February 2021

MACHINE LEARNING BASED DISTORTION CORRECTION

Kathan Shukla

MACHINE LEARNING BASED DISTORTION CORRECTION

ABSTRACT

A vehicle system may use a computer vision module to rectify distortions in images captured by one or more fish-eye cameras (hereinafter referred to as "fish-eye camera images") of the surroundings of a vehicle (e.g., an automobile, a motorcycle, a bus, a recreational vehicle (RV), a semi-trailer truck, a tractor or other type of farm equipment, train, a plane, a boat, a helicopter, a personal transport vehicle, etc.). In some examples, a machine-learning model (hereinafter referred to as "ML model") may calibrate (or, in other words, tune) intrinsic parameters (e.g., the fish-eye camera images' focal length ((x,y) coordinates), principal point ((x,y) coordinates), field of view (FOV) scale, etc.) used by the computer vision module. The ML model may calibrate the intrinsic parameters by identifying, in fish-eye camera images, specific points (e.g., elements, features, objects, etc.) with well-defined and known geometries in reality (e.g., such as straight lines, square corners, etc.) that have been distorted by the fish-eye cameras. The ML model may be trained to determine intrinsic parameters that correspond to a geometrical transformation that, when applied to the fish-eye camera images, dewarps the specific points such that the dewarped images accurately represent the specific points' geometries. The computer vision module may then generate maps that denote this geometrical transformation and apply the geometrical transformation to the fish-eye camera images to produce dewarped images.

DESCRIPTION

The present disclosure describes a vehicle system using a computer vision module to rectify distortions in images captured by one or more fish-eye cameras (hereinafter referred to as "fish-eye camera images") of the surroundings of a vehicle (e.g., an automobile, a motorcycle, a bus, a recreational vehicle (RV), a semi-trailer truck, a tractor or other type of farm equipment, train, a plane, a boat, a helicopter, a personal transport vehicle, etc.). In some examples, a machine-learning model (hereinafter referred to as “ML model”) may calibrate (or, in other words, tune) intrinsic parameters (e.g., the fish-eye camera images’ focal length ((x,y) coordinates), principal point ((x,y) coordinates), field of view (FOV) scale, etc.) used by the computer vision module. Responsive to receiving and based on the calibrated intrinsic parameters, the computer vision module may generate maps that denote this geometrical transformation and apply the geometrical transformation to the fish-eye camera images to produce accurate, dewarped images.

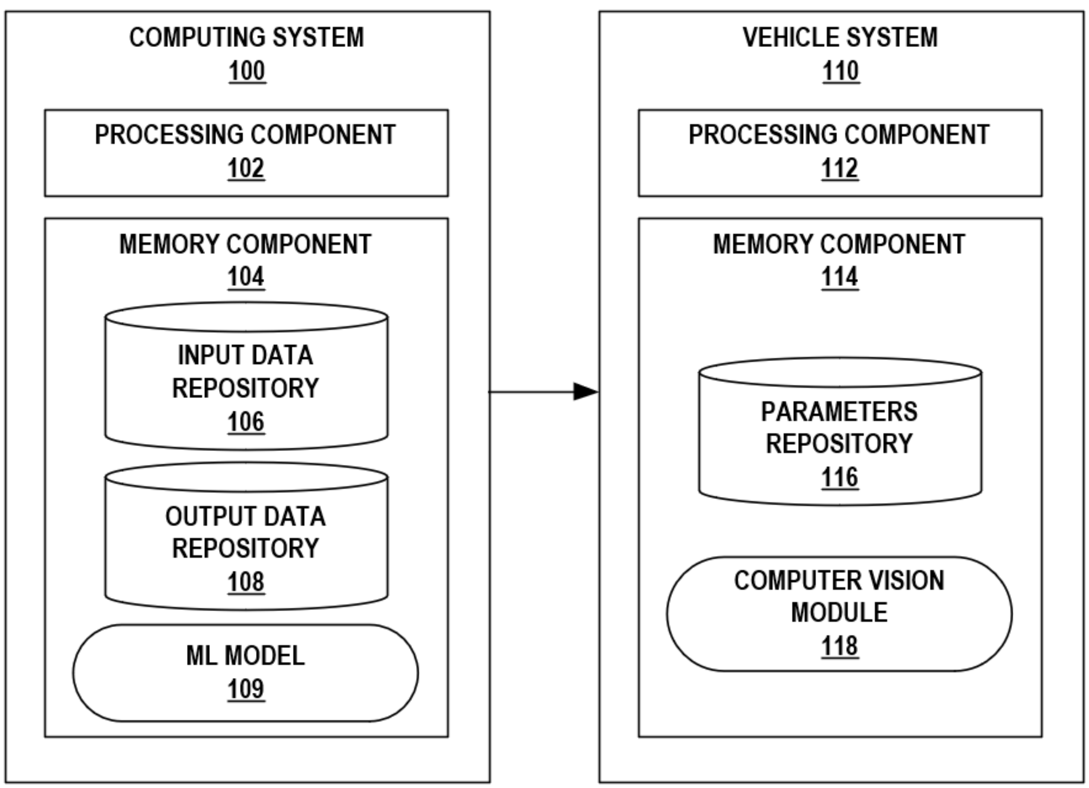

FIG. 1

FIG. 1 is a conceptual diagram illustrating a vehicle system 110 that uses a computer vision module 118 to rectify distortions in fish-eye camera images of the surroundings of a vehicle. As shown in FIG. 1, vehicle system 110 may include a processing component 112 and a memory component 114. Memory component 114 may store a parameters repository 116 and computer vision module 118. As shown in FIG. 1, vehicle system 110 may receive data, such as intrinsic parameters calibrated by a ML model 109, from a computing system 100. Computing system 100 may include a processing component 102 and a memory component 104. Memory component 104 may store an input data repository 106, an output data repository 108, and ML model 109.

Vehicle system 110 may represent an integrated head unit that controls vehicle systems, such as a camera system, an audio system, a heating, ventilation, and air conditioning (HVAC) system, a lighting system (for controlling interior and/or exterior lights), an infotainment system, a seating system (for controlling a position of a driver and/or passenger seat), and the like. Vehicle system 110 may be included in an automobile, a motorcycle, a bus, a RV, a semi-trailer truck, a tractor or other type of farm equipment, a train, a plane, a drone, a helicopter, a personal transport vehicle, or the like.

Computing system 100 may be any suitable remote computing system capable of calibrating and sending intrinsic parameters to vehicle system 110, in accordance with techniques described here. For example, computing system 100 may be one or more desktop computers, laptop computers, mainframes, servers, cloud computing systems, virtual machines, and the like. In some examples, computing system 100 may represent a cloud computing system that provides one or more services, such as distributing calibrated intrinsic parameters. That is, in some examples, computing system 100 may be a distributed computing system. One or more vehicle systems (e.g., vehicle system 110) may access the services (e.g., to download intrinsic parameters) provided by computing system 100.

Processing components 102 and processing components 112 may implement functionality and/or execute instructions associated with computing system 100 and vehicle system 110, respectively. Examples of processing component 102, 112 may include one or more of an application specific integrated circuit (ASIC), a field programmable gate array (FPGA), an application processor, a display controller, an auxiliary processor, a central processing unit (CPU), a graphics processing unit (GPU), one or more sensor hubs, and any other hardware configure to function as a processor, a processing unit, or a processing device. ML model 109 may be operable (or, in other words, executed) by processing component 102 to perform various actions, operations, or functions of computing system 100. Similarly, computer vision module 120 may be operable by processing component 112 to perform various actions, operations, or functions of vehicle system 110.

Memory components 104 and memory components 114 may store information used by computing system 100 and vehicle system 110, respectively. In some examples, memory components 104, 114 may include one or more computer-readable storage media. For example, memory component 104, 114 may be configured for long-term, as well as short-term storage of information, such as instructions, data, or other information. In some examples, memory components 104, 114 may include non-volatile storage elements. Examples of such non-volatile storage elements may include magnetic hard discs, optical discs, solid state discs, floppy discs, flash memories, forms of electrically programmable memories (e.g., EPROMs), electrically erasable and programmable memories (e.g., EEPROMs), and the like. In other examples, in place of, or in addition to the non-volatile storage elements, memory component 104 may include one or more so-called "temporary" memory devices, meaning that a primary purpose of these devices may not be long-term data storage. For example, the devices may comprise volatile memory devices, meaning that the devices may not maintain stored contents when the devices are not receiving power. Examples of volatile memory devices include random access memories (RAM), dynamic random access memories (DRAM), static random access memories (SRAM), and the like.

In some examples, ML model 109 of computing system 100 may represent one or more artificial neural networks (also referred to as neural networks). A neural network may include a group of connected nodes, which also may be referred to as neurons or perceptrons. A neural network may be organized into one or more layers. Neural networks that include multiple layers may be referred to as “deep” networks. A deep network may include an input layer, an output layer, and one or more hidden layers positioned between the input layer and the output layer. The nodes of the neural network may be connected or non-fully connected.

In some examples, ML model 109 may represent one or more convolutional neural networks. In some instances, a convolutional neural network may include one or more convolutional layers that perform convolutions over input data (e.g., stored in input data repository 106) using learned filters. Filters may also be referred to as kernels. Convolutional neural networks may be especially useful for vision problems (e.g., when the input data includes imagery such as still images or video).

In general, vehicle system 110 may use one or more fish-eye cameras to capture wide-angle images of the surroundings of a vehicle. However, fish-eye camera images may include distortions, such as a barrel distortion where image magnification decreases with distance from the optical axis. As a result, fish-eye camera images may appear to be mapped around a sphere or barrel. Such distortions may make interpreting and/or processing (e.g., by a driver, an advanced driver assistance system, etc.) the fish-eye camera images difficult. As such, it may be advantageous to at least partially rectify the distortions present in fish-eye camera images to produce dewarped images that accurately represent reality.

In accordance with techniques of this disclosure, vehicle system 110 may use computer vision module 118 to rectify distortions in fish-eye camera images of the surroundings of a vehicle. Computer vision module 118 may use intrinsic parameters (e.g., the fish-eye camera images' focal length ((x,y) coordinates), principal point ((x,y) coordinates), field of view (FOV) scale, etc.) calibrated by ML model 109. To calibrate the intrinsic parameters, ML model 109 may first receive input data from input data repository 106. The input data may include training data, such as imagery captured by fish-eye cameras (of the vehicle in which vehicle system 110 is included), and input data that has labels (or, in other words, ground-truth data).

Ground-truth data for ML model 109 may include specific points (e.g., elements, features, objects, etc.) in the imagery (e.g., of the surroundings of a vehicle) with geometries that are well-defined and known in reality (e.g., such as straight lines, square corners, etc.). Examples of specific points may include traffic signs (e.g., a stop sign, a speed limit sign, a yield sign, etc.), which typically have geometries defined in reality by straight lines. These specific points may be annotated (e.g., manually or automatically) with a label and bounding box that serve as ground-truth annotation labels. Using this ground-truth data, ML model 109 may be trained to reliably and accurately detect these specific points in undistorted images.

Subsequent to the training of ML model 109, ML model 109 may detect the specific points in dewarped images produced by ML model 109. For example, ML model 109 may receive arbitrary intrinsic parameters for producing dewarped images and then detect traffic signs in the dewarped images.

To calibrate the intrinsic parameters, ML model 109 may determine a loss function that compares (e.g., determines a difference between) the detection results (e.g., the number, type, and/or the like of traffic signs detected by ML model) for the dewarped images produced by ML model 109 to the detection results for undistorted images representing ground-truth data. ML model 109 may then optimize the loss function by determining the intrinsic parameters that results in an optimal number of differences (e.g., 0 differences) between the detection results for the dewarped images and for the undistorted images. Responsive to determining the calibrated intrinsic parameters, ML model 109 may store the calibrated intrinsic parameters in output data repository 108.

Computing system 100 may send the calibrated intrinsic parameters stored in output data repository 108 to vehicle system 110. Responsive to receiving the calibrated intrinsic parameters from output data repository 108 of computing system 100, vehicle system 110 may store the calibrated intrinsic parameters in parameters repository 116. Computer vision module 118 of vehicle system 110 may use the calibrated intrinsic parameters stored in parameters repository 116 to generate maps for dewarping the fish-eye camera images. The generated maps may denote a geometrical transformation for mapping the domain of a distorted image to the domain of an ideal image with no distortions.

For example, based on the calibrated intrinsic parameters, computer vision module 118 may map a fish-eye camera image (e.g., captured by a fish-eye camera of a vehicle) to at least two or more planar images that do not have distortion effects introduced by the fish-eye camera. The planar images may then be combined to form a single dewarped image corresponding to the fish-eye camera image. In this way, computer vision module 118 may apply the geometrical transformation to the fish-eye camera images to produce dewarped images that accurately represent reality.

One or more advantages of the techniques described here include rectifying distortions present in fish-eye camera images to produce dewarped images that accurately represent reality. These dewarped images may increase driving safety by, for example, making it easier for a driver to interpret the imagery of the surroundings of a vehicle. As a result, the driver may be more aware of the surroundings of the vehicle, potentially resulting in fewer traffic accidents. Similarly, dewarped images may be easier for an advanced driver assistance system to process. Thus, rectifying the distortions in fish-eye camera images captured by fish-eye cameras of a vehicle may increase the reliability of functions (e.g., object detection) performed by the advanced driver assistance system.

A brief overview of example machine-learned models and associated techniques has been provided by the present disclosure. For additional details, readers should review the following references: Machine Learning A Probabilistic Perspective (Murphy); Rules of Machine Learning: Best Practices for ML Engineering (Zinkevich); Deep Learning (Goodfellow); Reinforcement Learning: An Introduction (Sutton); and Artificial Intelligence: A Modern Approach (Norvig). It is noted that the techniques of this disclosure may be combined with any other suitable technology or combination of technologies, including those listed as references below.

References

1. US Patent Application No. 2015/0049193
2. US Patent Application No. 2020/0117918

3. Jeereddy, “Pedestrian Detection on Dewarped Fisheye Images using Deep Neural Networks.” Faculty of Computing, Blekinge Institute of Technology, May 2019.
4. Xiaoqing et al., “FishEyeRecNet: A Multi-Context Collaborative Deep Network for Fisheye Image Rectification.” arXiv, April 13, 2018.
5. Martin et al., “A Survey of App Store Analysis for Software Engineering.” IEEE Transactions on Software Engineering, September 2017.