generation of reconfigurable VIS that require flexible HW/SW architecture as an enabling factor to pursue the aimed reconfigurability.

The novelty of the current work and main contribution consists in the following:

- i) proposing a framework for designing a Flexible Visual Inspection System (FVIS—an adaptable inspection system) that can be easily reconfigured via a Reconfiguration Support System (RSS—a system supporting reconfiguration) according to a variant approach inspired by the variant process planning, originally defined for Computer-Aided Process or Inspection Planning (CAPP/CAIP) [16].
- ii) validating the proposed framework in a real industrial setting with a high degree of complexity (defect types

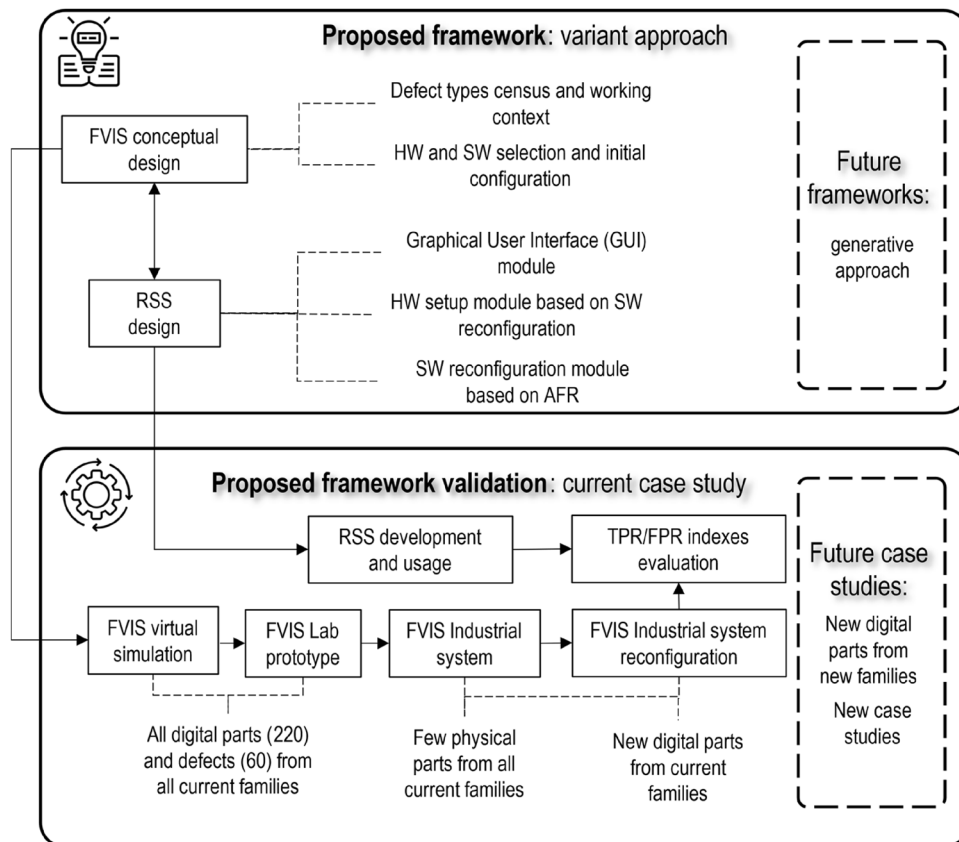


Fig. 1 The two contributions of the current work. Proposed framework definition (Section 2: the proposed framework: FVIS enabling RSS) and proposed framework validation (Section 3: experiments). Future directions are highlighted in dotted box for both the contributions

and products variance) to stress-test its actual flexibility and reconfigurability capabilities.

The two contributions of the paper are illustrated in Fig. 1 along with some promising future developments. Specifically, the theoretical components of the proposed framework are outlined on the top side of Fig. 1, while the case study validation setup for testing the practical implementation of the proposed framework and measuring some indicators to showcase the achieved results is shown in the bottom side of Fig. 1.

With a particular focus on the proposed framework, it is built on RSS, which is in turn based on the FVIS architecture. RSS is rooted in the computer-aided extraction of (digital) information from different sources of data (i.e., actual images, Computer-Aided Design (CAD) model information, and 3D model rendered images) by Automatic Feature Recognition (AFR), which is a process for recognizing geometrical and topological features automatically. The aim is to allow non-skilled operators to easily reconfigure the FVIS by extracting parameters and generating a Reconfiguration (ReCo) file (i.e., file containing both HW parameters, for example, sensors, and SW parameters of the vision inspection algorithms), through the Graphical User Interface (GUI) of the RSS. On the other hand, FVIS conceptual design concerns the census of all possible defect types to be inspected, the HW and SW initial selection and configuration according to the working context.

The effectiveness of this proposed framework is demonstrated through validation on a real inspection case study where the FVIS and RSS have been designed and implemented in accordance with the proposed framework. The validation process starts by accounting for all digital parts (220) and defect types (60) from 12 families. The first step concerns the FVIS virtual simulation, then the FVIS lab prototype is developed. Secondly, the FVIS industrial system is implemented in the actual plant manufacturing setting and run over 4 years. The implemented solution is evaluated via a quantitative index on the rate of true positive and false positive detection.

The proposed framework draws inspiration from other well-known industrial automation frameworks, which are reviewed in Table 1. A comparative overview of characteristics (rows) among different classes of manufacturing systems (columns) showcases representative differences and similarities of the proposed framework for visual inspection (in the last column).

Particular attention is given to flexibility enablers, which, as defined in the recent literature, encompass four critical properties of manufacturing systems with low ramp-up/reconfigurability cost, time, and effort: (1) modularity—major components are modular, (2) integrability—hardware and software modules provide interfaces for easy integration, (3) customization—equipment is designed to meet the specifications for processing a specific product family, and

(4) convertibility—hardware or software changes can be made quickly [3]. The percentages reported in the table are based on the level of satisfaction of these four enablers for each system.

Another important attribute concerning flexibility is the type of machine or system adopted. Reconfigurable Machines (RMs—machines that can adapt themselves for a set of operations to produce within a specific product family) form the foundation for RMSs. According to [17], machines used in a Dedicated Manufacturing System (DMS—rigid automation) are typically specialized for a specific, mass-produced part. They are designed to carry out a single operation with high precision, consistency, and efficiency making them relatively simple and cost-effective. In contrast, machines used in a Flexible Manufacturing System (FMS, e.g., flexible manufacturing cell) are designed for versatility and can produce a variety of parts by altering their computer programs and flexible HW. These Numerically Controlled (NC—computer programmable) machines are typically more expensive due to the need for high production volumes [17]. RMs falls between DMS and FMS machines, providing limited, specialized flexibility that allows for cost savings while enabling quick adaptation to changing product designs [17]. Unlike a standard VIS and similar to an RMS/FMS, the FVIS includes hard automation NC motorized components, such as cameras, lighting sources, grippers, and tailstocks, which can be reconfigured via software using a specific ReCo file generated through the AFR methods implemented in the RSS.

Other attributes, such as design principles (i.e., requirements the systems need to address), output (physical product/product's information resulting from the system), setup (approach used to setup the system), reprogramming/process planning (techniques adopted to reprogram the system), or output of the reprogramming (hardware/tool adopted to reprogram the system) are also considered in Table 1.

Table 2 reports the acronyms, full names, and definitions used throughout the paper, while the logical structure of the paper is outlined as follows: Section 1.1 defines the VIS background and core concepts; Section 1.2 introduces the main works related to Computer-Aided Vision Inspection (CAVI—adoption of computer-based tools, algorithms, and image processing techniques to automate and optimize visual inspection tasks), highlighting the need for a FVIS to achieve reconfigurability. In Section 1.3, a detailed review of the main techniques adopted in the literature for AFR via actual images, 3D CAD data, and synthetic images is provided. Section 2 details the proposed framework for the design of FVIS and RSS. Section 3 presents the industrial case study where the proposed framework has been applied. Section 4 reports the results both for the general framework (top-down) and for the case study (bottom-up), and Section 5 concludes the present work.

Table 1 Synopsis of manufacturing systems' (columns) attributes (rows). The last column refers to the general attributes of the proposed framework in the current work

	Rigid automation	Manufacturing unit	Manufacturing machine	Handling system	Measurement or inspection system	Proposed framework: visual inspection
Machine or system	Dedicated manufacturing system (e.g., transfer line)	Reconfigurable/flexible manufacturing system/cell	Computerized numerical controlled machines	Industrial robot (robotic arm)	Coordinate measuring machine (inspection system for metrology)	Flexible vision inspection system (other defects)
Acronym	DMS	RMS/FMS	CNC	IR	CMM	FVIS
Design principles	Designed around a specific part that is mass produced	RMS: designed around a particular part family FMS: designed to perform different products and different part families	Designed to perform most operations in a flexible manner	Designed to perform most operations in a flexible manner	Designed to perform most operations in a flexible manner	Designed around a particular part family (customized flexibility)
Output	Manufactured part/ assembled product	Manufactured part/ assembled product	Machined part	Manipulated, machined, painted part/assembled product	Inspected part, measured part	Inspected part
Flexibility enablers (i.e., modularity, integrability, customization, convertibility)	Hard automation (25%: customization)	Components integrated in a programmable environment (e.g., PLC, AGV, CNC, pallet, robot, vision system, CMM) (100%: all the four enablers)	Tool changer, pallet system, NC axes (50%: customization and convertibility)	Programmable joints and actuators, end effector, digital I/O and controller (50%: customization and convertibility)	NC axes, probes, optical sensors (50%: customization and convertibility)	Integrated NC HW components (e.g., cameras, lights, grippers, tailstocks) in a programmable environment (100%: all the four enablers)
Setup (i.e., reconfigure via HW, rearrange via SW)	Via HW	Via SW	Via SW	Via SW	Via SW	Via SW
Reprogramming/process planning (add new part) tool*	Not possible	Computer-Aided Manufacturing (CAM)	CAPP+CAM	Software Development Kit (SDK)	CAIP	Computer-Aided Vision Inspection (CAVI)
Reprogramming output**	None	NC for each machine and PLC/PC SW	(STEP-)NC part program, G file	Robot program (not standardized)	NC, STEP-NC	Reconfiguration (ReCo) file

*Input new parts of same family/ies: reprogram (part) + setup (machine or system), **input new parts of new family/ies: reconfigure (machine or system) + reprogram (part) + setup (machine or system)

Table 2 Glossary of acronyms in alphabetical order and their extended definitions

Acronym	Full name	Definition
AFR	Automatic Features Recognition	Computer-based process that automatically identifies and extracts specific geometric or topological features from digital data (e.g., images, CAD models)
AGG	Attributed Adjacency Graph	Graph data structure that contains both geometric and topological information. It represents the surfaces of the part as nodes connected by lines
AGV	Automated Guided Vehicle	A mobile robot equipped with navigation and guidance systems to transport materials or goods autonomously within industrial or warehouse environments
AI	Artificial Intelligence	Branch of computer science that aims to create systems that can learn from data and perform tasks that typically require human intelligence
AM	Additive Manufacturing	Production process creating three-dimensional objects by adding successive layers of material under computer control
AR	Augmented Reality	Technology that overlays digital information, such as images, text, or 3D models, onto the real-world environment through display
BOM	Bill Of Materials	Comprehensive list of raw materials, components, and parts required for a product. It facilitates production planning, procurement, and assembly processes
CA	Computer-Aided	Digital information
CAD	Computer-Aided Design	Digital tools to create detailed and precise 2D or 3D models of products or structures
CAM	Computer-Aided Manufacturing	Digital approach that uses computer software and automation converting CAD into instructions, planning and control for manufacturing machinery
CAIP	Computer-Aided Inspection Planning	Digital approach for optimizing inspection processes by creating strategies, paths, and techniques to ensure accurate quality control during manufacturing
CAPP	Computer-Aided Production Planning	Digital approach for optimizing manufacturing by generating instructions, tool selection, and operation sequences for efficient production
CAVIS	Computer-Aided Vision Inspection/System	Technology that utilizes computer-based tools, algorithms, and image processing techniques to automate and optimize visual inspection tasks in various industries
CMM	Coordinate Measuring Machine	Specialized metrology device used to measure the physical geometrical characteristics of an object through a probe
CNN	Convolutional Neural Network	Specialized type of architecture commonly used in computer vision tasks designed to adaptively learn spatial hierarchies of features from input images
CPSs	Cyber-Physical Systems	Integration of heterogeneous software and hardware components
CVJ	Constant Velocity Joints	CV joints allow smooth power transmission while accommodating suspension movement, resulting in improved vehicle performance and comfort
DMS	Dedicated Manufacturing System	Specialized production system designed to manufacture a specific, mass-produced part with high precision, consistency, and efficiency
FMS	Flexible Manufacturing System	Manufacturing system designed for versatility and adaptability using NC machines and automated material handling systems to process a variety of products
FN	False Negative	Number of defective products that have been incorrectly not classified as defective
FoV	Field of View	Observable area that a camera or sensor can capture, measured in degrees. It ranges from the center to the edges of the image
FP	False Positive	Number of non-defective products that have been incorrectly classified as defective (false alarm)
FPR	False Positive Rate	$FPR = FP / (FP + TN)$
FVIS	Flexible Vision Inspection System	Inspection system that provides adaptability for different vision inspection tasks
GUI	Graphical User Interface	Visual interface that allows users to interact with SW applications or computer systems through graphical elements such as icons, buttons, menus, and windows. Also addressed as Human Machine Interface (HMI), especially in commercial contexts

Table 2 (continued)

Acronym	Full name	Definition
HW	Hardware	Physical components, such as robotic arms, sensors, and computing devices, used for product handling, data acquisition, processing, and communication
I4.0	Industry 4.0	Transformative approach involving the integration of advanced technologies, to create smart, connected, and autonomous industrial systems
IGES	Initial Graphics Exchange Specification	File format enabling the exchange of 2D and 3D graphical data between different CAD systems
IoT	Internet of Things	Network of physical objects, devices, and other items embedded with sensors, software, and connectivity
IR	Industrial Robot	Programmable robots equipped with mechanical arms, sensors, and computer control systems enabling carry out repetitive tasks with precision and efficiency
ISO	International Organization for Standardization	Organization that develops and publishes international standards to ensure quality, safety, and efficiency in various industries worldwide
MAAM	Multi-Attributed Adjacency Matrix	Visual approach used to enhance recognition, variation of the AAG, to represent the relationships between different attributes associated with the surfaces of the part
ML	Machine Learning	Sub-set of AI that develops algorithms for computers to learn from data, make predictions, and identify patterns without explicit programming
CNC/NC	Computerized Numerical Control	Automated control of machine tools and other industrial equipment using numerical data
PBR	Physically Based Rendering	Computer graphics technique that creates realistic images by simulating real-world light interactions with materials
PC	Personal Computer	It includes components like CPU, memory, storage, and input/output devices
PDM	Product Data Management	System for efficiently managing product-related information throughout its lifecycle. It centralizes data like CAD files, BOMs, and specifications
PLC	Programmable Logic Controller	Industrial digital computer used to automate electromechanical processes, monitor inputs, and control outputs for machines or systems
ReCo file	Reconfiguration File	File containing both HW parameters of, for example, sensors and SW parameters of the vision inspection algorithms
RMs	Reconfigurable Machines	Machines that can adapt themselves for a set of operations to produce within a specific product family
RMS	Reconfigurable Manufacturing Systems	Adaptive production systems designed to quickly adjust their configurations to accommodate changes in production requirements, processes, or product designs
ROI	Regions Of Interest	Specific areas in images selected for further analysis, used in computer vision to optimize resource utilization, and focus on relevant information
RSS	Reconfigurable Support System	System that automatically extracts digital information from various digital sources using AFR, enabling non-skilled operators to easily reconfigure FVIS
STEP-NC	STandard for the Exchange of Product model data—NC	Define data models and protocols for better CNC machining communication, enhancing interoperability between CAD/CAM and CNC machines
STL	Standard Triangulation Language To Layer	3D file format used in 3D printing and CAD. It represents object surfaces with connected triangles, facilitating communication with 3D printers and CAD SW
SMEs	Small and Medium Enterprises	Businesses with limited employees and revenue, but significant contributions to the economy through job creation, innovation, and flexibility in various sectors
SW	Software	Computer programs, algorithms, and applications that control the operation of the system
TN	True Negative	The number of non-defective products that have been correctly classified as non-defective
TP	True Positive	The number of defective products that have been correctly classified as defective
TPR	True Positive Rate	$TPR = TP / (TP + FN)$

Table 2 (continued)

Acronym	Full name	Definition
TQM	Total Quality Management	Management approach aiming for continuous improvement and customer satisfaction, involving process management, and a culture of quality
VBA	Visual Basic for Applications	Integrated programming language enable creating macros and automating tasks, enhancing productivity, and customizing workflows
VIS	Vision Inspection Systems	Automated systems using HW and SW for contactless inspecting products through electromagnetic sensing
XAI	Explainable Artificial Intelligence	Human-understandable explanations for AI decisions, bridging the gap between complex AI algorithms and human comprehension
XML	Extensible Markup Language	Versatile data format for structured data representation. It uses custom tags, enabling human-readable and machine-readable data, facilitating interoperability

1.1 VIS background

Production flexibility and efficiency have become the new challenges in established manufacturing paradigms [18, 19], which have undergone a transition from mass production, based on a push approach, to mass customization, rooted in customer demand [10, 20, 21]. Growing customer expectations have lowered the maximum level of acceptable defects to drastic values in the order of parts per million (ppm). Therefore, for many manufacturing industries, such as automotive production, the role of quality assurance—e.g., Total Quality Management (TQM), Six Sigma, zero-defect manufacturing—has gained a strategic relevance [22].

Although quality management standards rely on the capability of the process to meet certain specifications and statistically ensure the related product conformance, final product inspection remains a central activity for manufacturing processes [23]. According to a definition from the International Organization for Standardization, in ISO 9000, inspection is the process of *determination of conformity, which occurs when a product meets specified requirements* [24]. In this context, human visual inspection is one of the most implemented approaches due to its flexibility (i.e., versatility in inspecting variable products and variable defects) [25]. On the other hand, it is a source of inefficiencies, such as time-consuming, inaccurate, and inconsistent activities due to unavoidable human variability, as well as other human-related physical limitations [23, 26].

Vision systems (also known as artificial or machine vision), which are widely used today to capture images and perform various tasks such as surveillance, guidance, tracking, location, sorting, and 3D reconstruction, in a wide range of application domains, extending from e.g., food processing [27] and environment monitoring [28] to civil construction [29], assisted medicine [30], and pharmaceutical inspection [31], have become a valuable alternative for automatic

contactless inspection (i.e., VIS or automated visual inspection) [13, 32, 33].

VIS can be defined as a set of existing off-the-shelf, commercial, or prototypical electromechanical (i.e., hardware) and information technology (i.e., software) components integrated in a novel way to solve inspection problems. The embedded technology includes contactless sensors in a broad electromagnetic wavelength band (e.g., X-ray, ultraviolet, visible, infrared, microwave, and multispectral).

From a general perspective, VIS essentially relies on the extraction and interpretation of information acquired from digital images and making decisions accordingly. The work proposed by [34] classified automatic visual inspection as a three-step process that can be collapsed into the following two-step process:

- **Image acquisition:** this step involves the hardware and software components required for image acquisition, storage, and processing. As shown in Fig. 2 (left side), the vision hardware mainly includes camera sensors [14], focal lenses, optical filters, port connections, lighting, and handling sub-systems. Software components include embedded computing integrated with vision data management tools and processing algorithms to perform specific visual inspection activities (Fig. 2, right side) [7].
- **Image (pre-/post-) processing and analysis:** once the image is acquired, a variety of visual inspection activities (e.g., surface control, pattern matching, dimension gauging, inner defects check, shape control, and general identification) are performed to extract the desired characteristics from the inspected object [35]. The characteristics are analyzed via context-related algorithms parametrized for the specific case. Among several inspection algorithms, two main categories can be defined, namely, rule-driven and Artificial Intelligence (AI)-driven [13].

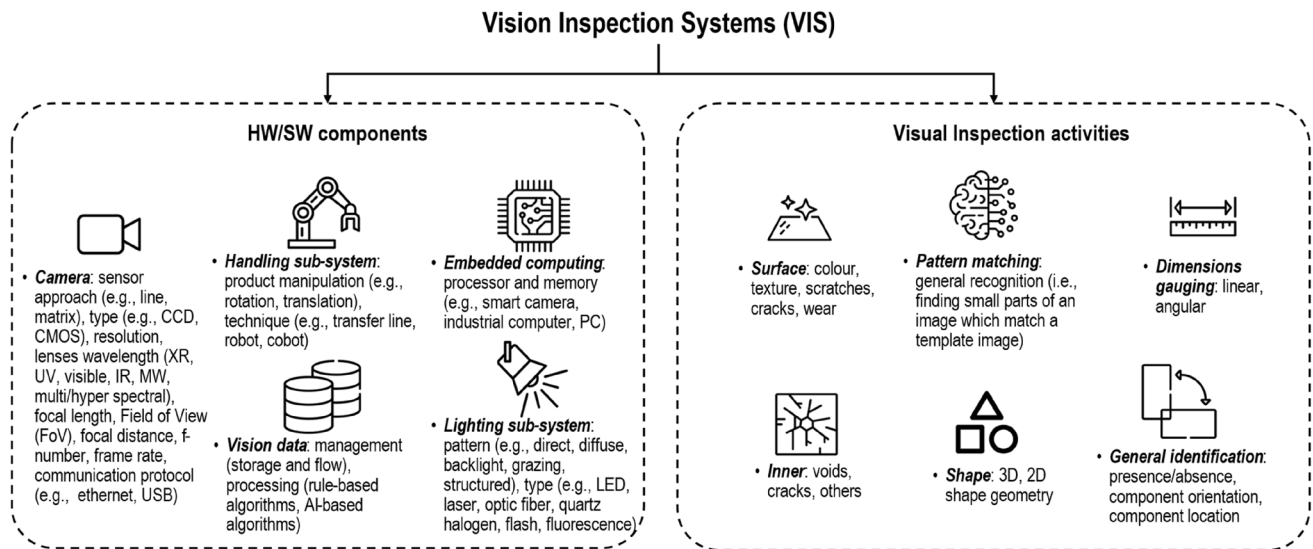


Fig. 2 Vision Inspection Systems (VIS) are vision systems made by conventional or customized HW/SW components and sub-systems (left) integration but specifically designed for inspection activities (right)

The rule-driven category includes classic computer vision algorithms such as color analysis, blob detection, and edge detection, while AI-driven is based on Machine Learning (ML) algorithms able to learn from data, recognize patterns, and make predictions without explicit programming (e.g., Convolutional Neural Network (CNN)) [13].

1.2 Computer-Aided Vision Inspection (CAVI)

Digital tools have brought significant value to the field of CAVI, and the integration of CAD technologies and data management systems such as Product Data Management (PDM—data base containing product-specific information) for offline programming and reconfiguration of vision inspection assets has been a promising research topic since the late 1980s [36]. Over the past decades, research into the reconfiguration of VIS has led to the development of visionary frameworks like the Computer-Aided Vision Inspection System (CAVIS) [37], where the authors propose a solution to overcome the need for specialized skills in the vision system reconfiguration, algorithm definition, and programming phases in assembly contexts. Despite the merits of these theoretical models in conceptualization and initiating CAVI reconfiguration, there have been only a few practical applications on flexible architectures enabling this approach in industry and scientific literature in recent years.

With the advent of I4.0, more powerful, flexible, and affordable hardware and software solutions have become available on the market. As a result, many Small and Medium Enterprises (SMEs) and companies facing high

technological and automation barriers are now transitioning to automated systems. In this rapidly evolving automation environment, there is a growing interest in introducing and implementing new technological concepts to fully interconnect production elements by bridging the physical and digital worlds [38], creating new opportunities for CAVI reconfigurable systems [32]. As noted by [13], the global machine vision system market is expected to grow rapidly from \$8.6 billion in 2020 to \$17.7 billion by 2027, driving the demand for improved performance and new capabilities such as reconfigurability.

Consequently, there is a growing demand for flexible and reconfigurable systems capable of eliminating the laborious and time-consuming hardware setup and software programming work [39]. In this context, parametric offline reprogramming is gaining increasing attention in various robot-based activities [15, 40–42]. Among robot-supported visual inspections, a new approach for specifying the spatial and temporal trajectory of visual inspection based on CAD models of workpieces has been proposed [43]. In this work, the authors claim the ability to reduce the effort required from the engineer to setup vision hardware, a task that remains a non-value added and tedious for production line engineers [43]. Similar to [43], our work considers CAD data for the system setup, offering a higher degree of flexibility enabled by NC programmable hardware components.

Another related work based on an adaptable CAVI system can be found in [44]. In their work, the authors propose a flexible hardware solution comprising standardized (e.g., lights, robots) and reconfigurable (i.e., handling system) components, integrated with flexible image processing

algorithms using CNN and Explainable Artificial Intelligence (XAI), along with a knowledge transfer system for software parametrization support.

Despite the close relationship with the general framework proposed in [44], we aim to provide a more general model not tied to a specific HW/SW architecture. To validate the proposed framework and present an industrial case study, a successful real-world implementation in the automotive industry is showcased.

1.3 Automated Feature Recognition (AFR) methods

The ReCo file serves as the link between the digital and physical worlds. It is used to reconfigure the NC hardware components of the VIS and contains the parameters for the inspection algorithms. To expedite the parametrization of the ReCo file, a virtual environment based on AFR is adopted to minimize the user's programming effort and facilitate the process of CAVI reprogramming.

As shown in Fig. 3, a trade-off emerges when embracing an increasing virtual AFR approach in the effort to calibrate the cyber-twinning entity to mimic the physical environment. As a matter of fact, creating an almost identical copy of a physical counterpart (i.e., with a very high degree of accuracy) is a demanding task. The whole process of developing highly accurate computer models and simulation of processes is time-consuming and demands significant investments [45]. This underscores the importance for industries to conduct a cost-benefit analysis before implementing a digitally twinned approach over another.

In the following Sections (1.3.1–1.3.3), we will review three main categories of possible approaches for extracting digitally twinned vision features (to provide input for the ReCo file parameters) for the reconfiguration of

the FVIS. As illustrated in Fig. 3, the first AFR approach, examined in Section 1.3.1, pertains to extracting digital information directly from actual images of the authentic product, captured on-site by the actual camera under operational lighting conditions. The second approach, treated in Section 1.3.2, is related to extracting geometrical information using a three-dimensional (3D) CAD virtual product model. The third approach, reviewed in Section 1.3.3, utilizes synthetic rendered images of the virtual product as source input. All these categories can be implemented in the proposed framework.

1.3.1 AFR via actual images data

In the field of industrial vision inspection [7], employing authentic images to parametrize the vision algorithms is a conventional approach facilitated by camera manufacturers, who embed their products with dedicated image analysis toolboxes or SW extensions [46]. Parametrization can be accomplished through user-friendly interfaces that enable the selection of Regions Of Interest (ROI) using drag and drop functionality, among other features.

On the other hand, more advanced approaches (e.g., deep learning-based) commonly employed for surface defects detection can be adopted to partially or fully automate the classification of features and extract information for parameterizing the ReCo file [47].

Undoubtedly, these approaches reduce the complexity of transitioning from the virtual environment to the real production environment. As a result, the reference system remains the same, and there is no need to consider additional lens distortion or calibration [48]. However, a significant issue persists: the impact on the production line flow.

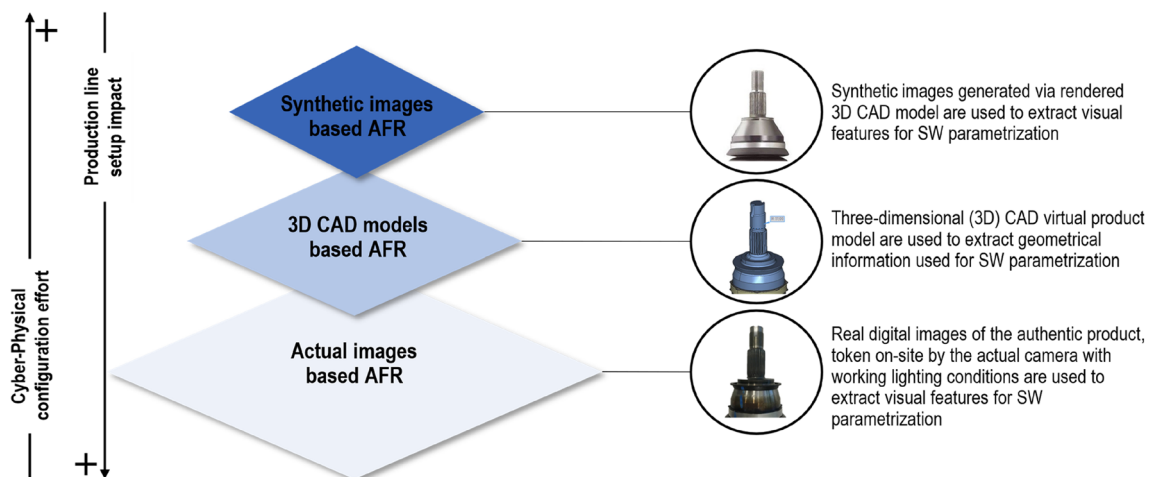


Fig. 3 Automatic Features Recognition (AFR) techniques at different levels of cyber-physical configuration effort and production line setup impact. A trade-off is needed between cyber-physical configuration effort and production line setup impact

Due to the time required for setting up the actual vision system, offline parametrization is not entirely offline since real images must be collected using the real machine, which interrupts the production flow. Therefore, the literature has explored more advanced solutions with zero impact on shop floor productivity and fully offline parameterization. Existing approaches are reviewed in the subsequent two sections.

1.3.2 AFR via 3D CAD data

CAD-based AFR methods are aimed to extract geometric features (e.g., holes, slots, pockets, bushings) and tolerance specifications (e.g., dimensions, parallelism, concentricity) from 3D models without the necessity of the real product [49, 50]. These methods are typically adopted for extracting specific features for manufacturing purposes but can also be used to extract inspection-related or vision-specific features.

AFR techniques for CAD models are well-established, and research has been ongoing in this field for decades [51–55]. CAD has been instrumental in streamlining the design process for new products, using recurring or scalable elements [56, 57] and group technology

[58]. The use of CAD, CAPP, Computer-Aided Manufacturing (CAM), and CAIP has significantly increased over the years, intensifying the already existing link between design and production fostered by the AFR process [54, 59, 60]. CAPP, CAM, and CAIP tools often involve the “decomposition” or “extraction” of distinct sets of faces to be machined or inspected, defining processing information such as the selection of cutting/inspection tools, machining/inspection sequence and strategy [61, 62]. AFR techniques are the first and most crucial step in transforming the 3D CAD data into processable information for manufacturing (i.e., CAPP, CAM) or inspection (i.e., CAIP) and can be readily adapted for CAVI.

Table 3 reports the main AFR techniques for 3D CAD data retrieved from recent state-of-the-art reviews [53–55].

Although many established CAD software programs now incorporate generic functions for AFR [63, 64], evidence of their industrial implementation is lacking [55]. Furthermore, with the increasing complexity of products and processes, the need to develop new, efficient strategies for feature extraction remains an open topic [53].

Table 3 Main AFR techniques for 3D CAD data summarized from [53–55] along with a synthetic description

AFR technique	Summary description
Syntactic-based	These approaches initiate the AFR process by translating shape features into a set of primitives, using language descriptors (i.e., grammatical rules for analyzing the textual information stored in the CAD files). If the textual data matches the grammar rules in the library, then an association with a particular class of features is made. This method was traditionally used to extract feature from 2D shapes, but it has been largely supplanted by more advanced and general techniques in recent years
Rule-based	These techniques are a generalization of syntactic-based methods since they operate with more general rules and with 3D models. The generalization is achieved through expert systems that rely on IF-THEN logic rules. Expert systems are defined by a knowledge base that contain a set of rules extracted from domain experts and translated into a simple form (IF B1 B2 B3, THEN A), a collection of questions, and general facts. An inference engine is used to derive the conclusion (e.g., if a predefined feature condition is satisfied, the corresponding area on the part is recognized as a feature) based on the facts provided during the interaction with a non-expert user
Graph-based	These techniques use graph elements that include both geometric and topological information. These elements, known as Attributed Adjacency Graph (AAG), are typically created from the component’s Boundary Representation (B-Rep). AAG is a graph composed of nodes representing the piece’s surfaces that are connected by lines. When two surfaces of the part are adjacent, the corresponding nodes in the graph are joined by a line that has a value “0” if the surfaces have a concave adjacency relationship and a value “1” if they have a convex relationship. Another visual approach to improve recognition is a Multi-Attributed Adjacency Matrix (MAAM)
Hint-based	The hint-based approach was developed to address issues related to the arbitrary interaction between features. This method combines visual and logical approaches using topological and geometrical information as clues for recognizing features on the part. These clues may include traces left by tools during the machining process, which provide additional information to aid in the recognition of complex features
Volumetric decomposition	These methods are based on the decomposing the volume of a component into smaller sub-volumes, which can be further divided into even smaller sub-volumes. One example of this approach involves determining a convex shell polyhedral and evaluating the difference in volume by comparing the part itself to the sum of its sub-volumes
Artificial Neural Networks (ANN)	AFR systems using neural networks have also gained significant attention in recent years. Through standard supervised algorithms, a set of feature examples is presented to the neural network, which is trained to recognize the geometrical and topological patterns associated with a given feature employing a black-box approach

As previously highlighted, the main advantage of adopting AFR based on 3D CAD data lies in the extraction of geometric features from a digital model. This enables the parametrization of the ReCo file without disrupting the production line. Additionally, this approach is significantly faster than the one outlined in Section 1.3.1, as it automates numerous activities associated with manually setting the desired tolerance values embedded in the 3D CAD model, which are automatically extracted.

However, a fully digital environment also introduces challenges in calibrating the real vision system. The extracted information must be converted to account for real-world lens distortion and other calibration factors. This requires additional calibration steps to migrate the parameters from the digital to the actual environment. Another obstacle to the widespread application of CAD-based automation in the industry is the compatibility among different commercial CAD file formats and the need for a neutral file format [65]. To address this, file formats such as the Standard Triangulation Language To Layer (STL), STandard for the Exchange of Product model data (STEP), or the Initial Graphics Exchange Specification (IGES) have been defined to unify and provide a common ground for developing new methods and algorithms [66–68].

1.3.3 AFR via synthetic images

Photorealistic rendering is a technique that allows the generation of images, starting from a digital 3D model, which are challenging to distinguish from actual photographs [69]. A significant segment of the realistic rendering market (e.g., architectural, automotive, and product visualization in general) has relied on Physically Based Rendering (PBR) technology [70].

The utilization of synthetic rendered images for the extraction of valuable information for VIS reconfiguration was first proposed within the broader CAVIS framework by Lanzetta et al. [37]. As an application, CAD-rendered images of the product have been successfully employed to extract the ROIs and set other vision system parameters [71], or to train supervised vision inspection algorithms in the field of Artificial Intelligence (AI) [72].

The use of synthetic rendered images as a hybrid approach allows for combining the benefits of virtual CAD data and real digital image data (Section 1.3.1 and 1.3.2). This advantage can be reached by simulating the camera parameters and settings within a fully digitally twinned reality [73, 74], reducing the gap between the digital model and the actual visual environment. In this case, no extra calibrations are needed to migrate the information retrieved from the rendered image to the real VIS.

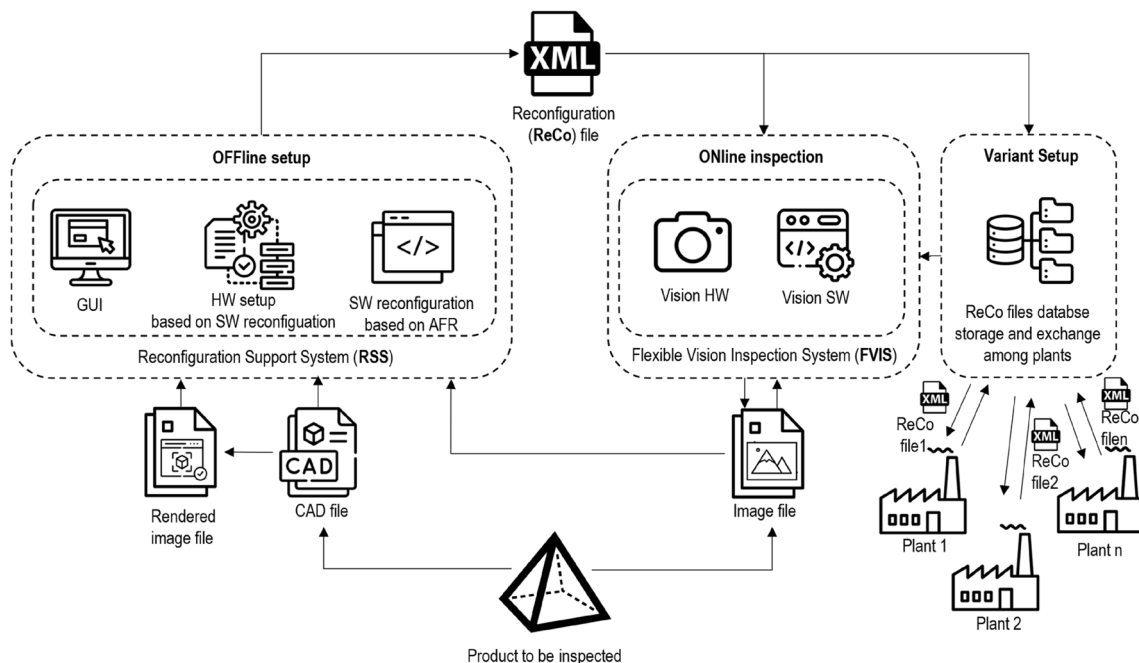


Fig. 4 Flexible Vision Inspection System (FVIS) and Reconfigurable Support System (RSS) framework

Nonetheless, despite the significant advancement of CAD systems over the past two decades, the use of synthetic images for feature extraction remains restricted due to the considerable number of parameters required in the rendering setup process (e.g., material, texture, lights, environment, camera specifications), as well as the computational power demands.

2 The proposed framework: FVIS enabling RSS

The proposed framework is illustrated in Fig. 4. At the bottom, the product to be inspected is the source for three different types of data, which allow for the three different AFR techniques presented earlier (Section 1.3.1–1.3.3). Depending on the selected approach, actual images, CAD or rendered images files are the input for the RSS, which enables offline reconfiguration via ReCo file using three main modules: the Graphical User Interface (GUI), hardware setup, and software reconfiguration.

During the AFR step, a ReCo file (e.g., in Extensible Markup Language (XML) format) is generated, and the VIS hardware setup and algorithms parameters are updated accordingly.

The online inspection begins taking images via the vision hardware, pre-processing, and analyzing them via the inspection software. The ReCo file can be stored in a shared database among plants, expediting the reconfiguration process for various products, following a variant approach inspired by the variant process planning defined for CAPP. The premise for this feature is that similar parts require a similar inspection task and, consequently, a similar ReCo file (i.e., similar HW/SW configuration).

The main two steps of the system design are presented as follows: Section 2.1 recalls the preliminary activities that should be carried out in the conceptual design of the FVIS. Section 2.2 delves into the offline reconfiguration process enhanced by the AFR approach implemented into the proposed RSS for the generation of ReCo files.

2.1 Flexible Vision Inspection System (FVIS) conceptual design

The FVIS conceptual design focuses on designing a flexible hardware and software configuration (right-hand side of Fig. 4, central-top side of Fig. 1) tailored to the product's defects to be inspected, physical and geometric characteristics of products and families, as well as environmental constraints (e.g., inspection context and production rate). It is important to note that this section solely serves to define the conceptual design methodology, which includes the two steps as follows:

- **Step 1 (defect types census and working context):** to establish a comprehensive list of defects, a combination of top-down and bottom-up approaches should be employed.

Top-down generation (i.e., product families database census): PDM exploration and products/families census can be easily achieved by retrieving the information stored into the CAD files. The main characteristics and components of each type, for product families, can be analyzed by checking the Bill Of Materials (BOMs). This procedure facilitates the generation of a list of possible related defects.

Bottom-up generation (i.e., historical data on defects): quality assurance records, derived from the experience with similar products, explicit requests from customers, inspection policies of suppliers, and raw material can be used as valuable historical data to define already registered defects.

The culmination of this process involves aggregating all the top-down and bottom-up generated defects. The list is further refined through cleaning (i.e., duplicate removal), and summarizing activities. Additionally, the working environment (e.g., physical variables such as ambient light, vibrations and atmospheric agents like dust, humidity) and production constraints (e.g., productivity, accuracy) must be considered. This Step 1 is inspired from the first design principle for a reconfigurable machine proposed by [17], in which “a reconfigurable machine is designed around a specific part family of products”. It also aligns with principle 2 for customized flexibility, as well as principle 3 for easy and rapid convertibility.

- **Step 2 (HW and SW selection and initial configuration):** the defects list defined in Step 1 must be processed to establish a set of defect clusters that will be adopted for hardware and software components selection.

The hardware and software must be selected based on the principle of maximum flexibility, including the use of NC motorized axes and PLC for hardware setup and control (e.g., robotic arm, moving cameras, and adjustable lighting conditions). Vision commercial libraries or custom-developed inspection algorithms can be implemented for software inspection. The flexibility can be associated with the coverage of each defect to be inspected by using different configurations of hardware and software components. Once the initial HW/SW configuration is selected and integrated into a flexible architecture (i.e., FVIS), individual features and attributes to be inspected will have its corresponding inspection hardware and software algorithm defined.

2.2 Reconfiguration Support System (RSS)

The frequent turnover of products resulting from the mix of components amplifies the need for reconfiguration. In common practice, even a straightforward reconfiguration for changeover of VIS requires specialized staff. Minor software modifications to ROIs or threshold algorithms are generally considered a complex task to outsource, except in the rare case of specialized internal staff possessing programming skills. Similarly, concerning the hardware setup, tasks like repositioning a camera or adjusting a light source and subsequent calibration require specialized staff if the system lacks proper flexibility features. This deficiency often leads to the necessity of redesigning the system and incurring substantial adaptation costs.

According to the FVIS architecture defined in Section 2.1, a user-friendly RSS is proposed as the second ingredient to enable a fully flexible framework. The graphical representation of the offline flexible reconfiguration process facilitated through the proposed RSS is presented on the left-hand side of Fig. 4 and central-top side of Fig. 1. The flexibility aspect is the main innovation of the proposed solution because it drastically reduces the FVIS programming time

(Section 1.1). In pursuit of this objective, the RSS encompasses three primary modules:

- **SW reconfiguration module:** this module is responsible for acquiring specific problem knowledge for ReCo file parametrization. It accomplishes this by extracting pertinent information, including inspection algorithm-related parameters (e.g., ROIs and thresholds) and NC hardware component-related parameters (such as actuator and NC motor settings), leveraging the AFR techniques implemented into the GUI. This information is then saved in the ReCo file.
- **HW setup module:** this module focuses on integrating the RSS reconfiguration commands stored in the ReCo file into the controller employed for NC programming. This allows for automatic changes in screw lengths through actuators for cameras, lighting, grippers, and tailstocks positions, as well as moving the robotic arm or other flexible mechatronics hardware via the SW interface.
- **Graphical User Interface (GUI):** it serves as the interface that assists users, providing information and describing the AFR process through digital support. The

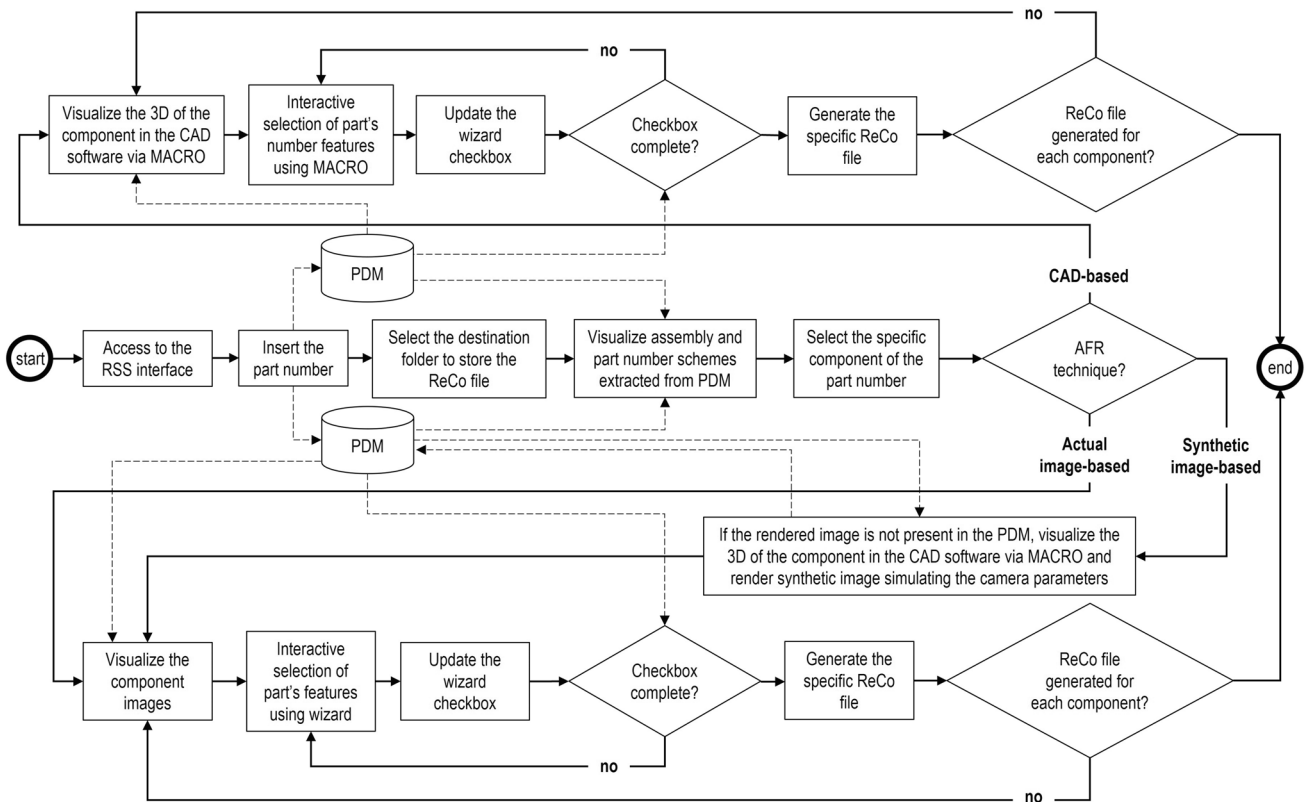


Fig. 5 Procedural workflow of the RSS GUI. Information flow is depicted with dotted lines. Solid lines indicates the logic flow of the activities (represented by squares). Decision steps are represented via diamond shapes. The central section of the figure depicts common

activities. After the AFR technique branch, the upper side represent CAD-based approach, while the lower side represents the actual/rendered image-based implementation

GUI also enables operators to add, update, and verify parameters stored in the ReCo file to optimize the visual inspection process. Additionally, the GUI is utilized during the inspection process to display defect reports to the operators on the production line. These functionalities can include explainable AI (XAI) models, which offer insight into how the inspection system reached its conclusions, as well as visualization tools to help the operator understand the inspection results. The XAI functionalities assist operators in understanding the detection rationale for a specific defect, its location, and how it can be managed according to the quality policies. The requirement for XAI functionalities can vary depending on the selected inspection algorithm (i.e., rule-driven versus AI-driven).

The main output from the reconfiguration process is a ReCo file, which is generated by the RSS in an interoperable format, such as XML, and is easily manageable by conventional operating system libraries. This file is used to update the FVIS configuration, effectively acting as an integrated AFR tool. Figure 5 offers a simplified visual representation of the workflow involving GUI interaction during AFR tasks. The three AFR approaches that are considered in the general framework are highlighted. Regardless of the chosen AFR approach, the part number information (such as assembly, schemes, and actual images) must be previously stored in PDM, even for new (i.e., never inspected) part numbers and related components.

The image-based (actual and synthetic) approach of the RSS AFR (as shown in the central and bottom side of Fig. 5) consists of a wizard interface to define the ROI and the operational parameters by selecting predefined points, lines, and areas on an image taken by the actual camera or from rendered images. As reviewed in Section 1.3, the strength of the actual images approach lies in operating on real images, which guarantees compensation for optical distortions. Additionally, it offers the possibility of calibrating the system by only computing the mm/pixel ratio for each component. In contrast, synthetic rendered images require more effort in the photorealistic rendering process to approximate actual images directly from CAD data; however, they ensure an offline process solution. Once the image (actual or rendered) is obtained, a customized interface containing a guided wizard procedure is utilized to parametrize the ReCo file. This method requires several steps, including searching for images stored in the PDM (if any), or imposing an online manual activity to take images of the part/generate the rendered images.

An alternative approach for offline reconfiguration is based on AFR directly from CAD files (central and upper side of Fig. 5). Currently, some commercial CAD software include the capability to develop macros using a Visual

Basic for Applications (VBA) engine. This enables the development of a single macro that can be adapted with minimal modifications for use in other CAD environments, particularly when standardized CAD data format (such as STEP and IGES) are employed.

Regardless of the adopted AFR approach, the generated ReCo file contains information about the product's inspection features (e.g., tab slot, dust shield, O-ring) and their attributes (such as tab slot's width and length). Each feature and attribute correspond to an inspection algorithm defined during the FVIS design phase. Alongside the extracted vision features for hardware and software setup, the ReCo file also contains details about the hardware components (e.g., size, resolution of the sensor, focal length of the optics, distance of the optics from the axis, distortion factors, actuators to be activated). This information is used when adopting the CAD-based AFR approach to convert the CAD model dimensions in millimeters into the respective dimensions in pixels required to parametrize the inspection algorithm that processes real images.

Once the new ReCo file is generated, it is stored in the program database, together with all the generated ReCo file (e.g., shared cloud among different production plants), aligning with the I4.0 approach. If a different plant in another location needs to inspect a product with an existing ReCo file, an updated version can be conveniently accessed for direct download and variant adaptation (as shown in the right side of Fig. 4).

Notably, mechanical parts with different functions, such as splines or threads, might exhibit similar appearance and share the same inspection algorithm, given the close relationship between vision features and geometrical features. As a result, it is possible to develop an application that selects an already existing ReCo file from a database (associated with a previously processed part number) with similar vision inspection features. This ReCo file contains configuration parameters closest to those required for industrializing the new code, and it can be adjusted as necessary using a variant approach.

3 Experiment

This section presents the implementation and testing of the proposed framework in a real-world industrial case. The specific case of automotive half shaft production is detailed in Section 3.1. The conceptual design of the FVIS for this case is outlined in Section 3.2, while the implementation of the RSS is presented in Section 3.3.

3.1 Case study background

The development of an FVIS and RSS arose from demanding customer requirements for a global leader in the half shaft production sector, requiring less than 10 ppm of defective products. A half shaft is a component that transfers torque

to a vehicle's wheel using two Constant Velocities joints (CVJ) and splines on each end [75, 76]. Figure 6(a) offers an overview of a typical half shaft product.

Half shafts are safety-critical mechanical components in vehicles, enduring mechanical stresses (e.g., torsion and bending), leading to wear, fatigue, and even fractures. Given their safety-critical nature and the persistent mechanical stresses, stringent quality control measures during manufacturing are essential to prevent defects and failures.

In the specific case, the manufacturer monitored 150 machining and assembly process parameters during half shaft production. However, human errors and defects in supplied parts could only be detected through 100% final product inspections before packaging and shipping (e.g., some hidden defects of the assembled half shaft require product manipulation to achieve a complete inspection).

Manual inspection remains common practice in many robotized assembly lines [77, 78]. Similar to the current case, manual operation is often preferred due to the complexity and variability (production mix) that

complicates automating the inspection process. However, due to human-related physical limitations, operators could only target a few critical defects during the visual inspections. Therefore, a plausible introduction of a FVIS lies in some of the intrinsically lower operators' perception capability and reliability compared to machine vision systems.

This work began in 2016 as an industrial-university cooperation to develop a general-purpose FVIS for quality inspection, reconfigurable via an RSS. Various motivations have pushed this work to the maximum potential. However, a primary driver has been achieving the complete automation of assembly.

The first dilemma was deciding between distributed inspection in the individual assembly stations or a single final inspection, providing feedback on all the previous operations. The initial outcome of this cooperation is an inspection station—online since 2017—at the end of an assembly cell, which can detect assembly or machining defects.

This system has been cloned and operating across various automated and semi-automated international

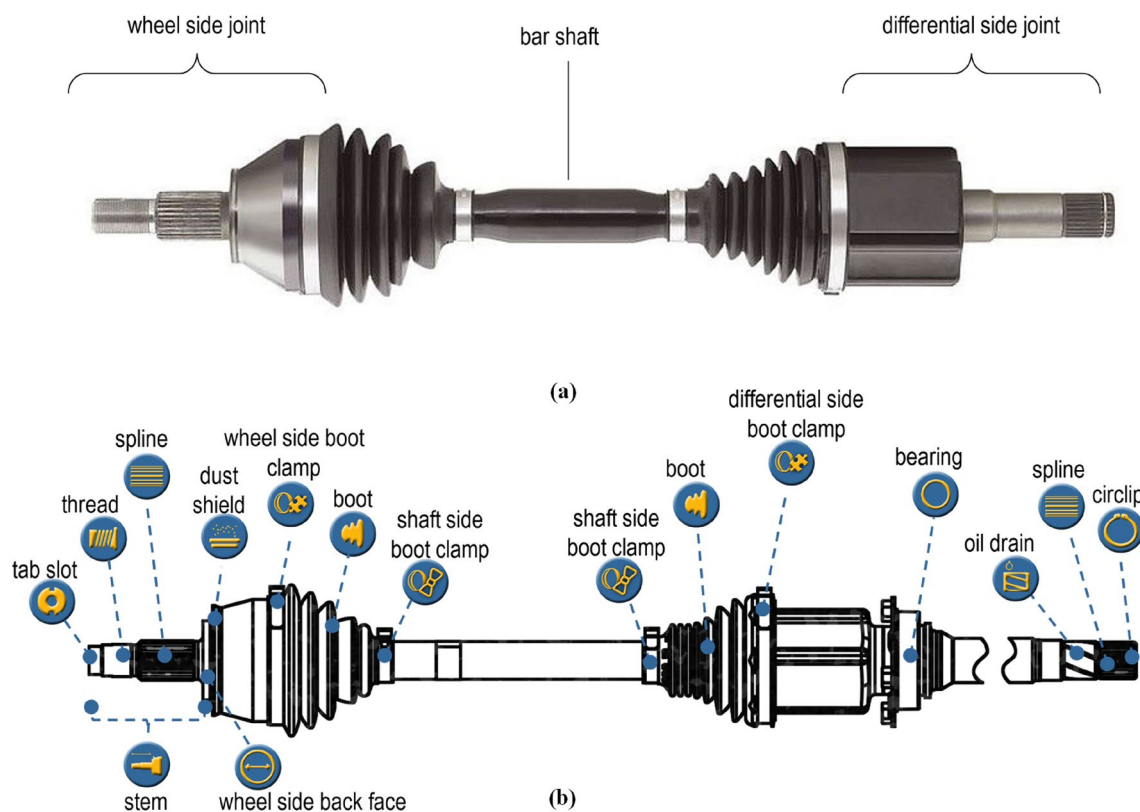


Fig. 6 **a** Half shaft is the dynamic connection between the differential and the driving wheel of a motor vehicle. It transmits power and facilitates steering, adapting to suspension movements, and isolating vibrations. Each half shaft consists of two CVJs. A CVJ is a mechanical coupling ensuring that the output shaft's rotation speed matches the input shaft at any working angle. A bar shaft connects

these two constant velocity joints. **b** Defects identification phase. Starting right to left the labels indicate the following defects: circlip, spline, oil drain, bearing, differential side boot clamp, boot, shaft side boot clamp, shaft side boot clamp, boot, wheel side boot clamp, dust shield, wheel side back face, stem, spline, thread, and tab slot

Table 4 FVIS design options (rows) and defects (columns). For each combination, a score is given as follows: 0 = impossible, 1 = difficult, 2 = good, 3 = optimal. For confidentiality, a sub-set of the 60 defect

types investigated is shown. The last column summarizes the average score for each design option (green scores are the selected) for the selection and clustering of solutions suitable for multiple needs

		Tab slot presence	Tab slot symmetry	Thread no chips	Spline no dent	Clamp presence	Clamp closure	Omega closure	Stem length	Joint diameter	Dust shield presence	Stem length	Rubber ring presence	Etc.	HW mean
Camera view	Frontal	0	1	0	2	1	0	2	3	3	2	1	1	-	1,33
	Axial	2	3	2	2	1	3	2	1	2	3	3	3	-	2,25
Background	Black	1	1	3	2	2	1	2	1	1	3	3	1	-	1,75
	White	2	3	3	3	3	1	3	3	3	1	3	2	-	2,5
Light position	Back	2	2	2	2	1	2	2	3	2	3	2	1	-	2
	Front	1	1	2	2	2	1	1	1	2	1	2	1	-	1,42
Light type	Diffuse	3	3	3	1	3	2	2	2	1	3	1	1	-	2,08
	Direct	1	1	1	2	1	2	2	2	1	2	2	2	-	1,58
Camera type	Matrix	1	2	2	2	1	3	1	3	2	2	3	2	-	2
	Linear	1	3	1	1	3	1	1	1	1	2	3	3	-	1,75
Camera distance	Close	2	3	1	1	1	2	2	1	2	3	2	1	-	1,75
	Far	1	1	3	3	3	3	1	2	3	1	1	3	-	2,08
Defect location	Y mean [cm]	58	106	49	45	53	87	79	20	27	73	5	45	-	8 NC cams
	Y range [cm]	28	0	15	10	29	36	46	5	15	17	0	8	-	
Etc.	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

assembly cells and can detect over 60 defects in less than 30 s, processing 2.500 parts/day. This achievement is enabled by reconfigurable hardware components such as eight NC cameras, grippers, tailstocks, and adjustable lights, as well as parametrizable inspection algorithms tailored to specific product types. Such a complex system is a significant challenge and represents a research achievement for its complexity. However, its detailed description is outside the scope of this paper, as the main focus centers on the general proposed FVIS and RSS framework. Specifically, Section 3.2 delineates the conceptual design of the FVIS, while Section 3.3 presents the RSS development rooted on AFR techniques.

3.2 FVIS conceptual design

According to the first step outlined in Section 2.1, Table 4 provides valuable information into the preliminary conceptual design of the flexible hardware components within the

machine vision system. This design takes into account the identified defects labelled in Fig. 6(b). The full list of defects was generated using a hybrid (i.e., top-down + bottom-up) approach according to Section 2.1.

Each column (i.e., defect type) in Table 4, is associated with a score that reflects the compatibility of hardware options (e.g., camera view, background, lighting, camera type, camera distance) with the specific defect. The final column shows the average score obtained for each solution; the green cells highlight the optimal score for VIS hardware components. The final row, labeled “defect location”, shows the average positions and ranges spatial coordinates of defects (in this instance, only y is considered due to the longitudinal product extension). Such measurements are used to define how many cameras should be placed and where (if moving the camera via NC or not, or move the part, e.g., rotation), according to the Field of View (FoV) to ensure the coverage of all potential defects. For the current scope and objectives of the

preliminary conceptual design of the vision system, the centimeter level of accuracy is satisfactory, and any minor uncertainties can be effectively managed through practical approaches such as overlapping FoV. Occasionally, a suboptimal solution may be selected if it accommodates various defects.

The mean and variability of half shaft defects' positions are derived from analyzing the part number families in output from the identification stage. Once the defects' position within the FoV are defined, spatial clustering is carried out according to the camera specifications (e.g., FoV, focal length, resolution). A similar procedure is followed for the software components selection (e.g., inspection algorithm type, sequence, parameters).

Additional factors and technicalities, including the working environment and production constraints, are also considered into the final FVIS architecture design.

In order to achieve a fully flexible solution from a reconfiguration perspective, eight NC motorized axes cameras have been selected to cover the identified ROI clusters. By considering the longitudinal extension of each cluster, the maximum distance at which cameras are expected to be positioned, as well as the focal length of the optics, and FoV, it is possible to control which camera should be activated and its specific position. In addition to the previously highlighted aspects of the hardware design, mapping defects' positions aids in defining the maximum

working range of the adaptable (i.e., universal for all the families) gripper and tailstock system for handling the parts. This system is also NC controlled (Fig. 7(a)). A similar approach was adopted for NC-controlled and -adjustable lights.

In Fig. 6(b), defects are clustered based on their positions, with more detailed sub-groups illustrated in Fig. 8. In the example shown, the 16 different sub-groups of defects checked in the area inspected by NC camera#8 are presented. This approach of defect clustering is then extended to each variant within the different product families, as exemplified in Fig. 7(b). The associated ROIs for each defect types (about 60) for the current half shaft models (about 220) are then exhaustively checked as shown in Fig. 7(a). This process aids in determining the moving range of each camera and light.

The examined set includes all the identified half shaft models in order to be representative of the historical, current, and forthcoming production scenarios. In other words, for the system to be flexible enough, all the future models must be combinations of previously examined features, like in the present case (i.e., variant approach, Fig. 1).

As for the inspection software side, given the diversity across product families (Fig. 7(b)), different algorithms must be defined for inspection purposes. In the interest of brevity, Table 5 provides a condensed selection of examples, related algorithms, and some of the parameters identified in

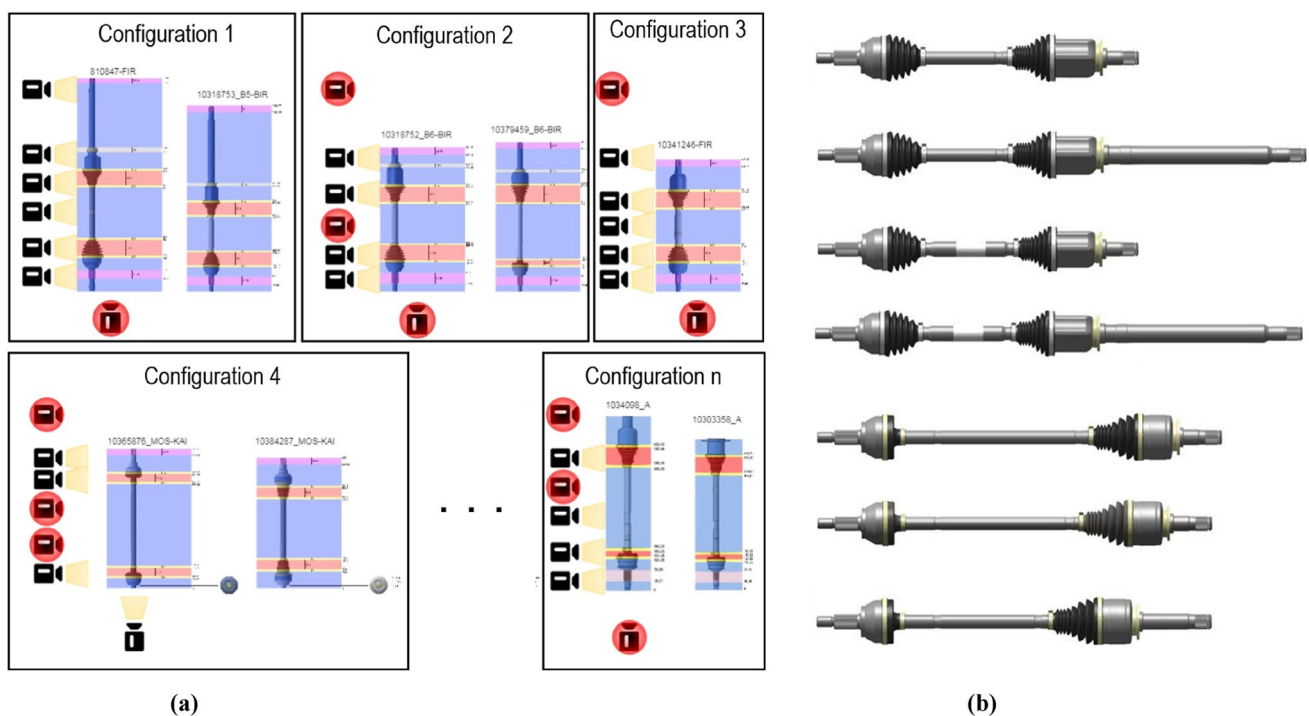


Fig. 7 **a** Different product family HW configuration (e.g., cameras activation and axial movement based on the parametrized ReCo file). Red cameras are deactivated. Camera, lights, grippers, and tailstocks can be repositioned using the NC axis. **b** Product families variability examples

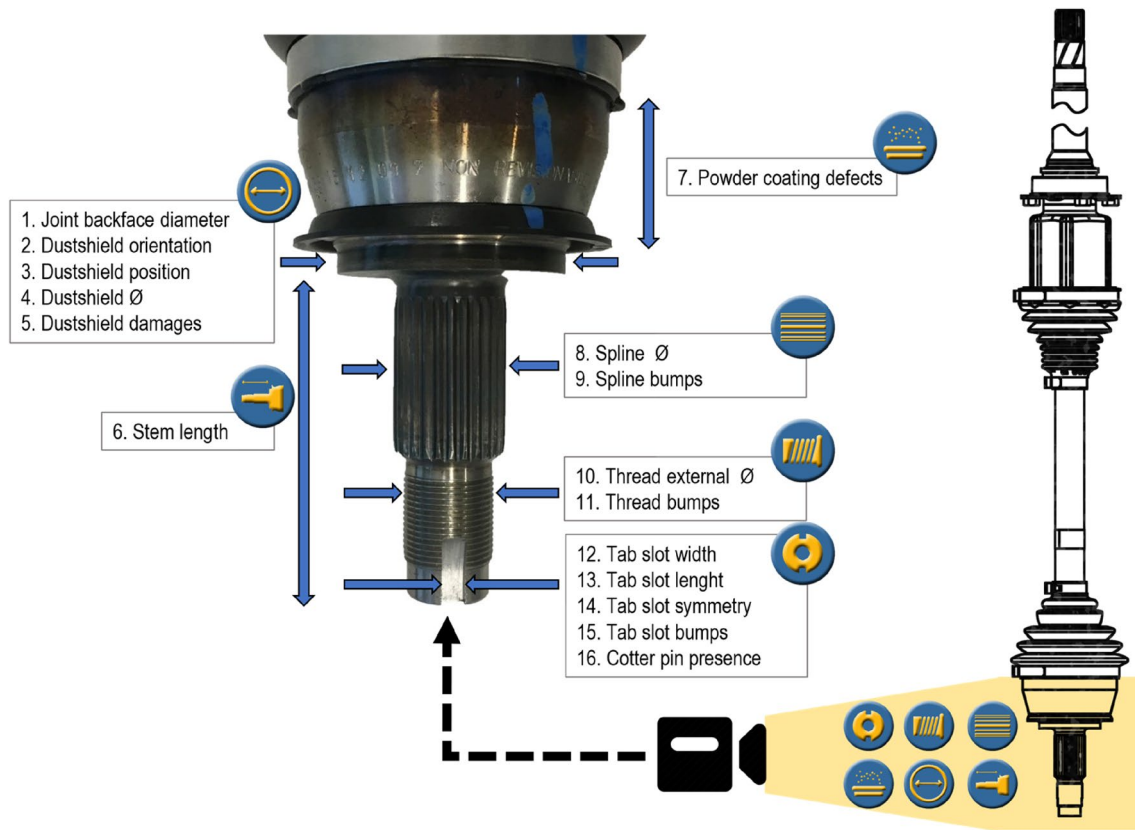


Fig. 8 Wheel joint side cluster consisting of 16 (out of 60) defect types categorized into 6 defect groups within the wheel joint side, highlighted by the yellow area. The identified defects are used to select the appropriate vision inspection algorithm and tool for quality check

the FVIS design stage. Given the requirements of intensive industrial application and the nature of defects, commercial software libraries have been selected and a specific commercial SW has been used (in alignment to the company's prior experience [46]). For each quality check, the chosen inspection algorithm, the corresponding software tool, key configuration parameters, output details, and the applied threshold to determine conformity are listed. Additionally, more advanced techniques for surface defect identification and classification can be adopted [47, 79, 80].

3.3 Implemented RSS for AFR

As anticipated and according to the approach proposed in Section 2.2, the RSS eliminates the necessity for an expert programmer and speeds up the ReCo file parametrization within an entirely virtual as well as offline environment for AFR, based on CAVI. The user's proficiency requirement pertains to the product rather than the FVIS. The sole task required for the user is to select the unified defect tags recognizable by the AFR software via the guided procedure implemented in the RSS GUI and

presented in Fig 9. In the procedure, a variant approach is adopted, enabling the user to select from a list of predefined defects organized according to tree logic product families (e.g., group A, B, C). The user is then able to assign tags for each identified defect outlined in Section 3.2 (Table 4).

The implemented RSS for the current case presents a CAD-based AFR approach. Once the product family half shafts have been selected and the CAD information retrieved from the PDM system, the AFR macro can discern the location and dimensions of the elements labeled by the user via an interactive procedure. According to the implemented AFR technique, a dedicated macro provides a digital interface to detect the type and the relevant dimensions (or presence) of the labeled 3D element. In Fig. 10, an example of the RSS GUI tagger developed for CAD-based AFR is depicted.

Once this step has been completed, the ReCo file in XML format is automatically generated and ready to reconfigure the FVIS hardware and parametrize the inspection software. Figure 11 provides an example of some lines for the generated ReCo file in XML format; encompassing various labels, including:

Table 5 Examples of quality checks carried out by the developed FVIS on a sub-set of 8 out of 60 currently implemented and operating. The inspection algorithm and its output are generalized to all the product families considered for the FVIS design. The parameters and thresholds are automatically extracted for each new reprogrammed

Half shaft component	Quality check	Inspection algorithm	Parameters	Output	Threshold
Spline	Spline \emptyset	Caliper	ROI (startX; startY; height; width); threshold	Distance between external spline sides	Distance range
	Spline no bumps	Blob	ROI (centerX; centerY; height; width); threshold; minarea	Blob area	Max area
Tab slot	Tab slots width	Blob	ROI (centerX; centerY; height; width); threshold	Width of slot's blob	Distance range
	Tab slots length	Blob	ROI (centerX; centerY; height; width); threshold	Height of slot's blob	Distance range
	Tab slots symmetry	FindLine	StartX; n° of calipers; distance between calipers; length of calipers; threshold	Difference between the angular coefficient of best fit lines of the points detected on slots' sides	Max value
	Tab slots no damages, no bumps	Blob	ROI (centerX; centerY; height; width); threshold; minarea	Blobs area	Max area
Clamp	Clamp closure conformity	Pattern matching	ROI (startX; startY; height; width); reference image	Percentage matching result	Min value
	Clamp size	Optical Characters Recognition (OCR)	ROI (startX; startY; height; width)	Text	Right text

part to be dynamically updated in the RSS AFR in Section 3.3. Inspection algorithms are first defined by the system developer and selected by the user for new part numbers through the wizard interface in Fig. 9

- **<job>**: This element represents the overall jobs and includes attributes such as “name”, “cameraName”, and “runCount”, which indicate the job’s name, the camera used, and the number of times the job is executed (36 times).
- **<tool>**: This element represents a tool utilized in the job. There are two tools shown in Fig. 9, namely “WSB_JointClamp” and “WSB_ShaftClamp”. Each tool has specific parameters enclosed within the **<params>** label.
- **<params>**: This section contains various parameters related to the tools. These parameters specify the regions of interest, detection points, and compensation points used in each tool.
- **<checkList>**: List of different checks performed using the tools. It includes elements such as “WSB JointClamp_diameterup”, “WSB JointClamp_diameterdown”, “WSB JointClamp_position”, and “WSB_JointSide_overflow”, each with respective attributes (e.g., nominal values, tolerances, and constants).

4 Results

The primary object of this work has been to develop a framework for a variant inspection vision system, called FVIS, and based on RSS, utilizing AFR applied to actual or rendered images, as well as CAD 3D models. The outcomes associated with this endeavor are presented in Section 4.1.

The second contribution of this work is the validation of the proposed framework through a challenging

case study conducted in the automotive industry (Section 4.2). The purpose of presenting this case study is to validate the effectiveness of the proposed framework in a real industrial environment. The selection of a particularly challenging case, which involves more than 60 different defects and 220 products spanning 12 product families, serves as high-quality benchmark for evaluating the applicability of the proposed framework. The experimental demonstration of the proposed approach showcases its capability to inspect a wide range of defects and products, making it well-suited for handling challenging scenarios involving multiple product families and diverse inspection tasks. The successful reprogramming capabilities for new products and the flexibility features of the hardware have been proven in this real case study. However, to fully establish the effectiveness and to expand the boundaries of validation, further testing across various experimental cases and real-world validation through multiple applications and years are necessary, as indicated in Fig. 12.

4.1 Framework results

The proposed framework enables the retrieval of information from the CAD-based AFR approach, generating Reconfiguration file (ReCo file) accordingly, and parametrically reconfiguring the flexible VIS. NC hardware components such as cameras, lighting systems, grippers, and tailstocks can be offline parametrized based on the specific ReCo file parameters. Inspection software algorithms can also be

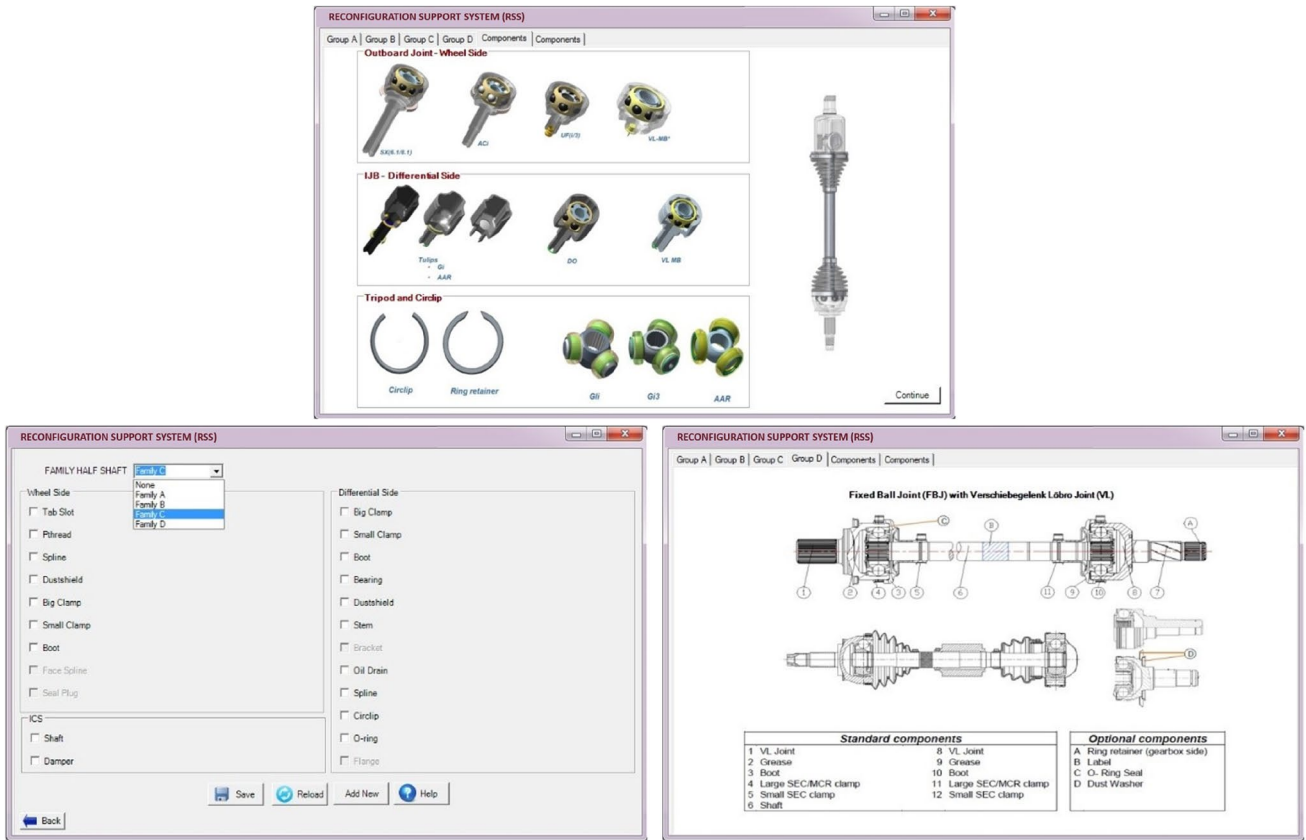


Fig. 9 Three screenshots of the developed user-friendly GUI part of the implemented Reconfiguration Support System (RSS) for the half shaft industrial case study

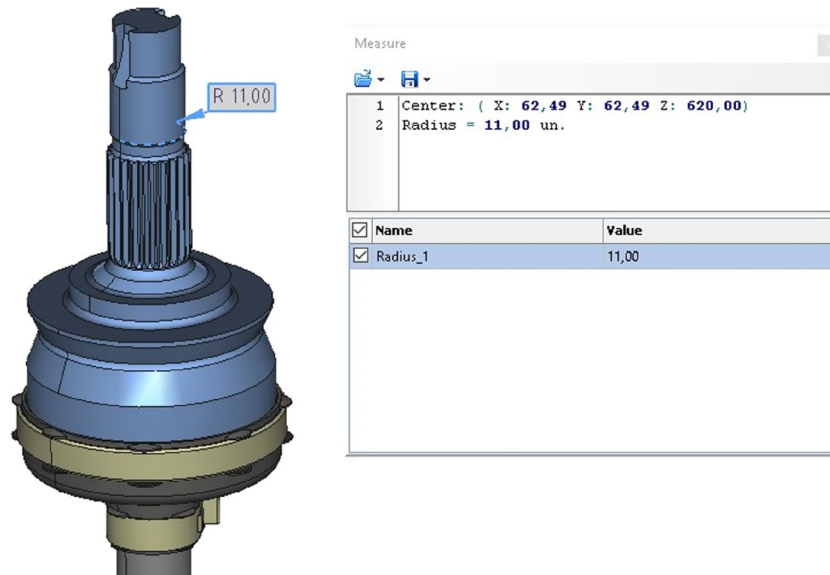


Fig. 10 Examples of assigning labels in Autodesk Inventor® via macros, which integrate the RSS with the CAD system and specific 3D model of the product under consideration to extract the specific

features and attributes. The blue highlighted area supports the user in identifying the part of the product under selection. The right pop-up shows the extracted features

```

...
<job name="Camera2" cameraName="Camera2" runCount="36">
  <tool name="WSB_JointClamp">
    <params>
      <ImageROI serializeAs="Rectangle" x="750" y="0" w="1600" h="3648"/>
      <ClampDetectionROI serializeAs="Rectangle" x="1670" y="1610" w="870" h="325"/>
      <JointCupSupROI serializeAs="SLSegment" x0="1510" y0="300" x1="1810" y1="300" s1="400" sd="90"/>
      <JointCupInfROI serializeAs="SLSegment" x0="1510" y0="3250" x1="1810" y1="3250" s1="400" sd="-90"/>
      <ClampSupEdge DetectionROI serializeAs="ARectangle" x="1930" y="95" w="400" h="285" r="90"/>
      <ClampInfEdgeDetectionROI serializeAs="ARectangle" x="1930" y="3195" w="400" h="285" r="90"/>
      <ClampDistortionCompensationLowerSup serializeAs="Point" x="2003" y="213"/>
      <ClampDistortionCompensationLowerInf serializeAs="Point" x="2000" y="3328"/>
      <ClampDistortionCompensationLowerCenter serializeAs="Point" x="1890" y="1720"/>
      <ClampDistortionCompensationUpperSup serializeAs="Point" x="2280" y="217"/>
      <ClampDistortionCompensationUpper Inf serializeAs="Point" x="2280" y="3328"/>
      <ClampDistortionCompensationUpperCenter serializeAs="Point" x="2234" y="1706"/>
      <Boot ClampDistanceROI serializeAs="Rectangle" x="1670" y="1610" w="450" h="310"/>
      <JoinECupEdgeIncSupROI serializeAs="SLSegment" x0="900" y0="450" x1="1200" y1="450" s1="500" sd="90"/>
    </params>
    <checkList>
      <WSB_JointClamp_diameterup nominalValue="110,85" infTol="0,35" supTol="0, 35"
k=0,0361689210361395" m="110,8"/>
      <WSB_JointClamp_diameterdown nominalValue="110,85" infTol="0, 35" supTol="0,35"
k="0,0361689210361395" m="110,8"/>
      <WSB_JointClamp position minValue="1" maxValue="3,5" k="0,0335322195704057"/>
      <WSB_JointSide_overflow underClampMaxValue="2" underClampk="0,0351851851852" overClampMaxValue="2"
overClampk="0,0351851851852" greasePresenceMaxValue="3" greasePresencek="0,0351851851852"/>
    </checkList>
  </tool>
<tool name="WSB_ShaftClamp">
  ...

```

Fig. 11 Example (extract) of a ReCo file in XML format for a new part to be inspected, generated by the developed RSS. It contains ROIs, conversion parameters, nominal values of dimensions, and relative tolerances. Each set of images taken by a single camera is processed by a job containing a series of tools specifically devel-

oped to carry out a certain number of checks on a specific area of the half shaft. Each tool operates calling the parameters. The additional required variables for the vision software (e.g., thresholds) are universal for all past and new parts

reparametrized using the same source of information stored in the ReCo file.

The proposed framework has several merits including the ability to improve knowledge management by embedding a RSS, which makes knowledge explicit and easy to access, manage, and transfer without spillover with staff turnover. Additionally, the system increases the amount of standard/compliance information retrieved during inspection, such as image databases. The system also aims to increase the efficiency of quality control by reducing programming time and increasing effectiveness offering the ability to carry out CAVI offline, which reduces the impact on shop floor productivity. Finally, the proposed framework also allows for hardware reconfiguration and software parametrization for new product families with similar vision features reducing the need for product-specific staff skills [81].

4.2 Case study results

A virtual prototype has been successfully implemented, and different hardware setup as well as inspection algorithms have been tested. The final FVIS architecture has been tested both virtually (Fig. 13(a)) and experimentally (Fig. 13(b)). Figure 13(a) depicts the 3D model of the final vertical layout configuration, showing four of the eight labelled NC cameras, NC ad hoc grippers and tailstocks

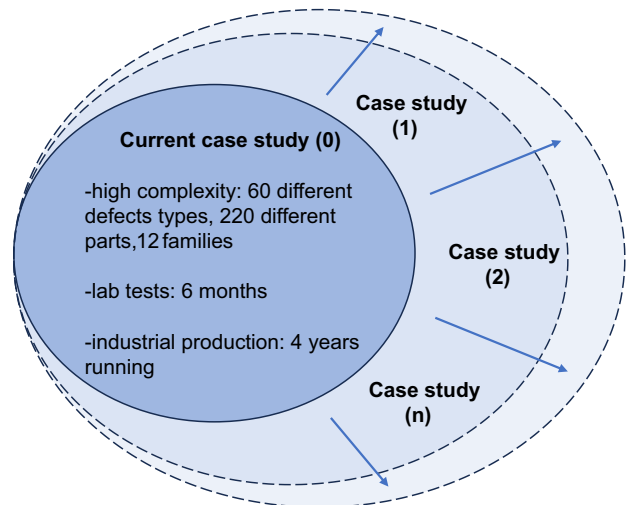


Fig. 12 Proposed framework validation and improvement. The dark blue area, representing current case study, highlights the initial step in validating the proposed framework. Its wide area indicates the adoption of a significant and complex case, encompassing both lab and industrial setups, which ensures a crucial first validation. The dotted lines and arrows depict the expansion of validation boundaries and possible improvement after implementing the framework in additional new case studies

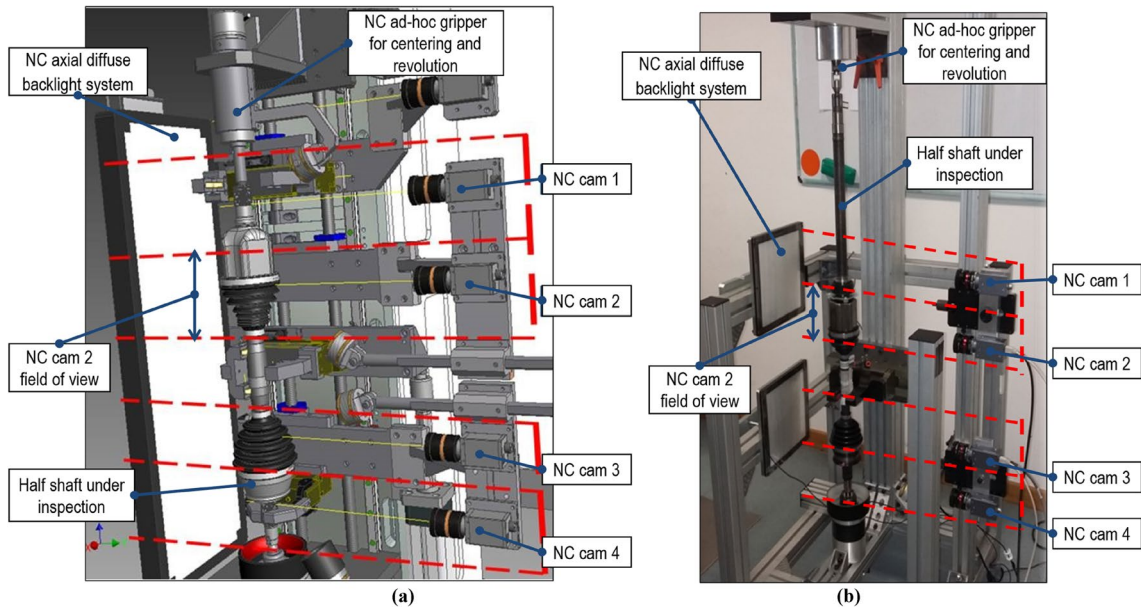


Fig. 13 **a** Autodesk Inventor® CAD virtual prototype design of the developed FVIS. Four NC camera out of 8. The half shaft is vertically placed and axially aligned by 360 rotational motorized special purpose grippers and tailstocks. **b** Lab's real prototype tests

for centering and revolution, NC axial diffuse backlight system, and a half shaft under inspection. This prototype validated the FVIS designed in Section 3.2.

In Fig. 13(b), a prototype of the FVIS represented in the 3D model has been tested in laboratory environment before achieving its final state on a production line.

The developed station inspects 100% of the assembled half shafts, following the main steps summarized as follows:

1. The half shaft is loaded into the inspection station by an anthropomorph robot.
2. Half shaft rotation starts via special NC grippers and tailstocks, all the lights are activated, and the door is closed.
3. The PLC controlling the inspection station sends a signal to the vision software, initiating the automatic inspection of the half shaft. Each camera captures 36 frames during a full rotation.
4. Captured images are transferred to the Personal Computer (PC) for processing via software.
5. Upon completion of the analysis, the results are sent back to the handling robot via the PLC deciding whether the half shaft is accepted and palletized for shipment, or if it necessitates manual reinspection for potential rework or disposal.
6. The GUI report is updated accordingly, displaying a red instead of a green box on the monitor of each camera if a defect has been detected (e.g., Figs. 14, 15)



Fig. 14 Cell operator's GUI on a touch screen on the inspection station at the end of the robotized assembly cell. Inspection of clamp differential side closure, positioning, lubricant for the part number

10302233. Green boxes (camera1, camera2, camera4DX, camera5) showcase conformity, while red (camera 3) highlights an identified defect



Fig. 15 A screenshot of the commercial vision software adopted. The block structure of the developed inspection software is shown on the left side. Right side provides an edge detected in red for the clamp under inspection

As for the quantitative results, the system’s efficiency is evaluated at the rate of True Positive (TPR) and False Positive (FPR) via the following formulas [82]:

$$TPR = \frac{TP}{TP + FN} \quad FPR = \frac{FP}{FP + TN}$$

where

TP (*True Positive*): the number of defective products that have been correctly classified as defective.

FN (*False Negative*): the number of defective products that have been incorrectly not classified as defective.

FP (*False Positive*): the number of non-defective products that have been incorrectly classified as defective (false alarm).

TN (*True Negative*): the number of non-defective products that have been correctly classified as non-defective.

TPR should tend to 100%, while FPR should tend to 0%. Note that a value over 0% for FPR can be chosen for conservative purposes, e.g., during the system development.

In the initial testing on the line, a sample of 60 products has been used, obtaining the following results: $TP = 15$, $FP = 4$, $TN = 41$, $FN = 0$. Therefore, $TPR = 100\%$, while $FPR = 9\%$. After fine tuning via a large experimental set from production and the optimization of threshold values for vision algorithms (e.g., binarization and pattern matching), performance reached 2% FPR and 100% TPR, also for new parts.

Table 6 Technical solutions and main aspects of the general implemented framework and the specific solution for the inspection station #R100GKN.

Technical solution	General proposed framework	Current implementation: half shaft inspection station #R100GKN
HW systems, sub-systems, and components	<ul style="list-style-type: none"> -Vertical layout -NC HW components -Programmable Logic Controller (PLC) -Automated loading system via robotic arm 	<ul style="list-style-type: none"> -8 NC cameras -NC lighting sources -PLC for motorized components control -Robot arm for part handling -NC special conformed grippers with Teflon wheels for product centering and revolution -Cognex VisionPro® v9.0 library
Vision SW algorithms controlled by PC	<ul style="list-style-type: none"> -Image analysis algorithms and management core -Multiple levels of GUI controller -Images compression and database storage for traceability -Communication with the PLC for (e.g., cycle start, processing result) -GUI for online report of defect types 	<ul style="list-style-type: none"> -Commercial libraries by Cognex VisionPro® ver. 9.0 -Two level GUIs controller (operator and superuser)
Reconfiguration Support System (RSS)	<ul style="list-style-type: none"> -HW setup module -SW reconfiguration module -GUI for HW and SW reconfiguration 	<ul style="list-style-type: none"> -RSS GUI developed in Python -CAD-based AFR developed in VB.net -ReCo file in .xml format

The system can assess more defects concurrently, but the occurrence of multiple defects in this type of production is considered a rare and exceptional event. Additionally, only certain defective parts can be reworked because most components cannot be disassembled without permanent damage (e.g., spline bumps cannot be recovered). As a result, after a defect is detected, the faulty part is segregated from the good parts and manually inspected to determine the appropriate actions for managing the non-conformity. Given the relative low frequency of defects, in most cases it is simply discarded. The re-programming time and cost for new parts inspection were also reduced by 40%, and low-level coding is not requested anymore. This transition has garnered unanimous operators' satisfaction (100% of the 10 interviewed operators dedicated to this task report an "easy-to-use" and "functional" system).

Not only the system implemented and presented is complex and challenging, it also has innovative flexibility features answering the main research questions: *how to virtualize the reconfiguration of a flexible vision inspection system and validate the proposed framework?*

Table 6 provides a summary overview of the general proposed framework and the implemented solution for the developed inspection station #R100GKN. Significantly, Table 6 highlights the adherence of the implemented system to the four key flexibility enablers presented in Section 1 (Table 1), namely, modularity, integrability, customization, and convertibility.

Modularity: the system's vertical layout enables flexible arrangement and easy integration of NC hardware components, including cameras and lighting systems. An automated loading system with a robotic arm enhances modularity, and NC special conformed grippers allow quick adjustments for different product sizes and shapes.

Integrability: the system focuses on seamless interaction between hardware and software components. A PLC serves as a centralized control unit, facilitating communication and coordination with motorized components. The system's communication with the PLC exemplifies its integrability.

Customization: the FVIS is highly customizable, catering to specific processing needs of diverse product families. An automated loading system with specialized grippers allows precise handling, adapting to various product configurations. Carefully selected NC hardware components can be easily customized for different inspection tasks.

Convertibility: the FVIS is remarkably convertible, allowing swift hardware and software changes for varying inspection demands. Vision SW algorithms controlled by a PC enable quick software modifications. The RSS with hardware setup and software reconfiguration modules facilitates rapid changes through a GUI developed in Python and a CAD-based AFR in VB.net.

As for main limitations, the current system also has challenges that may represent new research opportunities. These

include a lack of testing on different product types and a lack of a new AFR technique that relies on synthetic rendered images in digitally twinned environments. Additionally, the system has no specific focus on scalability and applying modularity, but these factors can be selected as driving factors in designing other FVIS.

5 Conclusion

The presented article introduces a comprehensive framework for the Flexible Vision Inspection System (FVIS), which is seamlessly supported by a Reconfiguration Support System (RSS). The core concept is based on the RSS generating Reconfiguration files (ReCo) through a user-friendly GUI and Automatic Feature Recognition (AFR) techniques. These ReCo files encompass both hardware parameters, such as sensors and light sources, and software parameters for the vision inspection software, such as position of ROIs and thresholds. This framework facilitates the creation of a unified parameter database that can be shared across multiple production lines and production plants. Extensive experimental testing has been conducted within a laboratory setup, focusing on an automotive half shaft inspection case, aimed at validating the framework's effectiveness. Subsequently, the framework has undergone further development and transformation into a fully operational system deployed at the factory, where it now serves as an integral component of the daily production processes successfully performing for 4 years. The complexity of the presented case study underscores the promising potential for the broader adoption of the proposed FVIS framework across a wide range of inspection settings. The potential for future research on a fully flexible (generative) approach has been addressed within the proposed flexible and reconfigurable framework [68].

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s00170-023-12175-6>.

Acknowledgements Project management: Michele Lanzetta, Gino Dini, Pierfrancesco Ceccarelli, University of Pisa Italy; Cesare Ottavi, Enrico Cannoni, Carlo Cavallini, Corrado Bonanno, Gianni Salvini GKN Driveline SpA Firenze Italy; Victor Munoz, GKN Driveline Mexico. Technical support from University of Pisa Italy: Stefano Chiodi (vision system), Giovanni Salvatore (station configuration), Yanet Vila (user-friendly semi-automated system prototype), Francesco Innocenti (CAD data extraction), Gabriele Moretti (experimental setup), and Flavio Antonelli (grippers design). Technical support from GKN Driveline SpA Firenze Italy: Valerio Raugi (station CAD design), Nicola Coscia (online tests), and Massimo Mazzi (station NC).

Author contribution All authors contributed to the study conception and design. Material preparation, data collection, and analysis were performed by Francesco Lupi, Michele Biancalana, Andrea Rossi, and Michele Lanzetta. The first draft of the manuscript was written by Michele Biancalana and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding Open access funding provided by Università di Pisa within the CRUI-CARE Agreement. This work was supported by financial resources. Author Michele Lanzetta has received research support from the GKN Driveline SpA Firenze Italy company.

Declarations

Competing interests Financial interests: Author Francesco Lupi, Michele Biancalana, and Andrea Rossi declare no competing interests. Author Michele Lanzetta has received research support from the GKN Driveline SpA Firenze Italy company.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Pansare R, Yadav G, Nagare MR, Jani S (2022) Mapping the competencies of reconfigurable manufacturing system with the requirements of industry 4.0, *J Remanuf.* 12:385–409. <https://doi.org/10.1007/S13243-022-00116-7/FIGURES/6>
- D'Avella S, Avizzano CA, Tripicchio P (2023) ROS-industrial based robotic cell for Industry 4.0: eye-in-hand stereo camera and visual servoing for flexible, fast, and accurate picking and hooking in the production line, *Robot Comput Integr Manuf.* 80:102453. <https://doi.org/10.1016/J.RCIM.2022.102453>
- Buerkle A, Eaton W, Al-yacoub A, Zimmer M, Kinnell P, Henshaw M, Coombes M, Chen W, Lohse N (2023) Towards industrial robots as a service (IRaaS): flexibility, usability, safety and business models, *Robot Comput Integr Manuf.* 81:102484. <https://doi.org/10.1016/J.RCIM.2022.102484>
- Lins RG, de Araujo PRM, Corazzim M (2020) In-process machine vision monitoring of tool wear for cyber-physical production systems, *Robot Comput Integr Manuf.* 61:101859. <https://doi.org/10.1016/J.RCIM.2019.101859>
- Yildiz E, Møller C, Bilberg A (2020) Virtual factory: digital twin based integrated factory simulations, *Procedia CIRP.* 93:216–221. <https://doi.org/10.1016/J.PROCIR.2020.04.043>
- Newman TS (1995) A survey of automated visual inspection, computer vision and image understanding. 61:231–262. <https://doi.org/10.1006/CVIU.1995.1017>
- Semeniuta O, Dransfeld S, Martinsen K, Falkman P (2018) Towards increased intelligence and automatic improvement in industrial vision systems, *Procedia CIRP.* 67:256–261. <https://doi.org/10.1016/J.PROCIR.2017.12.209>
- Ullah U, Bhatti FA, Maud AR, Asim MI, Khurshid K, Maqsood M (2021) IoT-enabled computer vision-based parts inspection system for SME 4.0, *Microprocess Microsyst.* 87:104354. <https://doi.org/10.1016/J.MICPRO.2021.104354>
- Alonso V, Dacal-Nieto A, Barreto L, Amaral A, Rivero E (2019) Industry 4.0 implications in machine vision metrology: an overview, *Procedia Manuf.* 41:359–366. <https://doi.org/10.1016/J.PROMFG.2019.09.020>
- Lupi F, Cimino MGCA, Berlec T, Galatolo FA, Corn M, Rožman N, Rossi A, Lanzetta M (2023) Blockchain-based shared additive manufacturing, *Comput Ind Eng.* 109:497. <https://doi.org/10.1016/J.CIE.2023.109497>
- Rossi A, Lanzetta M (2020) Integration of hybrid additive/subtractive manufacturing planning and scheduling by metaheuristics, *Comput Ind Eng.* 144. <https://doi.org/10.1016/J.CIE.2020.106428>
- Wang Z, Bai X, Zhang S, Billingham M, He W, Wang P, Lan W, Min H, Chen Y (2022) A comprehensive review of augmented reality-based instruction in manual assembly, training and repair, *Robot Comput Integr Manuf.* 78:102407. <https://doi.org/10.1016/J.RCIM.2022.102407>
- Smith ML, Smith LN, Hansen MF (2021) The quiet revolution in machine vision - a state-of-the-art survey paper, including historical review, perspectives, and future directions, *Comput Ind.* 130:103472. <https://doi.org/10.1016/J.COMPIND.2021.103472>
- Gastasini E, Capecci N, Lupi F, Gagliardi A, Saponara S, Lanzetta M (2021) An instrument for the characterization and calibration of optical sensors, *Sensors* 21:5141. <https://doi.org/10.3390/S21155141>
- Zheng C, An Y, Wang Z, Wu H, Qin X, Eynard B, Zhang Y (2022) Hybrid offline programming method for robotic welding systems, *Robot Comput Integr Manuf.* 73:102238. <https://doi.org/10.1016/J.RCIM.2021.102238>
- Zhou J, Camba JD (2021) Computer-aided process planning in immersive environments: a critical review, *Comput Ind.* 133:103547. <https://doi.org/10.1016/J.COMPIND.2021.103547>
- Katz R (2007) Design principles of reconfigurable machines, *Int J Adv Manuf Technol.* 34:430–439. <https://doi.org/10.1007/S00170-006-0615-2/METRICS>
- ElMaraghy H, Schuh G, Elmaraghy W, Piller F, Schönsleben P, Tseng M, Bernard A (2013) Product variety management, *CIRP Ann.* 62:629–652. <https://doi.org/10.1016/J.CIRP.2013.05.007>
- Hu SJ (2013) Evolving paradigms of manufacturing: from mass production to mass customization and personalization, *Procedia CIRP.* 7:3–8. <https://doi.org/10.1016/J.PROCIR.2013.05.002>
- Psarommatis F, May G, Dreyfus PA, Kiritsis D (2019) Zero defect manufacturing: state-of-the-art review, shortcomings and future directions in research. 58:1–17. <https://doi.org/10.1080/00207543.2019.1605228>
- Chiera M, Lupi F, Rossi A, Lanzetta M (2021) Lean maturity assessment in ETO scenario. *Appl Sci* 11:3833. <https://doi.org/10.3390/APPI1093833>
- Powell D, Magnanini MC, Colledani M, Myklebust O (2022) Advancing zero defect manufacturing: a state-of-the-art perspective and future research directions, *Comput Ind.* 136:103596. <https://doi.org/10.1016/J.COMPIND.2021.103596>
- Mital A, Govindaraju M, Subramani B (1998) A comparison between manual and hybrid methods in parts inspection, *Integr Manuf Syst.* 9:344–349. <https://doi.org/10.1108/09576069810238709/FULL/PDF>
- ISO 9000:2015(en), Quality management systems — fundamentals and vocabulary, (n.d.). <https://www.iso.org/obp/ui/#iso:std:iso:9000:ed-4:v1:en> (accessed November 1, 2022).
- Sahoo S, Lo CY (2022) Smart manufacturing powered by recent technological advancements: a review, *J Manuf Syst.* 64:236–250. <https://doi.org/10.1016/J.JMSY.2022.06.008>
- Lanzetta M, Rossi A, Puppato A (2016) Modelling activity times by hybrid synthetic method. *Production Planning & Control* 27(11):909–924
- Zhu L, Spachos P, Pensini E, Plataniotis KN (2021) Deep learning and machine vision for food processing: a survey, *Curr Res Food Sci.* 4:233–249. <https://doi.org/10.1016/J.CRFS.2021.03.009>

28. Lupi F, Rowley SJ, Chyba M, Lanzetta M (2021) Reconstruction of tubular structures from 2.5D point clouds: a mesophotic gorgonian coral case study, *ANZIAM J.* 63:C1–C14. <https://doi.org/10.21914/ANZIAMJ.V63.17151>
29. Martinez P, Al-Hussein M, Ahmad R (2019) A scientometric analysis and critical review of computer vision applications for construction, *Autom Constr.* 107:102947. <https://doi.org/10.1016/J.AUTCON.2019.102947>
30. Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, van der Laak JAWM, van Ginneken B, Sánchez CI (2017) A survey on deep learning in medical image analysis, *Med Image Anal.* 42:60–88. <https://doi.org/10.1016/J.MEDIA.2017.07.005>
31. Galata DL, Mészáros LA, Kállai-Szabó N, Szabó E, Pataki H, Marosi G, Nagy ZK (2021) Applications of machine vision in pharmaceutical technology: a review, *Eur J Pharm Sci.* 159:105717. <https://doi.org/10.1016/J.EJPS.2021.105717>
32. Javid M, Haleem A, Singh RP, Rab S, Suman R (2022) Exploring impact and features of machine vision for progressive industry 4.0 culture, *Sens Int.* 3:100132. <https://doi.org/10.1016/J.SINTL.2021.100132>
33. Lanzetta M, Culpepper ML (2010) Integrated visual nanometric three-dimensional positioning and inspection in the automated assembly of AFM probe arrays, *CIRP Ann.* 59:13–16. <https://doi.org/10.1016/J.CIRP.2010.03.047>
34. Cho CS, Chung BM, Park MJ (2005) Development of real-time vision-based fabric inspection system, *IEEE Trans Ind Electron.* 52:1073–1079. <https://doi.org/10.1109/TIE.2005.851648>
35. Mendoza F, Lu R (2015) Basics of image analysis, food engineering series:9–56. https://doi.org/10.1007/978-1-4939-2836-1_2/COVER
36. Lucas Y, Redarce T, Betemps M (1990) Integration of a machine vision system in a flexible workshop fitted out with CAD/CAM tools, *IFAC Proc Vol.* 23:619–623. [https://doi.org/10.1016/S1474-6670\(17\)52628-2](https://doi.org/10.1016/S1474-6670(17)52628-2)
37. Lanzetta M, Santochi M, Tantussi G (1999) Computer-aided visual inspection in assembly, *CIRP Ann.* 48:13–16. [https://doi.org/10.1016/S0007-8506\(07\)63121-7](https://doi.org/10.1016/S0007-8506(07)63121-7)
38. Lu Y, Liu C, Wang KIK, Huang H, Xu X (2020) Digital Twin-driven smart manufacturing: connotation, reference model, applications and research issues, *Robot Comput Integr Manuf.* 61:101837. <https://doi.org/10.1016/J.RCIM.2019.101837>
39. Erdős G, Paniti I, Tipary B (2020) Transformation of robotic workcells to digital twins, *CIRP Ann.* 69:149–152. <https://doi.org/10.1016/J.CIRP.2020.03.003>
40. Maiolino P, Woolley R, Branson D, Benardos P, Popov A, Ratchev S (2017) Flexible robot sealant dispensing cell using RGB-D sensor and off-line programming, *Robot Comput Integr Manuf.* 48:188–195. <https://doi.org/10.1016/J.RCIM.2017.04.004>
41. Zheng C, Xing J, Wang Z, Qin X, Eynard B, Li J, Bai J, Zhang Y (2022) Knowledge-based program generation approach for robotic manufacturing systems, *Robot Comput Integr Manuf.* 73:102242. <https://doi.org/10.1016/J.RCIM.2021.102242>
42. Beck J, Neb A, Barbu K (2021) Towards a CAD-based automated robot offline-programming approach for disassembly, *Procedia CIRP.* 104:1280–1285. <https://doi.org/10.1016/J.PROCIR.2021.11.215>
43. Lončarević Z, Gams A, Reberšek S, Nemeč B, Škrabar J, Skvarč J, Ude A (2021) Specifying and optimizing robotic motion for visual quality inspection, *Robot Comput Integr Manuf.* 72:102200. <https://doi.org/10.1016/J.RCIM.2021.102200>
44. Reichenstein T, Raffin T, Sand C, Franke J (2022) Implementation of machine vision based quality inspection in production: an approach for the accelerated execution of case studies, *Procedia CIRP.* 112:596–601. <https://doi.org/10.1016/J.PROCIR.2022.09.058>
45. West TD, Blackburn M (2017) Is digital thread/digital twin affordable? a systemic assessment of the cost of DoD's latest manhattan project, *Procedia Comput Sci.* 114:47–56. <https://doi.org/10.1016/J.PROCS.2017.09.003>
46. Software VisionPro | Cognex, (n.d.). <https://www.cognex.com/it-it/products/machine-vision/vision-software/visionpro-software> (accessed November 1, 2022).
47. Bouguettaya A, Mentouri Z, Zarzour H (2023) Deep ensemble transfer learning-based approach for classifying hot-rolled steel strips surface defects, *Int J Adv Manuf Technol.* 125:5313–5322. <https://doi.org/10.1007/S00170-023-10947-8/TABLES/3>
48. Braggins D, Connolly C (2009) Machine vision advances and applications, *Assembly Automation* 29:106–111. <https://doi.org/10.1108/01445150910945543/FULL/PDF>
49. Subrahmanyam S, Wozny M (1995) An overview of automatic feature recognition techniques for computer-aided process planning, *Comput Ind.* 26:1–21. [https://doi.org/10.1016/0166-3615\(95\)80003-4](https://doi.org/10.1016/0166-3615(95)80003-4)
50. Han JH, Requicha AAG (1997) Integration of feature based design and feature recognition, *Comput Aid Design.* 29:393–403. [https://doi.org/10.1016/S0010-4485\(96\)00079-6](https://doi.org/10.1016/S0010-4485(96)00079-6)
51. Liu SC, Gonzalez M, Chen JG (1996) Development of an automatic part feature extraction and classification system taking CAD data as input, *Comput Ind.* 29:137–150. [https://doi.org/10.1016/0166-3615\(95\)00081-X](https://doi.org/10.1016/0166-3615(95)00081-X)
52. Lucas Y, Redarce T, Jutard A (2002) CAD-based vision systems in pattern matching process. *Expert Syst.* 833–874. <https://doi.org/10.1016/B978-012443880-4/50067-3>
53. Shi Y, Zhang Y, Xia K, Harik R (2020) A critical review of feature recognition techniques, *Comput Aided Des Appl.* 17:861–899. <https://doi.org/10.14733/cadaps.2020.861-899>
54. Xu T, Li J, Chen Z (2022) Automatic machining feature recognition based on MBD and process semantics, *Comput Ind.* 142:103736. <https://doi.org/10.1016/J.COMPIND.2022.103736>
55. Babic B, Nestic N, Miljkovic Z (2008) A review of automated feature recognition with rule-based pattern recognition, *Comput Ind.* 59:321–337. <https://doi.org/10.1016/J.COMPIND.2007.09.001>
56. Zehtaban L, Elazhary O, Roller D (2016) A framework for similarity recognition of CAD models, *J Comput Des Eng.* 3:274–285. <https://doi.org/10.1016/J.JCDE.2016.04.002>
57. Liu E, Hsiao SW, Hsiao SW (2014) A decision support system for product family design, *Inf Sci.* 281:113–127. <https://doi.org/10.1016/J.INS.2014.04.039>
58. Zehtaban L, Roller D (2013) Beyond similarity comparison: intelligent data retrieval for CAD/CAM designs, *Comput Aided Des Appl.* 10:789–802. <https://doi.org/10.3722/CADAPS.2013.789-802>
59. Deja M, Siemiatkowski MS (2018) Machining process sequencing and machine assignment in generative feature-based CAPP for mill-turn parts, *J Manuf Syst.* 48:49–62. <https://doi.org/10.1016/J.JMSY.2018.06.001>
60. Wong FSY, Chuah KB, Venuvinod PK (2006) Automated inspection process planning: algorithmic inspection feature recognition, and inspection case representation for CBR, *Robot Comput Integr Manuf.* 22:56–68. <https://doi.org/10.1016/J.RCIM.2005.02.005>
61. Zhang H, Zhang S, Zhang Y, Liang J, Wang Z (2022) Machining feature recognition based on a novel multi-task deep learning network, *Robot Comput Integr Manuf.* 77:102369. <https://doi.org/10.1016/J.RCIM.2022.102369>
62. Hasan BA, Wikander J, Onori M (2016) Assembly design semantic recognition using SolidWorks-API, *Int J Mech Eng Robot Res.* 5:280–287. <https://doi.org/10.18178/ijmerr.5.4.280-287>
63. Recognizing features interactively - 2019 - SOLIDWORKS help, (n.d.). https://help.solidworks.com/2019/english/SolidWorks/fworks/t_Recognizing_Features_Interactively.htm?id=475bd8f3965b4f1a8925f854675405b7#Pg0 (accessed November 1, 2022).
64. Feature recognition | inventor | Autodesk app store, (n.d.). <https://apps.autodesk.com/INVNTOR/it/Detail/Index?id=9172877436>

- [288348979&appLang=en&os=Win64](#) (accessed November 1, 2022).
65. Pottmann H, Leopoldseder S, Hofer M, Steiner T, Wang W (2005) Industrial geometry: recent advances and applications in CAD, *Comput Aid Design*. 37:751–766. <https://doi.org/10.1016/J.CAD.2004.08.013>
 66. Yang J, Li N, Shi J, Tang W, Zhang G, Zhang F (2021) Foundation of 3D printing and CAD file formats used in the industry. *Multimater 3D Print Technol*:17–42. <https://doi.org/10.1016/B978-0-08-102991-6.00002-1>
 67. ISO - ISO 10303-21:2016 - Industrial automation systems and integration — product data representation and exchange — Part 21: implementation methods: clear text encoding of the exchange structure, (n.d.). <https://www.iso.org/standard/63141.html> (accessed November 2, 2022).
 68. Lupi F, Maffei A, Lanzetta M, (2023) CAD-based autonomous vision inspection systems, *Procedia Computer Science*, pp 1–6
 69. Křivánek J, Chevallier C, Koylazov V, Karlík O, Jensen HW, Ludwig T (2018) Realistic rendering in architecture and product visualization:1–5. <https://doi.org/10.1145/3214834.3214872>
 70. Hong S, Lee C, Chin S (2018) Physically based optical parameter database obtained from real materials for real-time material rendering, *J Vis Lang Comput*. 48:29–39. <https://doi.org/10.1016/J.JVLC.2018.06.004>
 71. Shitov DD, Gorbunova EV, Chertov AN, Korotaev VV (2014) Method of automatic color rendering settings for machine vision systems. 9217:188–195. <https://doi.org/10.1117/12.2061742>
 72. Schmedemann O, Baaß M, Schoepflin D, Schüppstuhl T (2022) Procedural synthetic training data generation for AI-based defect detection in industrial surface inspection, *Procedia CIRP*. 107:1101–1106. <https://doi.org/10.1016/J.PROCIR.2022.05.115>
 73. 3D rendering software & animation | KeyShot, (n.d.). <https://www.keyshot.com/> (accessed November 1, 2022).
 74. Chaos: 3D rendering & simulation software, featuring V-Ray, (n.d.). <https://www.chaos.com/> (accessed November 1, 2022).
 75. Cardozo WS, Weber HI (2018) A compact formulation for constant velocity joint kinematics, *Mech Mach Theory*. 121:1–14. <https://doi.org/10.1016/J.MECHMACHTHEORY.2017.10.009>
 76. Heisler H (2002) Transmission bearings and constant velocity joints, *Adv Vehicle Technol*:193–225. <https://doi.org/10.1016/B978-075065131-8/50007-5>
 77. Runji JM, Lin CY (2020) Markerless cooperative augmented reality-based smart manufacturing double-check system: case of safe PCBA inspection following automatic optical inspection, *Robot Comput Integr Manuf*. 64:101957. <https://doi.org/10.1016/J.RCIM.2020.101957>
 78. Jiang Y, Huang Z, Yang B, Yang W (2022) A review of robotic assembly strategies for the full operation procedure: planning, execution and evaluation, *Robot Comput Integr Manuf*. 78:102366. <https://doi.org/10.1016/J.RCIM.2022.102366>
 79. Mentouri Z, Moussaoui A, Boudjehem D, Doghmane H (2020) Steel strip surface defect identification using multiresolution binarized image features, *J Failure Anal Prevention*. 20:1917–1927. <https://doi.org/10.1007/S11668-020-01012-7/TABLES/5>
 80. Mentouri Z, Doghmane H, Moussaoui A, Bourouba H (2020) Improved cross pattern approach for steel surface defect recognition, *Int J Adv Manuf Technol*. 110:3091–3100. <https://doi.org/10.1007/S00170-020-06050-X/TABLES/7>
 81. Lupi F, Mabkhot MM, Boffa E, Ferreira P, Antonelli D, Maffei A, Lohse N, Lanzetta M (2023) Automatic definition of engineer archetypes: A text mining approach. *Computers in Industry* 152:103996. <https://doi.org/10.1016/j.compind.2023.103996>
 82. Satorres Martínez S, Ortega Vázquez C, Gámez García J, Gómez Ortega J (2017) Quality inspection of machined metal parts using an image fusion technique, *Measurement*. 111:374–383. <https://doi.org/10.1016/J.MEASUREMENT.2017.08.002>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.