FRAMOS

WHITE PAPER

How to Automate Manufacturing with Imaging

by Stefan Waizmann

www.framos.com

The Optimal Approach and Appropriate Technologies for Manufacturing Automation

CONTENTS

WHITE PAPER
How to Automate Manufacturing with Imaging

01 | SUMMARY

Automation is one of the most important steps in the optimization of production processes towards Industry 4.0. Machine vision is a suitable technology for all sectors and industries, as it can improve quality and increase the efficiency of production lines. Image processing offers several advantages for various applications: It opens new possibilities for efficient quality and production control as well as for warehouse and production logistics.
Generally, an image-processing system consists of a camera, lens, lights and cables, along with a processing unit and control electronics for synchronization and interaction with the production line. The successful development of such systems typically involves a multi-step approach comprised of five project phases: rough specification, feasibility study, detailed specification, system implementation and system integration.
This White Paper introduces the three predominant concepts of image-processing systems and will help production managers to choose the approach best suited to particular projects and conditions.

The software in "classic" image-processing systems must be individually programmed by experienced development teams to provide maximum flexibility in the selection of hardware and software libraries. Development costs tend to be very high and are often difficult to estimate accurately in advance. The classic approach is recommended if an application requires very specific algorithms or if it is expected to utilize all available computational power. The cost-effectiveness of this approach increases as the number of deployed systems grows.

A more recent approach combines traditional image-acquisition and processing hardware with image-analysis methods from the field of artificial intelligence—the so-called self-learning algorithms. This approach accelerates the development of test systems especially for natural materials and other objects, which have high variance that make analysisings with classical approaches difficult.

The most innovative approach, based on Smart Cameras, leads the way towards a new generation of image-processing technology. These complete solutions integrate the full set of image-processing components and a software package with an easy-to-use graphical programming interface. With this concept, less-experienced developers and production engineers are able to implement complex systems quickly at low risk. The effort required for development, testing and documentation is typically only a fraction of that under the "classical" approach.

Many factors and criteria influence the selection of an approach that is most appropriate for a specific challenge in manufacturing automation. A concluding table in Section 6 serves as a checklist and individual decision aid, and lists relevant questions and answers.

02 | THE ADDED VALUE OF MACHINE VISION IN MANUFACTURING AUTOMATION

"Inspection," "Control" and "Identification" are the most common applications to increase the quality, effectiveness and efficiency of production systems through image processing:

02.1 Inspection | Wheter it is from electronic components to baked goods, from engine blocks to rubber bands, businesses and consumers both expect a 100% quality. Therefore, 100% of the produced items should undergo quality checks. In addition, producers benefit from accurate statistics on the numbers and causes of rejects. The lesser the variance in the shape and color of produced goods, the easier it is to meet the above requirements with industrial image-processing.

How exactly would you inspect your products? A first possible approach is to measure objects optically by capturing images of them with a camera and measuring them with customized algorithms. The higher the resolution of the camera and the lower the optical distortion of the lens, the more accurate the results will

Common approaches for quality inspection:

1. Optical measurement
2. Surface inspection
3. Color inspection

be. In comparison to tactile coordinate measurement, the physical laws of optics limit the theoretically achievable accuracy at a lower level. However, optical measurement techniques offer a tremendous speed advantage, since systems with standard industrial cameras are already able to check dozens or even hundreds of individual objects per second. Machine vision is also the only reliable way to inspect printed products, such as bills and deposit tokens. A second approach to quality assurance is the inspection of natural or machined surfaces. Vision systems inspect the surfaces for mechanical errors or contaminations. A third inspection scenario reviews the color and texture of packages, textiles or materials to ensure a consistent quality and uniform appearance.

Application examples for “Inspection”:
- Geometric measurement of parts made by milling, cutting or bending
- Surface inspection of die cast components
- Print inspection on food packaging
- Glass inspection for medical tools
- Paint inspection in automotive production

02.2 Control | The second major area of application is production control. Here, the output of an image-analysis system is used to modify the parameters of an active system to obtain the desired results. In modern automobile production, for example, cameras review the amount and position of adhesives applied by a robot. In case of deviations from the target results, the robot adapts its settings triggered by the image processing program. The same is true for bottling facilities, where cameras monitor the fill level of bottles and the control system adjusts the quantities when target / actual deviations occur. In the sorting and recycling of garbage such as glass, an image-processing system controls the air vents that blow out differently colored glass parts into different sorting lines. Assembly or logistics robots use industrial cameras and computer systems with increased processing power and reliable algorithms to orient themselves appropriately so that they can select and place objects properly. Machine Vision makes many of these applications possible, while simultaneously reducing costs and improving quality. Machine Vision also contributes to production safety, an issue of growing importance. Image processing can detect the presence and the exact position of people or body parts and trigger warnings of pending danger or even stop the production line automatically.

Application examples for “Control”:
- Controlled adhesive application in car-body manufacturing
- Robot navigation to welding spots
- Sorting and recycling of garbage
- Robotic packaging of frozen fish

02.3 Identification | In logistics and in the manufacturing of complex products, many hundreds—sometimes even hundreds of thousands—of different objects must be processed. For intensive automation, each object must be identifiable; 1-dimensional barcodes and, more recently, 2-dimensional codes, e.g. QR- or Data Matrix codes, are used. The latter are able to encode more information in a smaller space. Laser scanners are widely used to read bar codes. Camera-based systems, however, are also able to read 2-dimensional codes and to collect additional information, such as an object’s size or position. On electronic parts, there is often not enough space to print readable number codes, along with machine-readable 1D or 2D codes. Therefore, these components are identified by optical character recognition of alphanumeric product codes.

Application examples for “Identification”:
- Identification of parcels in shipping logistics based on bar-codes
- Recognition of the type of a module in production logistics based on 2D codes
- Text recognition on electronic parts

03 | SUCCESS FACTORS: EXTERNAL CONSULTING VS. SYSTEM INTELLIGENCE AND INTERNAL COMPETENCIES

This chapter provides an overview of the most common and promising approach to follow, when there is a good general idea of which step of the production process should be automated.
The first step is to be absolutely clear about objectives. These in-

Rough requirement specification > Feasibility study > Detailled system specification > Implementation & test > Integration & test in environment

Project Plan for Manufacturing Automation with Image Processing

clude the expected commercial benefits, the opportunities to invest and expected hard technical constraints, such as available space, the maximum standstill period of production during final system integration and other concerns. The next important consideration is the availability of in-house skills and resources for project implementation, especially for the adaptation of the production plant's control logic.

> These questions should be answered a-priori:
> 1 What do you want to achieve? Quantify your goal!
> 2 What are your commercial constraints? Clarify the budget, deadlines and internal resources!
> 3 What are your technical constraints? Clarify the available space and the interfaces to the control system of the production line!

Once the main internal parameters are clarified, the procedure shown in Figure 1 has proven to keep investment risks low and achieve the desired results for technically and economically feasible projects.
When specifying rough requirements for the target system, it is best to involve a competent partner to consider important questions, such as:
a. What type of application are you dealing with (see Chapter 2)?
b. What are the characteristics of the objects to be analyzed (size of objects, speed of movement, size of elements or defects to be detected, etc.)
c. What are the requirements regarding system peripherals (space limitations, production control units and communication busses, etc.)?
d. What actions should be performed on a success and/or failure
e. Which development tasks will be implemented internally and externally?

To obtain unbiased advice, contact a dedicated consultant or technically skilled distributor with a broad portfolio of imaging components for an evaluation of the available technologies and products suited to a specific application.
Before you start to develop a complete system, all parties should understand the image-processing task as deeply as possible. This can be ensured by a feasibility study where appropriate algorithmic approaches are selected and tested. The study will help define the achievable goal, guide the final development and integration effort, and estimate complete system costs. Depending on the system complexity, it may be advisable to engage a suitable company for development and integration. Select a System Integrator that does not just build on its expertise in image processing hardware and software, but also actualizes the interoperation of the specific control system with the production line. Ask for recommendations from your consulting company or from a trusted distributor. If the automation task can be solved with a smart-camera-based solution, the project can be implemented with in-house resources and system expertise.

> Criteria for identifying the best system integrator:
> - Reference projects with similar imaging requirements
> - Broad experience with algorithm development
> - Existing experience with the production control system
> - Local proximity
> - Implementation based on fixed price offer

A feasibility study at an early stage is of great help for these considerations. The study creates a detailed system specification, which largely serves as a statement of work for the subsequent development and integration projects. Make sure that system tests are also specified as acceptance criteria during the implementation phase.

> „One of the most complex parts of an image-processing system is its algorithm for image analysis."

04 | REQUIRED HARDWARE AND ITS SELECTION

Hardware components of an imaging system:

- Camera + lens
- Illumination
- Processing unit
- Cabling

or

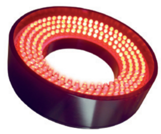
- Smart camera
- Cabling to the production-control-units

Even if a huge variety of applications for industrial image processing exists, most systems consist of the same basic building blocks.

Lens I The lens focuses light from the objects onto the image sensor of the camera. Focal length, field of view, spatial resolution and size of the projected image circle are the major quantitative parameters. Qualitatively, low distortion, smooth transmission and low variance in optical properties over several production lots are critical to ensure a consistent performance across multiple assembly lines and/or test stations.

Industrial Camera I The camera is an important and technically complex system component. The image sensor, together with the camera electronics and communication interface, determine, among other parameters, resolution, frame rate and minimum exposure time of a camera. The higher the resolution, the smaller the size of the elements that can be recognized in captured images. The higher the frame rate, the larger the number of objects that can be recorded and transmitted to the system for analysis in a given time. The minimum exposure time determines how fast the objects may move without creating blurred images. Keep in mind that the minimum exposure time must be less than the reciprocal of the frame rate. Otherwise, the frame rate will be limited by the reciprocal of the minimum exposure time. Should this be the case, one will either need to slow down the inspection time or increase the illumination so as to reduce the required minimum exposure time and thus increase the frame rate of the camera.

LED Illumination I When the image-processing system must analyze fast-moving objects or function well despite variations in ambient light, powerful artificial illumination is needed. Almost all applications use LEDs, due to their durability and reliability. Control electronics, such as steppers or photoelectric cells, activate the camera and LED strobe. Flash-mode operation lowers power consumption and reduces the thermal aging of LEDs, while generating significantly greater light intensity. The use of colored LEDs and a matching camera filter further reduces the influence of ambient light while providing additional contrast to images that make the captured images easier to process. In addition, and especially for quality-assurance applications, the strategic positioning of light sources improves the identification of specific defects based on their shadows and/or reflections.



Cabling I Cables also play an important role. Reliable power and error-free transmission of control pulses and video data are essential for a system with the requirement for 100% operational runtime. In particular, when cameras are mounted to moving devices such as robotic arms, cables must withstand continuous bending and torsional forces along with being properly shielded so as to not be affected by any electromagnetic noise that may be present.

Image Processing PC I To obtain the results of the image analysis in real time and with a short delay, a powerful computing unit is necessary. The computational load increases at least linearly with camera resolution; at a minimum, color processing triples the computational effort compared to grayscale imaging. In addition, the computer must support the data interface of the camera and should provide additional interfaces to the control units of the production line. In environments with dust or moisture, make sure that the image-processing computer does not contain continuously moving parts such as cooling fans. It is essential that the design engineer precisely match all system components to meet the requirements. Hardware components un-

suitable to the application or insufficiently matched with each other account for 50% of the problems of image-processing systems. Matching requires considerable expertise and experience. Secure advice from an experienced hardware distributor or other experts. In the case of smart cameras, the matching of most system components is obsolete. Some smart-camera manufacturers offer perfectly matched image processors that support the implementation of multi-camera systems (see next point and Chapter 5).

Smart Camera I To significantly save time and reduce the risk of incompatibility, smart cameras combine lens, camera, lighting, lighting control and processing unit in a single compact device. This ensures that essential hardware components match each other perfectly, although the high integration density limits available computational power that one would get with a fully integrated system. The fact that hardware elements are well harmonized, however, constitutes a significant advantage in the planning and final implementation of an automation system. Thanks to new generations of strong processors from the IT sector and the availability of a broad range of resolutions, frame rates, lenses and lighting systems, these products are the perfect fit for many applications. Most smart cameras also incorporate dedicated software frameworks with graphical programming interfaces, which minimize the risk, time and cost of system development. Chapter 6 provides further details.

05 I System Architectures for Automation Applications

Recent advances in software and algorithms for image analysis opens up new opportunities to develop more powerful and more efficient vision systems in shorter periods of time. This chapter introduces, compares and provides examples of the three most common industrial system architectures. Chapter 6 lists a series of conclusions and this white paper ends with a list of relevant questions and answers presented in table format.

05.1 | INDUSTRIAL CAMERAS AND TRADITIONAL PROGRAMMING

Criteria	Property
Hardware	Classical industrial video camera Separate components (lens, illumination, processing unit) to be matched
Development approach	Programming with higher languages like C++ or C#
Algorithms for image processing	Flexibly selectable from free and paid third-party libraries or in-house development
Required competence	Experienced programmers of software architectures and image processing
Economic applications	When these aspects are required: Highest possible processing speed Universal flexibility of the software Special algorithms High number of deployed systems

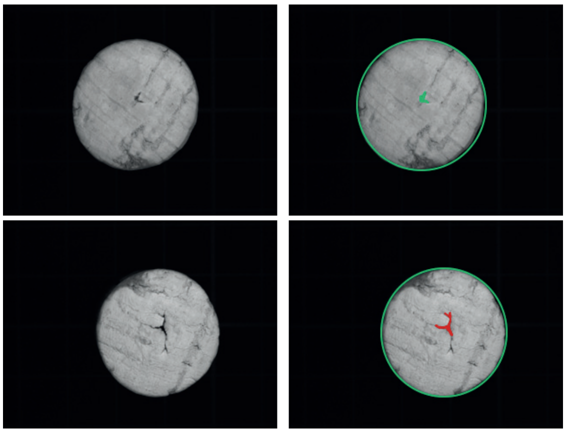
This classical and still widely used system consists of multiple hardware components and often low-cost or free software in high-level languages such as C++ or C#. Development teams take full responsibility for hardware selection and tuning, and the implementation of all hardware and software interfaces between libraries. This provides the greatest flexibility in system design and may minimize hardware and software licensing costs, but typically requires considerably more development effort, which is hard to estimate. At least some of the development team should have extensive experience in industrial image processing and implementation of efficient algorithms, since large, multidimensional datasets must be processed in a very short time. Thus, this approach is recommended only when very specific algorithms are required or when large numbers of systems must be implemented over the long run. If it is clear from the beginning that an image-processing system will be produced and marketed in large quantities, it may be worthwhile to assign development to internal resources, thus saving hardware and software licensing costs.

Advantages	Disadvantages
Greater flexibility in design of final solution	Software connection to peripherals must be programmed
Available computational performance can be fully exploited	Image-recording components must be matched manually
Opportunity to adopt manifold algorithm libraries	Development requires experienced programmers
Allow customized solutions	The development effort is often high, hard to estimate and time-consuming
Use/build only what is needed	Leads to long-term dependency on the system integrator, since the developed software is in most cases not comprehensible

05.2 | INDUSTRIAL CAMERAS WITH ARTIFICIAL INTELLIGENCE

Criteria	Property
Hardware	Classical industrial video camera Separate components (lens, illumination, processing unit) to be matched
Development approach	Programming with high-level languages like C++ or C# Tools based on Artificial Intelligence just need to be trained and linked into the software application
Algorithms for image processing	The system learns self-controlled, how to analyze and classify images of parts, e.g. as “OK” or “not OK”
Required competence	Programmers experienced in software architectures and image processing
Economic applications	Inspection of natural products like fruits, vegetables and meat Inspection of products made of natural materials, like cork and pastries Inspection of highly complex objects, like faces of analogue quality watches

This type of system uses classic hardware, i.e. an industrial camera, lens and lighting in combination with a suitable computer. The software is implemented with advanced programming languages and usually purchased software tools, which perform the training and run-



Original images (left) and results of the analysis (right) with positive decision (above) und negative decision (below)

time execution of the image analysis utilizing self-learning algorithms and other methods from the area of artificial intelligence.

The software needed to connect peripheral devices for control and image acquisition also requires greater programming skills, and thus significant development expertise. However, current self-learning algorithms significantly facilitate image analysis. These include convenient software tools and sample images to help during the first phase of training. The learning algorithm creates mathematical models from images of good parts and bad parts and defines the decision rules independently. Thus, the implementation of the algorithm is often faster than with purely rule-based methods. This can be particularly true with images of natural objects with high rates of variance, such as fruits, vegetables, baked goods and other products made from natural materials. This approach can also be helpful in production processes requiring close examination of many details (i.e. those with few “bad” parts). Some software tools designed for this purpose ensure the reliable detection of bad parts, even in situations where training focuses exclusively on good parts. Examples include the manufacture of high-quality watches and the qualitative monitoring of pieces with natural variations. One weakness of this approach is the difficulty of manually setting parameters for detection criteria.

Due to their high added value for some applications and the significant research and development efforts required, software tools for artificial intelligence methods tend to be expensive. In addition, these require larger amounts of memory and processing power than smart cameras can provide. Therefore, the user or system developer must take care of the imaging hardware, such as camera, lens, processing unit, etc. on the basis of its own competence of with external advice. Some programming experience is also required during overall system implementation. However, in a development project, the implementation of the analysis algorithm often carries the greatest risk, which can be mitigated by using artificial intelligence to test and evaluate even during the early phase of the feasibility study.

An example of how image processing can be successfully used with artificial intelligence is cork scanning, as shown in the above figure. Important quality criteria are shape, radius and smoothness of surface. These criteria identify good training algorithms independently by analysis of good and bad parts. The recognition algorithm applies these criteria to newly produced corks by verifying whether outer edges comply with target values (green circle). In addition, the algorithm detects particularly dark surface areas and evaluates, among other things, the number of contiguous dark pixels. The top cap has a small dark area in the center, although not large enough to be rejected. The lower cap, however, has a large notch in the surface and therefore, the system rejects it.

Advantages	**Disadvantages**
Reliable inspection of complex or highly variable objects	The software interfaces to the peripherals must be implemented
Low risk in time and costs at the final implementation	The components for the image recording must be matched to each other manually
Accelerates the development of the algorithms for image analysis extremely	Leads to long-term dependency on the system integrator, since the developed software is in most cases not comprehensible and internal
Beneficial at the examination of natural and heavily structured objects	buildup of competencies is required Examination of strict geometric measures and tolerances is hard to actualize
	The development requires experienced programmers

05.3 | SMART CAMERAS AND GRAPHICAL PROGRAMMING

Criteria	Property
Hardware	Smart Camera integrates camera, lens, processing unit and often also the illumination
Development approach	Algorithms are assembled within a Graphical User Interface by ready-to-use processing blocks, which are just configured by their parameters
Algorithms for image processing	Broad selection of methods for image pre-processing and analysis (depends on the chosen product)
Required competence	Engineers with up to 3 days of training, depending on their previous knowledge about image processing
Economic applications	Geometric measurement of parts produced by bending, cutting or milling Verification of print results on packages Presence / absence detection of certain elements Identification of parts via text, bar codes or 2D codes

With this system architecture the matching of camera, optics and processing unit becomes obsolete thanks to their integration into a Smart Camera. In addition, an LED illumination, which is sufficient for diverse applications, is often built into the camera already. Many smart cameras on the market include software-development platforms based on the so-called Graphical Programming, which offers technical and economic advantages over traditional programming in high-level languages such as C ++ or C #:

Software I The software interface to a wide range of cameras and to control hardware for production lines is already implemented and ready to use.

Hardware I Clearly structured software interfaces help link additional components, if needed.

Development costs I The greatest time and cost advantage of these solutions is that image-processing algorithms need not be written, line by line, in programming languages. Rather, they are "programmed graphically," i.e., the developer selects basic image-processing functions from a menu and puts them into a flowchart. Each function block performs a documented operation on the image and can be configured with specific parameters. Within a few hours, highly complex algorithms are assembled by the concatenation of function blocks. Hence, the underlying software architecture automatically optimizes the algorithm for execution on conventional multi-core CPUs. The drastic reduction in development time is extremely helpful in feasibility studies, as a first version of the analysis algorithm can be implemented and evaluated with comparatively little effort.

Easy access I Some platforms also offer a comfortable design environment for the graphical user interface (GUI). These display current system status, analysis results and statistics, and enable the user to access key system parameters. Implementing the GUI itself can be extremely expensive when using modern libraries in higher programming and requires some programming experience.

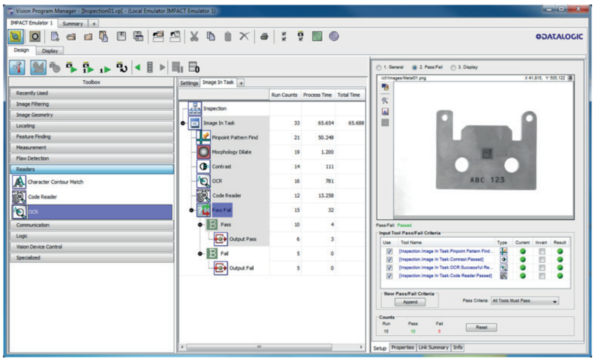
Documentation I Graphical programming provides much simpler traceability of algorithms used, which supports subsequent development. Documentation is often neglected in classic software development.

WHITE PAPER
How to Automate Manufacturing with Imaging

Software platforms with Graphical Programming should feature the following:

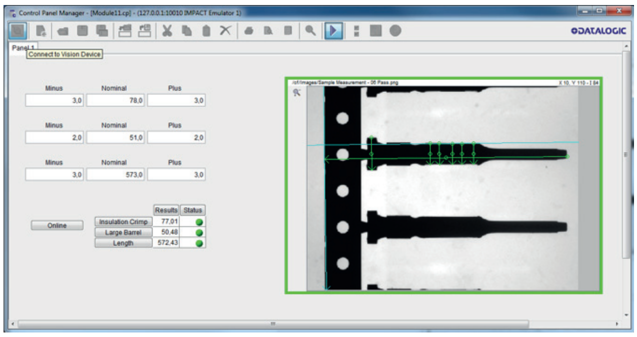
- Interfaces to a broad portfolio of standard hardware
- Flexible and documented options to interface with further hardware
- Broad variety of image processing functions with comprehensive parametrization
- Comfortable graphical development environment for algorithm development
- Module for the statistical evaluation of the image analysis
- Comfortable graphical development environment for Graphical User Interfaces for the display of statistics, parametrization and the results of each analysis
- Opportunity to execute the software on Smart Cameras and image processing computers

This approach offers the most rapid implementation, often within a few hours, entails relatively low risk and requires the least amount of experience with machine vision.
Disadvantages include a limited selection of cameras with appropriate image-resolution rates and acquisition speeds. However, with the correct optics and lighting solutions and by using specifically matched image processors a much broader range of standard cameras can be applied, even in multi-camera setups. The software libraries for graphical programming usually lag behind those for high-level languages in terms of functionality. While much development time may be saved, the costs of hardware and software licenses are higher than for traditional industrial cameras, making this approach especially efficient only when small numbers of systems are to be implemented.



Example of a software platform for Graphical Programming with implemented algorithm for geometric measurement, text and 2D code recognition

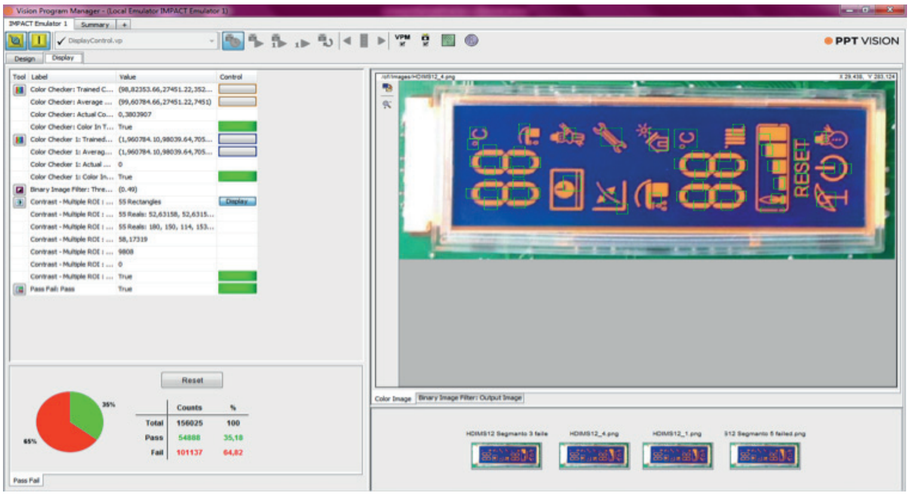
Thus, the approach of smart cameras and graphical programming is primarily for rule-based detection methods, as well as for measurement, code reading, or text recognition tasks. The approach is helpful for the identification of packaging, components or whole modules. Graphical programming can also be used to create test and measurement algorithms to ensure that manufactured components with precise specifications, such as semiconductor chips, circuit boards and displays, comply with quality requirements. Chapter 6 lists key decision criteria to determine which image-processing approach is best for specific production challenges.
For the inspection of the display, all elements are controlled to light up. The camera then takes a picture of the display under defined conditions and hands it over internally to the algorithm for analysis. Here a total of 55 "regions of interest" are defined. Figure 9 illustrates them by green rectangles. A set of rules for each rectangle determines whether the piece passes the quality test. The algorithm checks each addressed display indicator regarding its availability, contrast to the background and color.



Example for the user interface of a system to inspect metal anchor bolts, which has been created and linked to the image processing algorithm by Graphical Programming

The figure below shows the user interface of a system for the verification of LCD displays as a classic example of the benefits of a Smart Camera based approach.

Inspection of LCD displays for presence / absence, contrast and color of display elements



Advantages	Disadvantages
No programming know-how required	Higher costs for hardware and licenses
Implementation is easy to understand and requires just minimal additional documentation	Limited function set; extension requires programming experience again
Camera, lens and processing unit are all integrated and perfectly matched	Available set of cameras and image sensors is limited
Low risk in time and costs thanks to assessable development efforts	Lighting and optics may not match the requirements of the parts under inspection to adequately provide the contrast in the captured images making their processing longer and more difficult
Very low implementation effort for algorithms and graphical user interfaces	

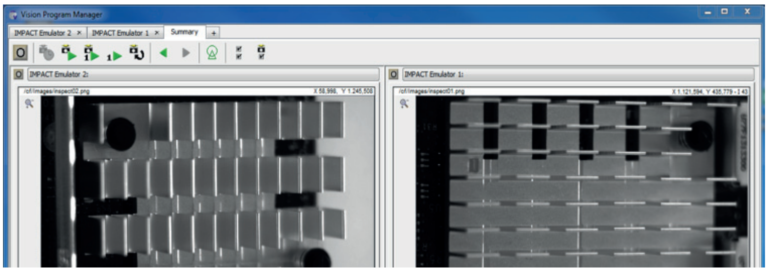
05.4 | PARTICULAR ADVANTAGE: SMART CAMERAS IN MULTI-CAMERA-SETUPS

Some providers of Smart Camera systems additionally offer adequate image processing PCs. Thus, even after the implementation of the algorithm the system designer has the freedom to decide about the processing hardware with little extra effort.
In industrial automation, there are often applications where the use of more than one camera is very helpful or even necessary:

Case 01 | The object to be examined is too large for a camera. For example, during the inspection of solar panels, large screens, or wide strips of fabrics, there is often no economically feasible camera which can provide appropriate resolution levels and recording speeds. The only effective solution often involves multiple cameras arranged appropriately.

Case 02 | Simultaneous inspection of an object from multiple angles. This can serve the purpose of 3-dimensional measurement of an object or to analyze its surface and contours from several sides at a time.

Case 03 | Recording at different resolutions. In camera guided robots, one camera with a wide field of view captures coarsely the spatial orientation and ensures the safety of operation, while often an additional camera delivers accurate images from the point of operation of the robot's tool.



Inspection of cooling fins by simultaneous capture from two angels

Case 04 | Central processing of camera shots from various points in the production process. In a production system, cameras may be used for automation in several places, while a central processing unit performs the image analysis.

The implementation of an image-processing system with more than one camera is a particular challenge in several aspects. To accommodate the need to analyze large amounts of data simultaneously, the software must make the most efficient use of available computing power, for instance. In the above-mentioned Cases 1, 2 and 3, camera images must be in perfect sync. While this challenge may sound trivial, it can be difficult for programmers using higher programming languages.

However, there are software frameworks with graphical programming that avoid the pitfalls of multi-camera systems and support system and algorithm development with little effort, cost and risk. Given these advantages, more and more experienced programmers are happy to give up the opportunity of the deepest control and optimization of a self-developed multi-camera software application, e.g. in C ++. They rather choose the benefits of Graphical Programming for dedicated image processing PCs with Smart Cameras.

SELECTION CRITERIA FOR SYSTEM DECISION

	Industrial Camera + Traditional Programming	Industrial Camera + Artificial Intelligence	Smart Camera + Graphical Programming
System specification			
Components for image capturing	Industrial camera, lens, illumination	Industrial camera, lens, illumination	Smart Camera (integrated lens and illumination)
Processing hardware	Industrial PC	Industrial PC	Smart Camera
Software	Advanced programming languages, primarily with free libraries	Advanced programming languages with tools for machine learning and recognition	Libraries with Graphical Programming
Primary fields of applications			
Number of systems to be deployed	Medium to high (> 20), depending on the higher development effort compared to other approaches	Medium, 20 to 100	Low, 1 to 20
Applications	Serial products, which use image processing, like pick-and-place machines, deposit stations, banknote inspection devices, etc.	Recognition of defects and abnormalities in production, like inspection of highly variable natural products such as food	Rule-based inspection, like measurement, presence / absence recognition, code and character recognition

SELECTION CRITERIA FOR SYSTEM DECISION

	Industrial Camera + Traditional Programming	**Industrial Camera + Artificial Intelligence**	**Smart Camera + Graphical Programming**
Development			
Time consumption / costs	Very high and mostly hard to estimate	Medium. Analysis algorithms widely solved a-priori; software interfaces must be implemented	Low. various function blocks and interfaces already available
Required competencies	Long-lasting experience in software development, algorithms and image processing	Experience in software architectures; basic competencies in machine learning plus few training days	Basic competencies in programming and image processing plus few training days
Risk	Very high at complex software architectures and algorithms	Low. effort can be estimated after the feasibility study already	Low. sometimes just few hours; effort can be estimated adequately after feasibility study
Software			
Variety of algorithms	Unlimited, thanks to own developments and various freely available libraries	Limited to the chosen software tool	Limited to the chosen software tool; can be extended by programming in advanced languages
Interfacing with hardware components	Unlimited, with advanced and low level programming languages	Limited to the chosen software tool; can be extended by programming in advanced languages	Limited to the chosen software tool; can be extended by programming in advanced languages
License costs	Often low, depending on the libraries used	High	Often included in the hardware costs
Hardware			
Costs	Can be widely reduced to the actual technical requirements	Can be widely reduced to the actual technical requirements	Higher than with classical components in favor lower development costs
Requirements on computing power	Depending on the applied algorithms	Rather high due to complex feature extraction and pattern recognition	Rather low at simple image analysis and rule based recognition
Overall costs			
Overall costs	Rather high due to complex development with expensive programmers	Driven by development efforts of programmers; Reduced by quick implementation of the image analysis	Higher hardware costs are over-compensated by very low development effort

WHITE PAPER

How to Automate Manufacturing with Imaging

Stefan Waizmann
Head of Product Management

06 | CONCLUSION

Decisions about approach thus hinge primarily on a consideration of application requirements, the availability of developers experienced in image-processing systems, as well as the desired balance of development cost and production cost.
The difficult answer to the question „Which system should I use for my automation?“ is thus based on the following key aspects:

1 What is to be measured or tested?
2 Which criteria must be checked?
3 Is the appropriate image-processing expertise available?
4 Do I have appropriate programmers?
5 How many identical systems are scheduled to go into operation?
6 How quickly must the system be productive?
7 What is the return on my investment for each method (development costs vs. savings in better quality and/or lower rejection rates, man-power,...)?

For the identification of objects using optical-code reader and bar code, or 2-D code and the verification of geometric or color specifications, then smart cameras and graphical programming are usually—and by a significant margin—the cheapest, fastest and most reliable solution. The same applies to the study of simple surfaces for defects.
When natural products, such as fruits, vegetables, meat, baked goods and the like, have to be classified according to their shape, size, color, or defects, then highly-developed software tools with self-learning algorithms from the field of artificial intelligence are a very fast and reliable solution. This is also the optimal approach when rule-based verification of very complex components is too expensive, even with graphical programming. Again, the Artificial Intelligence is able to reliably detect faulty parts even after a training phase with just positive examples.
However, at especially high demands on speed, algorithms or optimal large volume costs, it can be necessary to fall back to the classic programming approach with advanced languages, despite of the above modern alternatives.

Often, production managers and process engineers are willing to automate, but lack access to developers with the requisite expertise in programming and image processing. Without these skills, no image-processing system with high-level languages can be implemented. Seek advice from distributors with technical expertise to choose the appropriate hardware and to access the short-term training process engineers typically need to quickly and effectively implement their own automation systems with smart-camera-based approaches.

ABOUT FRAMOS | “Teaching machines to see” is what drives us. For FRAMOS, image processing is not just a technical discipline, but a fascination, the future and our mission all at the same time. Since 1981 FRAMOS is a leading technology provider in industrial, scientific and medical image processing. Headquartered in Munich and with 4 subsidiaries worldwide we enable manufacturers, system integrators and researchers to benefit from imaging technologies. Our team of approximately 80 associates offers a fully comprehensive portfolio of imaging components, technical consulting and support. Thanks to many years of experience in the industry, we offer engineering services for custom camera development as well as complete turn-key-solutions. We are proud of our participation in a revolutionary technology. The spirit of innovation and development of our early days has remained at the heart of our company.

CONTACT I Stefan Waizmann I Head of Product Management
s.waizmann@framos.com I FRAMOS GmbH