159159

MACHINE VISION SYSTEM HAVING FLEXIBLE SET-UP FEATURES, METHOD AND PRODUCT THEREOF

Field of the Invention

[0001] The present invention relates to machine vision systems used in manufacturing processes or systems. In particular, this invention relates to a machine vision system having flexible set-up features, a method, and a product having a computer program for the method recorded thereon.

Background of the Invention

[0002] Machine vision systems used in manufacturing processes are known to have application software that controls various hardware such as, for example, cameras, lighting and motion tables. Generally, such application software includes different components that are configured to accomplish different tasks by controlling the various hardware. Hence, when new hardware or when a new task is required of existing hardware of a machine vision system, then application software for the vision inspection system has to be re-configured or modified. In some instances, a new machine vision system may be required because a task may not be supported by the existing hardware.

[0003] Re-configuring or modifying application software as such is tedious and costly. One technique to reduce development costs when modifications are needed to application software is provided by US Patent No. 5,742,504 Meyer et al that describes a method and system for quickly developing application software for use in a machine vision system using a computer system. However, US Patent No. 5,742,504 is limited to frame grabbers that support cameras and image sources but not other types of hardware.

[0004] Modifying application software is also described in US Patent No. 6,298,474B1 Blowers et al. in which a method and system interactively develops a

G00002

graphical control-flow structure and associated application software for use in a machine vision system without users having to write codes. A user is thus able to use the graphical control-flow structure via a display interface to modify the application software. However, the graphical control-flow structure cannot simply accommodate new hardware without substantial programming. In some instances, a machine vision system is limited to specific hardware and, therefore, new hardware cannot be accommodated by that machine vision system.

[0005] Therefore, a need clearly exists for a machine vision system that has flexible set-up features to overcome, or at least alleviate, limitations of existing machine vision systems.

Summary of the Invention

[0006] The present invention seeks to provide a machine vision system having flexible set-up features, a method for setting-up the machine vision system, and a product comprising computer usable medium having a computer program for the method recorded thereon.

[0007] Accordingly, in one aspect, the present invention provides a machine vision system having flexible set-up features, the machine vision system comprising:

a plurality of application components, the plurality of application components being respectively associated with a plurality of applications of the machine vision system;

a plurality of image processing components coupled to the plurality of application components;

a plurality of hardware interface components, the plurality of hardware interface components being associated with hardware;

and

a framework, coupled to the plurality of application components and the plurality of hardware interface components, the framework having a plurality of modules for enabling one or more applications of the plurality of applications for the hardware

using one or more hardware interface components of the plurality of hardware interface components and based upon end-user requirements.

[0008] In another aspect, the present invention provides a method for setting-up a machine vision system, the method comprising the steps of:

receiving input from a requestor;

determining access rights of the requestor based upon the input;

and

enabling one or more applications using one or more hardware interface components and based upon end-user requirements when the access rights is indicative of the requestor being permitted to set-up the machine vision system, the end-user requirements being associated with one or more application components and one or more image processing components.

[0009] In a further aspect, the present invention provides a product comprising computer usable medium having a computer program recorded thereon for setting-up a machine vision system, the product comprising:

computer program code means for receiving input from a requestor;

computer program code means for determining access rights of the requestor based upon the input;

and

computer program code means for enabling one or more applications using one or more hardware interface components and based upon end-user requirements when the access rights is indicative of the requestor being permitted to set-up the machine vision system, the end-user requirements being associated with one or more application components and one or more image processing components.

Brief Description of the Drawings

[0010] A preferred embodiment of the present invention will now be more fully described, by way of example, with reference to the drawings of which:

[0011] FIG. 1 illustrates a functional block diagram of a machine vision system in accordance with the present invention;

[0012] FIG. 2 shows a plurality of hardware interface components of the machine vision system of FIG. 1;

[0013] FIG. 3 shows a plurality of application components coupling to a plurality of image processing components of the machine vision system of FIG. 1;

[0014] FIG. 4 is a modular diagram of a framework and associated files of the machine vision system of FIG. 1;

[0015] FIG. 5 is a functional block diagram illustrating interfacing of hardware keys and software keys with the framework of FIG. 4 depending on access rights;

[0016] FIG. 6 is a flowchart of a method for setting-up the machine vision system of FIG. 1;

[0017] FIGs. 7A, 7B and 7C are screen shots of graphical user interfaces for assigning, selecting and configuring in the setting-up method of FIG. 6;

[0018] FIG. 8 is a flowchart of a method for using the machine vision system of FIG. 1 after the setting-up method of FIG. 6;

and

[0019] FIG. 9 is a functional block diagram that illustrates a computer system storing program code means for implementing the methods of FIG. 5 and FIG. 6.

Detailed Description of the Drawings

[0020] A machine vision system having flexible set-up features, a method for setting-up the machine vision system, and a product comprising computer usable medium having a computer program for the method recorded thereon in accordance with a preferred embodiment of the invention are described. In the following description, details are provided to describe the preferred embodiment. It shall be apparent to one skilled in the art, however, that the invention may be practiced without such details. Some of these details may not be described at length so as not to obscure the invention.

[0021] There are many advantages of the preferred embodiment of the invention. One advantage of the invention is that the machine vision system has a framework that enables flexible set-up of one or more applications for different types of hardware based upon end-user requirements. The framework supports these multiple applications in a time-critical multi-threaded client server environment so that such multiple applications can be run concurrently by end-users.

[0022] Another advantage of the invention is that the machine vision system has an open architecture with a framework that is hardware-independent to support the different types of hardware. Therefore, the machine vision system supports not only different frame grabbers but also other types of hardware such as, for example, sensors.

[0023] A further advantage of the invention is that the machine vision system supports application components that include details of various parts associated with the application components. As a result, teaching or introducing a new part or a new hardware for the new part does not require extensive code modifications by developers of the machine vision system. Therefore, such teaching or introducing is simpler compared with existing machine vision systems.

[0024] Yet another advantage of the invention is that the machine vision system has a security scheme to control access rights based upon different levels of users. Hence, developers, configurators or end-users are appropriately accorded access rights when interfacing with the machine vision system. Consequently, the security scheme prevents, or at least alleviates, unauthorized modification of the machine vision system. Among other things, the security scheme involves a hardware key when development, configuration or use of the machine vision system is required.

[0025] Referring now to FIG.1, a functional block diagram of a machine vision system 10 in accordance with a preferred embodiment of the invention is illustrated. The machine vision system 10 comprises a plurality of application components 12, a plurality of hardware interface components 14, a framework 16, a plurality of image processing components 18 coupled to the plurality of application components 12, vision libraries 20 and an image buffer 22. The plurality of hardware interface components 14 includes a grabber interface 24 that couples to the image buffer 22. The plurality of hardware interface components 14 couples to one or more hardware 26.

[0026] The plurality of hardware interface components 14 is associated with hardware coupled to the machine vision system 10. FIG. 2 shows the plurality of hardware interface components 14 that includes the grabber interface 24, a lighting interface 204, a handler interface 206, a production log interface 208, a data acquisition interface 210, a camera interface 212, a motion interface 214, and a security support component 216. The security support component 216 is associated with a hardware key to support secure access to other hardware interface components 14.

[0027] The plurality of application components 12 is respectively associated with a plurality of applications of the machine vision system 10. The plurality of image processing components 18 couples to the plurality of application components 12 and the vision libraries 20. FIG. 3 shows the plurality of application components 12

that includes an alignment component 302, a measurement component 304, a screw surface component 306, a screw dimension component 308, and a 3D ball grid array component 310.

[0028] Also shown in FIG. 3 is the plurality of image processing components 18 coupled to the plurality of application components 12. The plurality of image processing components 18 includes a calibration component 320, a light status component 322, a surface component 324, a 3D construction component 326, and a math component 328.

[0029] It is to be noted that the plurality of application components 12, the plurality of hardware interface components 14, and the plurality of image processing components 18 are based upon Component Object Model (COM) specification as developed by Microsoft®. For example, the plurality of hardware interface components 14 defines a set of COM interface functions. Each of these COM interface functions is associated with one or more driver functions for each of the hardware. Hence, one or more driver functions for a new hardware are already incorporated into one or more of the plurality of hardware interface components 14. Therefore, setting up the machine vision system 10 to support new or different hardware is possible because the set of COM interface functions can now be easier modified using driver functions of the new or different hardware compared with existing machine vision systems.

[0030] The framework 16 couples to the plurality of application components 12 and the plurality of hardware interface components 14 and has a plurality of modules for configuring the machine vision system 10. The plurality of modules of the framework 16 enables configuration of the machine vision system 10 to thereby enable execution of one or more tasks with hardware. Such tasks are associated with at least one application.

[0031] FIG. 4 is a modular diagram 400 of the framework 16 and associated files of the machine vision system 10. The plurality of modules shown in FIG. 4 includes a start-up module 402, an initialization module 404, a configuration module 406, a teach module 408, a production module 410, and a system module 412.

[0032] The plurality of modules couples to various files having data to set-up or to use the machine visions system 10. As shown in FIG. 4, the configuration module 406 couples to configuration files 420. Production files 422 couple to the production module 410 and the system module 412. A part file 424 couples to the teach module 408, the production module 410, and the system module 412. A lot file 426 couples to the production module 410 and the system module 412.

[0033] Other external interfacing with the framework 16 includes application teaching input 440 provided to the teach module 408, hardware key data 450 provided to the start-up module 402, and calibration input 460 provided to the system module 412.

[0034] Referring now to FIG. 5, a functional block diagram illustrates interfacing of hardware keys and software keys with the framework 16 depending on access rights when development, configuration or end-use of the machine vision system 10 is required. The security scheme is implemented with both a hardware key such as, for example, a dongle, and a software key that are associated with either a developer, a configurator or an end-user.

[0035] To develop or to configure the machine vision system 10, a configuration hardware 502 and a configuration software key 504 are required by the framework 16 before development or configuration can take place. The configuration hardware key 502 has various data fields that include a staff identification (ID) 506, an encrypted personal identification number (PIN) 508, license constants 510, application numbers 512 and an application counter 514. The configuration

software key 504 has two data fields, namely, a staff identification 520 and a personal identification number (PIN) 522. The data fields from both the configuration hardware key 502 and the configuration software key 504 enable a configurator to configure the machine vision system 10 via the framework 16.

[0036] To configure the machine vision system 10 via the framework 16, a configurator has to provide input 530 via a graphical user interface (not shown in FIG. 5). Such input 530 includes a hardware key serial number 532 and a plurality of application numbers 534a, 534b, 534c, 534d. The application numbers 534a, 534b, 534c, 534d correspond to a plurality of applications associated with the plurality of application components 12.

[0037] For an end-user to use the machine vision system 10 after configuration by the configurator, an end-user hardware key 540 and an end-user software key 550 are required by the framework 16. The end-user hardware key 540 is based upon the license constants 510 and the hardware key serial number 532. The end-user software key 550 has a data field for software license numbers. These software license numbers is a series of encrypted numbers identifying one or more of the application numbers 534a, 534b, 534c, 534d. The framework 16 also couples to a software license file 560 to access or to provide data related to the software license numbers.

[0038] A method 600 for setting-up the machine vision system 10 is illustrated by a flowchart in FIG. 6. Inputs from a requestor are received at various steps of the method 600 and processed by the framework 16 to determine access rights of the requestor. The method 600 starts at step 602 after which the framework 16 at step 604 extracts security data from a hardware key provided by the requestor intending to access the machine vision system 10. The security support component 216 is used in this extracting step 604.

[0039] At this point, the framework 16 does not know whether the requestor is an end-user, a developer, or a configurator. Based upon the security data, the framework 16 then determines at decision step 606 whether the hardware key is an end-user hardware key.

[0040] The security data provides information that is pre-stored in the hardware key and that identifies the requestor. Thus, if the security data includes the staff ID 506 and the encrypted PIN 508, then such information is read and checked by the framework 16 to confirm that the requestor is a developer or a configurator. In this situation, the hardware key that is provided is then processed as a configuration hardware key 502. Otherwise, if the security data does not include the staff ID 506 and the encrypted PIN 508, but provides the hardware serial lock number 532, then the requestor is determined by the framework 16 as an end-user. In this situation, the hardware key that is provided is then processed as an end-user hardware key.

[0041] For a "Yes", the method 600 proceeds to enable the end-user to use the machine vision system 10 according to access rights of that end-user. The use of the machine vision system 10 by the end-user is described later using another flowchart. Otherwise, with a "No" following the decision step 606, the framework 16 processes any subsequent input from the requestor as input from a developer or a configurator.

[0042] Accordingly, the method 600 proceeds to step 608 at which the framework 16 receives configuration software key input from the requestor. The configuration software key input includes the staff ID 520 and the PIN 522. At decision step 610, the configuration software key input is validated by checking with data stored in the configuration hardware key 502 provided by the requestor. Setting-up the machine vision system 10 is then allowed only if the configuration software key input matches data stored in the hardware key, which has to be a configuration hardware key 502.

[0043] For a "Yes" following the decision step 610, the method 600 proceeds to step 612 at which the framework 16 sets the machine visions system 10 to an idle mode. The idle mode provides a wait state during which the framework 16 is waiting to receive inputs from the requestor. Otherwise, with a "No" following the decision step 610, the method 600 proceeds to step 614 at which the framework 16 exits from method 600 and ends at step 616. By ending as such at step 616, the framework 16 does not allow any development, configuration or use of the machine vision system 10.

[0044] Following the idle mode, the framework may receive inputs from the requestor to configure the machine vision system 10. Such inputs start at step 618 at which the framework 16 assigns one or more applications based upon assigning input received from the requestor and associated with end-user requirements.

[0045] Thereafter, the method 600 continues to step 620 at which the framework 16 receives selecting input from the requestor to select hardware associated with each of the applications. The hardware that is selected is associated with at least one of the hardware interface components 14.

[0046] Upon the selecting step 620, the framework 16 then receives configuring input at step 622 from the requestor to configure the hardware that has been selected and that are associated with each of the applications as assigned at step 618. Configuration data received at the configuring step 622 are saved in the configuration files 420 at step 624.

[0047] The method 600 returns to the idle mode following the saving step 624. The method 600 may end from the idle mode if the requestor decides to exit from the framework 16. In this situation, the requestor provides an exit input that is received by the framework 16 to exit and the method then proceeds to the exit step 614 to, thereafter, end the method at the end step 616. Thus, by ending at the end step 616 in this manner and after the assigning step 618, the selecting step 620, the

configuring step 622, and the saving step 624, the machine vision system 10 is therefore configured for end-usage.

[0048] Therefore, the method 600 enables one or more applications using one or more of the hardware interface components 14 and based upon end-user requirements when access rights is indicative of the requestor being permitted to set-up the machine vision system 10. It is to be noted that the end-user requirements are associated with one or more of the application components 12 and one or more of the image processing components 18. Consequently, having been thus configured, the machine vision system 10 can now be used by end-users provided such end-users are verified by the security scheme of the framework 16.

[0049] FIGs. 7A, 7B and 7C are screen shots of graphical user interfaces 700, 730,760 for the requestor who is a developer or a configurator when carrying out the assigning step 618, the selecting step 620, the configuring step 622, and the saving step 624 of the method 600. These graphical user interfaces 700, 730,760 are provided for entry of inputs by the requestor.

[0050] Referring now to FIG. 8, a flowchart of a method 800 for using the machine vision system 10 after setting-up by the method 600. It is to be noted that the method 800 requires that any requestor or user needs to provide a hardware key from which security data is extracted. Hence, for the method 800, the extracting step 604 of the method 600 is also required and performed as described earlier. Upon completing the extracting step 604 and determining at the decision step 606 that the hardware key is an end-user hardware key associated with an end-user, the method 800 starts at step 802.

[0051] Thereafter, the method 800 continues with the framework 16 being in an idle mode 804. The idle mode 804 provides a wait state during which the framework 16 is waiting to receive end-user inputs that initiate various actions. Such end-user inputs include part teaching at step 806 and change of parameters at

step 808. The part teaching step 806 and the change of parameters step 808 may be optionally saved.

[0052] For tolerance setting of a part, the framework 16 checks at decision step 810 whether the part teaching step 806 for that part has been completed. For a 'Yes', the framework 16 proceeds to step 812 at which tolerance setting for the part is performed. The tolerance setting can be optionally saved at step 812. Otherwise, for a 'No', the framework 16 returns to the idle mode 804. The framework 16 also returns to the idle mode 804 after the tolerance setting step 812.

[0053] For inspecting a part manually, the framework 16 first checks at decision step 814 whether teaching and tolerance setting for the part is completed. For a "Yes" following the decision step 814, the method 800 proceeds to step 816 at which parameters and tolerances for the part are loaded. For a "No" following the decision step 814, the method 800 returns to the idle mode 804.

[0054] Upon loading the parameters and tolerances, the method 800 allows manual inspection of the part at step 818. Manually inspecting the part at step 818 may be performed either with a single inspection or continuously. The choice of single inspection or continuous inspection for a part allows the end-user to check that part teaching for that part has been performed correctly. After manual inspection at the step 818, the method 800 returns to the idle mode 804.

[0055] After the steps of manually inspecting a part, automatic inspection of that part during production may be carried out as follows. The framework 16 first checks at decision step 820 whether teaching and tolerance setting for the part is completed. For a "Yes" following the decision step 820, the method 800 proceeds to step 822 at which parameters and tolerances for the part are loaded. For a "No" following the decision step 820, the method 800 returns to the idle mode 804.

[0056] Upon loading the parameters and tolerances at step 822, the method 800 proceeds to decision step 824 at which the framework 16 determines whether to store production data for the part. For a "Yes" following the decision step 824, the method 800 proceeds to setting up a lot file at step 826 for the part. The lot file keeps track of statistics such as, for example, machine status and part quality for a specified lot of the part.

[0057] After setting up the lot file, the method 800 proceeds to step 828 to inspect the part automatically during production. Step 828 also takes place after a "No" following the decision step 824.

[0058] In addition to the above steps, the method 800 also provides for the idle mode 804 to receive an end-user input to exit from the method 800 at step 830. Upon receiving such an end-user input, the method 800 proceeds to step 832 at which the method 800 ends.

[0059] The steps of the setting-up method 600 or the method 800 for using the machine vision system 10 can be run or executed via a computer program that is recorded on computer usable medium. Such computer usable medium may include random access memories (RAMs), read-only memories (ROMs), hard disks, and the different types of compact discs or floppy disks. A product that comprises the computer program recorded on computer usable medium can be executed with the machine vision system 10 such as, for example, a computer system 900 as illustrated in FIG. 6.

[0060] The computer system 900 can be, for example, a server with different access rights for different users. The computer system 900 comprises a computer 902 that includes a storage media 904 for storing the framework 16. The storage media 904 can serve as the computer usable medium with the computer program recorded thereon. In this case, the storage media 904 can be one or more memory devices such as random access memories (RAMs), read-only memories (ROMs), or

hard disks. Alternatively, the storage media 904 can be a memory device reader such as, for example, a compact disc drive or a floppy disk drive. Such a memory device reader enables a processor 906 to access the computer program of the product.

[0061] A display 908 is coupled to the framework 16 via a video controller 910. The display 908 enables display of graphical user interfaces (GUIs) when executing some of the steps of the setting-up method 600. Data flow within the computer 902 is by means of a data bus 912 that couples the video controller 910, the storage media 904, the processor 906, and a network input/output (I/O) interface 914. User input means comprises a mouse 916 and a keyboard 918 that couples to the data bus 912 via an input/output (I/O) interface 920.

[0062] The computer system 900 is coupled via a communication resource 922 to a network 924. The communication resource 922 may be a cable such as, for example an optical cable, a coaxial cable or a carrier frequency. The network 924 can be an Intranet, a local area network or the Internet.

[0063] The present invention therefore provides the machine vision system 10, the method 600 for setting-up the machine vision system 10 and the product 900 to overcome, or at least alleviate, limitations of existing machine vision systems.

[0064] It will be appreciated that although one preferred embodiment has been described in detail, various modifications and improvements can be made by persons skilled in the art without departing from the scope of the present invention.

Claims

1. A machine vision system having flexible set-up features, said machine vision system comprising:

   a plurality of application components, said plurality of application components being respectively associated with a plurality of applications of said machine vision system;

   a plurality of image processing components coupled to said plurality of application components;

   a plurality of hardware interface components, said plurality of hardware interface components being associated with hardware;

   and

   a framework, coupled to said plurality of application components and said plurality of hardware interface components, said framework having a plurality of modules for enabling one or more applications of said plurality of applications for said hardware using one or more hardware interface components of said plurality of hardware interface components and based upon end-user requirements.

2. The machine vision system of Claim 1, wherein said plurality of hardware interface components defines a set of Component Object Model (COM) interface functions.

3. The machine vision system of Claim 2, wherein each of said set of COM interface functions is associated with one or more driver functions for each of said hardware.

4. The machine vision system of Claim 1, wherein said plurality of hardware interface components comprises a security support component.

5. The machine vision system of Claim 4, and further comprising a hardware key, said hardware key being associated with said security support component.

6. The machine vision system of Claim 5, wherein said hardware key comprises a dongle.

7. A method for setting-up a machine vision system, said method comprising said steps of:

receiving input from a requestor;

determining access rights of said requestor based upon said input;

and

enabling one or more applications using one or more hardware interface components and based upon end-user requirements when said access rights is indicative of said requestor being permitted to set-up said machine vision system, said end-user requirements being associated with one or more application components and one or more image processing components.

8. The method of Claim 7, wherein said receiving step comprises the step of extracting security data from a configuration hardware key using a security support component, said configuration hardware key being provided by said requestor.

9. The method of Claim 8, wherein said receiving step further comprises the step of receiving configuration software key input from said requestor.

10. The method of Claim 9, wherein said determining step comprises the step of validating said configuration software key input with data stored in said configuration hardware key.

11. The method of Claim 10, wherein said enabling step comprises the step of assigning said one or more applications based upon assigning input received from said requestor and associated with said end-user requirements.

12. The method of Claim 11, wherein said enabling step further comprises the step of selecting hardware based upon selecting input received from said requestor and associated with each of said one or more applications, said hardware being associated with at least one of said hardware interface components.

13. The method of Claim 12, wherein said enabling step further comprises the step of configuring said hardware based upon configuring input received from said requestor and associated with each of said one or more applications.

14. The method of Claim 13, wherein said configuring step comprises the step of providing one or more graphical user interfaces for entry of input by said requestor.

15. A product comprising computer usable medium having a computer program recorded thereon for setting-up a machine vision system, said product comprising:

computer program code means for receiving input from a requestor;

computer program code means for determining access rights of said requestor based upon said input;

and

computer program code means for enabling one or more applications using one or more hardware interface components and based upon end-user requirements when said access rights is indicative of said requestor being permitted to set-up said machine vision system, said end-user requirements being associated with one or more application components and one or more image processing components.

16. The product of Claim 15, wherein said computer program code means for receiving comprises computer program code means for extracting security data from a configuration hardware key using a security support component, said configuration hardware key being provided by said requestor.

17. The product of Claim 16, wherein said computer program code means for receiving further comprises computer program code means for receiving configuration software key input from said requestor.

18. The product of Claim 17, wherein said computer program code means for determining comprises computer program code means for validating said configuration software key input with data stored in said configuration hardware key.

19. The product of Claim 18, wherein said computer program code means for enabling comprises computer program code means for assigning said one or more applications based upon assigning input received from said requestor and associated with said user requirements.

20. The product of Claim 19, wherein said computer program code means for enabling further comprises computer program code means for selecting hardware based upon selecting input received from said requestor and associated with each of said one or more applications, said hardware being associated with at least one of said hardware interface components.

21. The product of Claim 20, wherein said computer program code means for enabling further comprises computer program code means for configuring said hardware based upon configuring input received from said requestor and associated with each of said one or more applications.

22. The product of Claim 21, wherein said computer program code means for configuring comprises computer program code means for providing one or more graphical user interfaces for entry of input by said requestor.