



- (51) **International Patent Classification:**
G06K 9/00 (2006.01) *G06N 3/04* (2006.01)
- (21) **International Application Number:**
PCT/US2015/066664
- (22) **International Filing Date:**
18 December 2015 (18.12.2015)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
62/094,681 19 December 2014 (19.12.2014) US
- (71) **Applicant:** UNITED TECHNOLOGIES CORPORATION [US/US]; 1 Financial Plaza, Hartford, Connecticut 06103 (US).
- (72) **Inventors:** GIERING, Michael J.; 39 Tumblebrook Road, Bolton, Connecticut 06043 (US). REDDY, Kishore K.; 75 Hockanum Blvd., Apt. 2413, Vernon, Connecticut 06066 (US). VENUGOPALAN, Vivek; 358 Ellen Lane, Bridgewater, New Jersey 08807 (US).
- (74) **Agent:** BARON, Eric J.; 20 Church Street, Hartford, Connecticut 06103 (US).
- (81) **Designated States** (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,

HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

- (84) **Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

- as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))
- as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))
- of inventorship (Rule 4.17(iv))

Published:

- with international search report (Art. 21(3))
- before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))

(54) **Title:** MULTI-MODAL SENSOR DATA FUSION FOR PERCEPTION SYSTEMS

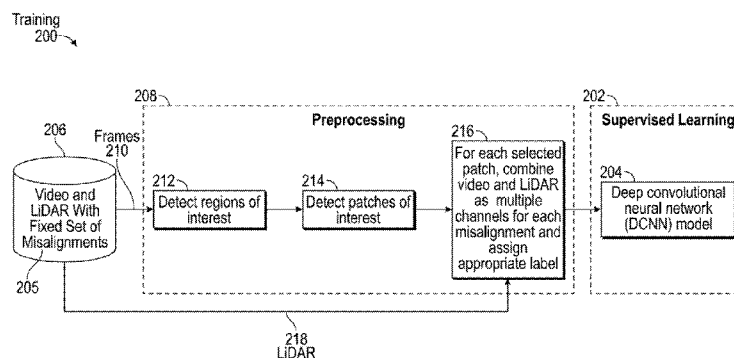


FIG. 2

(57) **Abstract:** A method includes fusing multi-modal sensor data from a plurality of sensors having different modalities. At least one region of interest is detected in the multi-modal sensor data. One or more patches of interest are detected in the multi-modal sensor data based on detecting the at least one region of interest. A model that uses a deep convolutional neural network is applied to the one or more patches of interest. Post-processing of a result of applying the model is performed to produce a post-processing result for the one or more patches of interest. A perception indication of the post-processing result is output.

MULTI-MODAL SENSOR DATA FUSION FOR PERCEPTION SYSTEMS

BACKGROUND

[0001] This disclosure relates to processing of sensor data, and more particularly to multi-modal sensor data fusion for perception systems.

[0002] Navigation and situational awareness of optionally manned vehicles typically requires the integration of multiple sensing modalities, such as Light Detection and Ranging (LiDAR) for depth and video for two-dimensional imaging. To extract meaning from multiple sensors, the sensor data must be fused to relate time series data from a variety of sensors that capture spatial information. For many tasks, such as pedestrian and object detection tasks that make use of multiple sensors, decision support methods rest on the assumption of proper data fusion. One aspect of data fusion is registration, where data collected from sensors having different coordinate systems are merged into a common coordinate system. Typical approaches in LiDAR-video registration, for instance, build separate vision and LiDAR feature extraction methods and identify common anchor points in both.

BRIEF DESCRIPTION

[0003] According to an embodiment, a method includes fusing multi-modal sensor data from a plurality of sensors having different modalities. At least one region of interest is detected in the multi-modal sensor data. One or more patches of interest are detected in the multi-modal sensor data based on detecting the at least one region of interest. A model that uses a deep convolutional neural network is applied to the one or more patches of interest. Post-processing of a result of applying the model is performed to produce a post-processing result for the one or more patches of interest. A perception indication of the post-processing result is output.

[0004] In addition to one or more of the features described above, or as an alternative, further embodiments could include where detecting the at least one region of interest includes performing edge detection on one or more images in the multi-modal sensor data.

[0005] In addition to one or more of the features described above, or as an alternative, further embodiments could include where the edge detection includes occlusion edge detection.

[0006] In addition to one or more of the features described above, or as an alternative, further embodiments could include where the model is trained using a supervised learning process for auto-registration of the multi-modal sensor data and outputs classification values.

[0007] In addition to one or more of the features described above, or as an alternative, further embodiments could include where the post-processing includes aggregating the classification values and identifying a dominating label representing an alignment offset between at least two of the sensors to determine whether a misalignment exists.

[0008] In addition to one or more of the features described above, or as an alternative, further embodiments could include where the model is trained using an unsupervised learning process using a deep auto-encoder and outputs reconstruction errors.

[0009] In addition to one or more of the features described above, or as an alternative, further embodiments could include where the post-processing includes aggregating the reconstruction errors and identifying a misalignment between at least two of the sensors based on the aggregation of the reconstruction errors.

[0010] In addition to one or more of the features described above, or as an alternative, further embodiments could include where the perception indication is a misalignment indicator used to adjust at least one actuator of the at least two sensors having the misalignment.

[0011] In addition to one or more of the features described above, or as an alternative, further embodiments could include where the model is trained using a supervised learning process for edge and occlusion edge labeling of the multi-modal sensor data and outputs classification values.

[0012] In addition to one or more of the features described above, or as an alternative, further embodiments could include where the post-processing includes aggregating the classification values and smoothing the classification, and further where the perception indication includes identification of occlusion edges.

[0013] In an embodiment a system includes a sensor system and a perception processor. The sensor system includes a plurality of sensors having different modalities. The perception processor is operable to fuse multi-modal sensor data from the sensors, detect at least one region of interest in the multi-modal sensor data, detect one or more patches of interest in the multi-modal sensor data based on detecting the at least one region of interest, apply a model including a deep convolutional neural network to at least one detected patch of interest, perform post-processing of a result of applying the model to produce a post-processing result for the at least one detected patch of interest, and output a perception indication of the post-processing result.

[0014] Technical function of the embodiments described above includes fusion of data from different sensor modalities, model development, sensor actuator control, and/or occlusion edge detection.

[0015] Other aspects, features, and techniques of the embodiments will become more apparent from the following description taken in conjunction with the drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] The subject matter which is regarded as the present disclosure is particularly pointed out and distinctly claimed in the claims at the conclusion of the specification. The foregoing and other features, and advantages of the present disclosure are apparent from the following detailed description taken in conjunction with the accompanying drawings in which:

[0017] FIG. 1 illustrates a vehicle with a perception system according to an embodiment;

[0018] FIG. 2 illustrates a training process using supervised learning to train a model according to an embodiment;

[0019] FIG. 3 illustrates a relationship between channels, patches, and images according to an embodiment;

[0020] FIG. 4 illustrates a deep convolutional neural network model according to an embodiment;

[0021] FIG. 5 illustrates a testing process using the model of FIG. 2 according to an embodiment;

[0022] FIG. 6 illustrates a training process using unsupervised learning to train a model according to an embodiment;

[0023] FIG. 7 illustrates a testing process using the model of FIG. 6 according to an embodiment;

[0024] FIG. 8 illustrates a training process using occlusion edge detection to train a model according to an embodiment; and

[0025] FIG. 9 illustrates a testing process using the model of FIG. 8 according to an embodiment.

DETAILED DESCRIPTION

[0026] Embodiments dynamically fuse information from available data channels for perception related tasks and alleviate the need for anchor points between sensor modalities.

Example sensor modalities include LiDAR, video, Radio Detection And Ranging (RADAR), Short Wavelength Infrared (SWIR), and Global Positioning System (GPS) sensor data, for instance. Additionally, derived sensor modalities can be used as data channels, such as horizontal and vertical components of optical flow. Optical flow provides information of scene dynamics and is expressed as an estimate of velocity at each pixel from two consecutive image frames. To determine optical flow, a motion field change between two frames can be measured by the motion of a pixel brightness pattern, where the changes in image brightness are due to the camera and/or object motion. Embodiments can include training a model and testing of multi-modal sensor data using the model to perform various perception related tasks such as sensor alignment adjustment, visualization, and navigation decision making.

[0027] Referring now to the drawings, FIG. 1 illustrates a vehicle 100 with a perception system 101 according to an embodiment. Although vehicle 100 is depicted as a ground vehicle, embodiments can be applied to any type of vehicle or mobile machine that is optionally manned or autonomous, such as an aircraft, watercraft, spacecraft, robot, or the like. The perception system 101 includes a perception processor 102 interfaced with a sensor system 104 that includes multiple modalities. In the example of FIG. 1, sensor system 104 includes at least one imaging sensor 106 and at least one ranging sensor 108. The imaging sensor 106 can be a video camera that provides color (e.g., red (R), green (G), blue (B)) and/or grayscale images to the perception processor 102. The ranging sensor 108 can be a LiDAR sensor or other similar sensor capable of determining depth/distance (e.g., an ultrasonic sensor). The perception processor 102 can control positioning of the imaging sensor 106 using actuator 110 and can control positioning of the ranging sensor 108 using actuator 112. The sensor system 104 can include other sensors types, such as gyroscopes, GPS, accelerometers, and the like.

[0028] The perception processor 102 is a processing system which can include memory to store instructions that are executed by one or more processors. The executable instructions may be stored or organized in any manner and at any level of abstraction, such as in connection with a controlling and/or monitoring operation of the sensor system 104 of FIG. 1. The one or more processors can be any type of central processing unit (CPU), including a microprocessor, a digital signal processor (DSP), a microcontroller, an application specific integrated circuit (ASIC), a field programmable gate array (FPGA), or the like. Also, in embodiments, the memory may include random access memory (RAM), read only memory (ROM), or other electronic, optical, magnetic, or any other computer

readable medium onto which is stored data and algorithms in a non-transitory form. The perception processor 102 can include one or more graphics processing units (GPUs) which may support vector processing using a single instruction multiple datapath (SIMD) architecture to process multiple layers of data substantially in parallel.

[0029] FIG. 2 illustrates a training process 200 using supervised learning 202 to train a model 204 according to an embodiment. In the example of FIG. 2, a data source 206 provides training data 205 to develop the model 204 after preprocessing 208 is performed. The training data 205 in data source 206 can originate from data captured by the sensor system 104 of FIG. 1 during a training phase. The training process 200 may be performed as part of an off-line process using a separate processing system other than the perception processor 102 of FIG. 1. Alternatively, the perception processor 102 may be configured in a training phase to implement the training process 200 of FIG. 2.

[0030] In the example of FIG. 2, training data 205 includes video and LiDAR data with a fixed set of misalignments. For instance, since the imaging sensor 106 and ranging sensor 108 are mounted at different positions on the vehicle 100 of FIG. 1, the sensors 106 and 108 may not be aligned with precision for the same field of view. Further, the actuators 110 and 112 enable the sensors 106 and 108 to be dynamically repositioned. For purposes of training, one or more sets of misaligned data with a known degree of misalignment are used to train the model 204. Video frame data 210 from training data 205 can be provided to a region-of-interest detector 212 that may perform edge detection or other types of region detection known in the art as part of preprocessing 208. A patch detector 214 can detect patches (i.e., areas) of interest based on the regions of interest identified by the region-of-interest detector 212 as part of preprocessing 208. For example, a threshold can be applied on a percentage of pixels with edges in a given patch. A labeler 216 merges LiDAR data 218 from the training data 205 with video data 210 from selected patches of interest as detected by the patch detector 214 as part of preprocessing 208. For each selected patch, the labeler 216 fuses video frame data 210 and LiDAR data 218 as multiple channels for each misalignment and assigns labels. For example, if the training data 205 includes nine different misalignment sets, then labeling can be performed for each of the nine sets of misaligned data. The labeled data from the labeler 216 is used for supervised learning 202 to train model 204 using a convolutional neural network (CNN) which may also be referred to as a deep CNN or DCNN. The model 204 for auto-registration is trained to classify the current misalignment of the LiDAR-video data streams into a predefined set of offsets. Supervised learning 202 compares classification results of the model 204 to a ground truth and can

continue running iterations of the model 204 until a desired level of classification confidence is achieved. The preprocessing 208 and training of the model 204 of FIG. 2 are further described with respect to FIGS. 3 and 4.

[0031] FIG. 3 illustrates a relationship between channels 302, patches 304, and images 306. As one example, at each video frame time step, inputs to preprocessing 208 of FIG. 2 include C channels of data with C ranging from three to six channels 302. The channels 302 can include grayscale Gr or (R,G,B) information from the video frames 210, horizontal and vertical components of optical flow (U,V) derived from changes between the video frames 210, and depth information L from LiDAR data 218. The data from each modality in channels 302 are reshaped to a fixed size of 800 x 256 values in the example of FIG. 3 as images 306, which are partitioned into p x p patches 304 at a prescribed stride. Each p x p patch 304 is stacked across C channels 302, effectively generating a vector of C dimensions. Preprocessing parameters are denoted by patch size p, stride s, and the number of input channels C.

[0032] Preprocessing 208 is repeated N times, where N is the number of offset classes. For each offset class, the video (R,G,B) and optical flow (U,V) channels can be kept static, and the depth (L) channel from the LiDAR data 218 can be shifted by an offset misalignment between the imaging sensor 106 and the ranging sensor 108 of FIG. 1. In order to accurately detect the misalignment in the sensor data, a threshold can be set to limit the information available in each channel. The LiDAR data 218 typically has regions of sparsity and hence the LiDAR patches with a variance (e.g., $\sigma^2 < 15\%$) are dropped from the final data set. This leads to the elimination of the majority of foreground patches in the data set, reducing the size of the training set.

[0033] FIG. 4 illustrates a deep convolutional neural network (DCNN) model 400 according to an embodiment. The DCNN model 400 is an example of the model 204 of FIG. 2. The DCNN model 400 shares weights in the training phase (e.g., training process 200), resulting in fewer model parameters and more efficient training as compared to other machine learning methods. DCNNs in general are particularly useful for problems in which local structure is important, such as object recognition in images and temporal information for voice recognition. Alternating steps of convolution and pooling generate features at multiple scales, which in turn imbue the DCNN model 400 with scale invariant characteristics.

[0034] The DCNN model 400 is a feedforward artificial neural network that has more than one layer of hidden units between its inputs and outputs. Each hidden unit, j, uses a

nonlinear mapping function, often the logistic function, to map its total input from the layer below, x_j , to the scalar state, y_j , that it sends to the layer above, where b_j is the bias of unit j , i is an index over units in the layer below, and w_{ij} is the weight to unit j from unit i in the layer below. The values of y_j and x_j can be computed according to equation 1.

$$y_j = \frac{1}{1+e^{-x_j}} \text{ where } x_j = b_j + \sum_i y_i w_{ij} \quad (\text{Equation 1})$$

[0035] For DCNNs, convolution is performed at convolutional layers to extract local structure features from the features of the previous layer. Additive bias is applied at this point. This is followed by a local pooling step. A nonlinear mapping (e.g., a sigmoid) is applied after either the convolution or pooling layer and can vary by implementation. Iteratively repeating the convolution and pooling steps results in a DCNN architecture for DCNN model 400 as illustrated in FIG. 4.

[0036] The value for each spatial point (x,y) on the j th feature map in the i th layer is denoted as v according to equation 2, where b_{ij} is the bias for the feature map, m indexes over the set of feature maps in the $(i-1)$ th layer connected to the current feature map, w is the value at the position (p,q) of the kernel connected to the k th feature map, and P_i and Q_i are the height and width of the kernel respectively.

$$v_{ij}^{xy} = \tanh(b_{ij} + \sum_m \sum_p^{P_i-1} \sum_q^{Q_i-1} w_{ijm}^{pq} v_{(i-1)m}^{(x+p)(y+q)}) \quad (\text{Equation 2})$$

[0037] In the context of sensor registration, the target layer of the DCNN model 400 can be chosen to represent the degree to which sensor information is misaligned. This information can in turn be used to properly register sensor data by physical manipulation of the sensors or within the system software. Methods for improving the accuracy of DCNN model 400, such as dropout, may be applied. Dropout (e.g., averaging results over multiple lower dimensional versions of the model) can be especially useful for applications where the amount of available data is of a marginal size to learn the number of necessary parameters in the DCNN model 400.

[0038] In embodiments having initial data with sensors properly registered, the DCNN model 400 can be trained on versions of the data at various known offsets, such as the example of FIG. 2 to train model 204. The input to the DCNN model 400 can be a matrix representation of channels 408 from sensors 106, 108 and any supplemental information nodes (e.g., LiDAR 402, video 404, optical flow 406). The DCNN model 400 can be created with standard iterative layers of convolution 410 and pooling 412, terminating in a soft-max layer 414 for classification of any input sensor pairings as one of the known offsets. The

soft-max layer 414 is a fully connected layer that enables the offset prediction to be interpreted as a distribution or as a discrete classification result.

[0039] In the example of FIG. 4, the DCNN model 400 includes three pairs of convolution-pooling layers that estimate the offset between the LiDAR-video inputs at each time step. For each patch within a time step, there are N variants with the LiDAR-video-optical flow inputs offset by the predetermined amounts. The DCNN model 400 outputs to soft-max layer 414, thereby providing an offset classification value for each patch of the frame. Patches, such as patches 304 of FIG. 3, may be implemented as 32 x 32 patches stacked across the different channels and provided as the input channels 408 to the DCNN model 400. All six channels 408 (RGBLUV) may be used or fewer/alternate input configurations can be used in embodiments. As one example, a first convolutional layer 410A uses 32 filters (or kernels) of size 5 x 5 x C with a stride of 1 pixel and padding of 2 pixels on the edges. A first pooling layer 412A generates input data (of size 16 x 16 x 32) for a second convolutional layer 410B. The second convolutional layer 410B can use 32 filters of size 5 x 5 x 32 with a stride of 1 pixel and padding of 2 pixels on the edges. A second pooling layer 412B, similar to the first pooling layer 412A is used to generate input with size 8 x 8 x 32 for a third convolutional layer 410C that uses 64 filters of size 5 x 5 x 32 with the stride and padding configured the same as the second convolutional layer 410B. The third pooling layer 412C with a similar configuration as the first and second pooling layers 412A, 412B connects to soft-max layer 414 with labels corresponding to classes generated from the shifted data. In one embodiment, the DCNN model 400 was trained using stochastic gradient descent with a mini-batch size of 100 epochs. The DCNN model 400 can be configured with Rectified Linear Units (ReLUs) that train several times faster than equivalents with tanh connections.

[0040] FIG. 5 illustrates a testing process 500 using the model 204 of FIG. 2 according to an embodiment. The testing process 500 can be implemented by the perception system 102 of FIG. 1. Multi-modal sensor data 502 from sensors 106 and 108 having different modalities can be fused in a live stream from sensor system 104, for instance, including video and LiDAR data. Fusing of multi-modal sensor data 502 can also include creating one or more derived input channels for data derived from sensor data, such as optical flow. A region-of-interest detector 504 analyzes the multi-modal sensor data 502 to detect any regions of interest such as edges. The region-of-interest detector 504 can perform the same operations as the region-of-interest detector 212 of FIG. 2, for instance, using edge detection techniques known in the art.

[0041] If no regions of interest (e.g., edges) are detected by the region-of-interest detector 504, then block 506 continues processing of the multi-modal sensor data 502 as more samples/frames of data are acquired. If at least one region of interest is detected by the region-of-interest detector 504, then block 506 passes the region of interest information to a patch detector 508. The patch detector 508 can perform the same operations as the patch detector 214 of FIG. 2. For example, the patch detector 508 can detect patches (i.e., areas) of interest based on the regions of interest identified by the region-of-interest detector 504. A threshold can be applied on a percentage of pixels with edges in a given patch to reduce the data set size passed to the model 204 trained by training process 200 of FIG. 2.

[0042] At block 510, if the patch detector 508 does not detect at least one patch of interest, then processing of the multi-modal sensor data 502 continues as more samples/frames of data are acquired. If the patch detector 508 detects one or more patches of interest in the multi-modal sensor data 502, then the model 204 is applied to the one or more patches of interest using a deep convolutional neural network based on the training process 200 of FIG. 2. In the example of FIG. 5, a result of the model 204 includes classification values 512 that are provided to post-processing 514.

[0043] The post-processing 514 can include aggregating 516 the classification values 512 and identifying 518 a dominating label representing an alignment offset between sensors 106, 108 to determine whether a misalignment exists. For example, various horizontal and vertical offsets between sensors 106 and 108 can have classification values assigned (e.g., class1 = no offset, class2 = pixel offset (12, 32), class3 = pixel offset (0, 24), class4 = pixel offset (16, 8), class5 = pixel offset (20,-16), etc.). When classifications are aggregated from each patch for a given frame, a dominating (i.e., most common) label is used as a final alignment/misalignment indicator post-processing result. At block 520, if the alignment is deemed to be correct (e.g., within a threshold misalignment), then processing of the multi-modal sensor data 502 continues as more samples/frames of data are acquired. If there is a misalignment detected at block 520, then a decision making block 522 determines how to adjust either or both of the actuators 110, 112 by mapping 524 the dominating label from block 518 to positioning of the actuators 110, 112. For instance, if the dominating label maps to a misalignment offset X/Y between the actuators 110, 112, the decision making block 524 can command adjustments to either or both of the actuators 110, 112 to correct the misalignment. Commanding adjustments to both actuators 110, 112 in parallel may correct the misalignment faster as both move a smaller amount individually to remove a relative misalignment difference between sensors 106 and 108.

[0044] FIG. 6 illustrates a training process 600 using unsupervised learning 602 to train a model 604 according to an embodiment. In the example of FIG. 6, a data source 606 provides training data 605 to develop the model 604 after preprocessing 608 is performed. The training data 605 in data source 606 can originate from data captured by the sensor system 104 of FIG. 1 during a training phase. The training process 600 may be performed as part of an off-line process using a separate processing system other than the perception processor 102 of FIG. 1. Alternatively, the perception processor 102 may be configured in a training phase to implement the training process 600 of FIG. 6.

[0045] In the example of FIG. 6, training data 605 includes video and LiDAR data from aligned sensors, e.g., sensors 106 and 108 of FIG. 1. Given multi-modal sensor data with no prior knowledge of the system, it is possible to register the data streams. For illustration, training process 600 is described with respect to a LiDAR-video registration example. By creating an over-constrained deep auto-encoder (DAC) definition, the DAC can be driven to capture mutual information in both the LiDAR and video data by reducing the randomness of a DAC bottleneck layer (i.e., a reduction layer) well beyond the rank at which optimal reconstruction occurs. Minimizing the reconstruction error with respect to relative shifts of the LiDAR-video data reflects that the current alignment of the sensor data has the greatest correlation possible (i.e., smallest misalignment). This method can be applied for both spatial and temporal registration.

[0046] A deep neural network auto-encoder (DNN-AE) takes an input $x \in R^d$ and first maps it to the latent representation $h \in R^{d'}$ using a deterministic function of the type $h = f_{\theta} = \sigma(Wx + b)$ with $\theta = \{W, b\}$ where W is the weight and b is the bias. This “code” is then used to reconstruct the input by a reverse mapping of $y = f_{\theta'}(h) = \sigma(W'h + b')$ with $\theta' = \{W', b'\}$. The two parameter sets are usually constrained to be of the form $W' = WT$, using the same weights for encoding the input and decoding the latent representation. Each training pattern x_i is then mapped onto its code h_i and its reconstruction y_i . The parameters are optimized, minimizing an appropriate cost function over the training set $D_n = \{(x_0, t_0), \dots, (x_n, t_n)\}$.

[0047] The first step includes using a probabilistic Restricted Boltzmann Machine (RBM) approach, trying to reconstruct noisy inputs. The training process 600 can involve the reconstruction of a clean sensor input from a partially destroyed/missing sensor. The sensor input x becomes corrupted sensor input $\bar{x}$ by adding a variable amount v of noise distributed according to the characteristics of the input data. An RBM network is trained initially with

the same number of layers as envisioned in the final DNN-AE in model 604. The parameter v represents the percentage of permissible corruption in the network. The model 604 is trained to de-noise the inputs by first finding the latent representation $h = f_{\theta}(\bar{x}) = \sigma(W\bar{x} + b)$ from which to reconstruct the original input $y = f_{\theta'}(h) = \sigma(W'h + b')$.

[0048] As part of preprocessing 608, video frame data 610 from training data 605 can be provided to a region-of-interest detector 612 that may perform edge detection or other types of region detection known in the art, similar to region-of-interest detector 212 of FIG. 2. A patch detector 614 can detect patches (i.e., areas) of interest based on the regions of interest identified by the region-of-interest detector 612 as part of preprocessing 608, similar to patch detector 214 of FIG. 2. Data fuser 616 can merge LiDAR data 618 from the training data 605 with video data 610 from selected patches of interest as detected by the patch detector 614 as part of preprocessing 608. The video frame data 610 and LiDAR data 618 fused as multiple channels for each misalignment are provided for unsupervised learning 602 of model 604. Although depicted as a deep convolutional auto-encoder (CAE), the model 604 can use a CAE or a DNN-AE, and more generally, a deep auto-encoder.

[0049] FIG. 7 illustrates a testing process 700 using the model 604 of FIG. 6 according to an embodiment. The testing process 700 can be implemented by the perception system 102 of FIG. 1. Multi-modal sensor data 702 from sensors 106 and 108 having different modalities can be fused in a live stream from sensor system 104, for instance, including video and LiDAR data. Fusing of multi-modal sensor data 702 can also include creating one or more derived input channels for data derived from sensor data, such as optical flow. A region-of-interest detector 704 analyzes the multi-modal sensor data 702 to detect any regions of interest, such as edges. The region-of-interest detector 704 can perform the same operations as the region-of-interest detector 612 of FIG. 6, for instance, using edge detection techniques known in the art.

[0050] If no regions of interest (e.g., edges) are detected by the region-of-interest detector 704, then block 706 continues processing of the multi-modal sensor data 702 as more samples/frames of data are acquired. If at least one region of interest is detected by the region-of-interest detector 704, then block 706 passes the region of interest information to a patch detector 708. The patch detector 708 can perform the same operations as the patch detector 614 of FIG. 6. For example, the patch detector 708 can detect patches (i.e., areas) of interest based on the regions of interest identified by the region-of-interest detector 704. A threshold can be applied on a percentage of pixels with edges in a given patch to reduce the data set size passed to the model 604 trained by training process 600 of FIG. 6.

[0051] At block 710, if the patch detector 708 does not detect at least one patch of interest, then processing of the multi-modal sensor data 702 continues as more samples/frames of data are acquired. If the patch detector 708 detects one or more patches of interest in the multi-modal sensor data 702, then the model 604 is applied to the one or more patches of interest using a deep convolutional neural network based on the training process 600 of FIG. 6. In the example of FIG. 7, a result of the model 604 includes a reconstruction error 712 that is provided to post-processing 714.

[0052] The post-processing 714 can include aggregating 716 the reconstruction error 712 from each patch for a given frame and analyzing 718 the reconstruction error to identify the direction of misalignment between sensors 106, 108. At block 720, if the alignment is deemed to be correct (e.g., within a threshold misalignment), then processing of the multi-modal sensor data 702 continues as more samples/frames of data are acquired. If there is a misalignment detected at block 720, then a decision making block 722 determines how to adjust either or both of the actuators 110, 112 by mapping 724 the misalignment from block 718 to positioning of the actuators 110, 112.

[0053] The testing process 500 of FIG. 5 and testing process 700 of FIG. 7 can be used to adjust the alignment of sensors 106 and 108 of FIG. 1 depending on whether supervised or unsupervised learning is performed. In order to perform occlusion edge detection, the sensors 106 and 108 must be aligned. Therefore, either testing process 500 of FIG. 5 or testing process 700 of FIG. 7 can be performed to align the sensors 106 and 108 prior to executing a training process 800 of FIG. 8 to perform occlusion edge detection to train a model 804 with supervised learning 802 according to an embodiment. In the example of FIG. 8, a data source 806 provides training data 805 to develop the model 804 after preprocessing 808 is performed. The training data 805 in data source 806 can originate from data captured by the sensor system 104 of FIG. 1 during a training phase, where the data are labeled with occlusion edges for training. The training process 800 may be performed as part of an off-line process using a separate processing system other than the perception processor 102 of FIG. 1. Alternatively, the perception processor 102 may be configured in a training phase to implement the training process 800 of FIG. 8.

[0054] In the example of FIG. 8, training data 805 includes video and labeled LiDAR data with occlusion edges. Occlusion edges are useful for a wide array of perception related tasks including object recognition, feature selection, grasping, obstacle avoidance, navigating, path-planning, localization, mapping, stereo-vision and optic flow. In addition to numerous applications, occlusions edges can be used in perception applications for figure/ground

determination. Once occlusion boundaries have been established, depth order of regions becomes possible which aids navigation, simultaneous localization and mapping (SLAM) and path planning. Occlusion edges help image feature selection by rejecting features generated from regions that span an occlusion edge. As these are dependent on viewpoint position, removing variant features saves further processing and increases recognition accuracy.

[0055] In many object recognition problems, the shape of the object is better for recognition rather than its appearance, which can be dramatically altered, e.g., by painted objects, shadows, camouflage and people wearing different clothes. However, shape determination is not the approach for SIFT-based object recognition algorithms. Furthermore, knowledge of occlusion edges helps with stereo vision and optic flow algorithms. In robotics, geometric edges of objects demarcate spatial extents helping with grasping, manipulation as well as maneuvering through the world without collision and therefore, knowledge of occlusion edges is essential.

[0056] As part of preprocessing 808, video frame data 810 from training data 805 can be provided to a region-of-interest detector 812 that may perform edge detection, occlusion edge detection, and/or other types of region detection known in the art. A patch detector 814 can detect patches (i.e., areas) of interest based on the regions of interest identified by the region-of-interest detector 812 as part of preprocessing 808, similar to patch detector 214 of FIG. 2. A labeler 816 merges labeled LiDAR data 818 from the training data 805 with video data 810 from selected patches of interest as detected by the patch detector 814 as part of preprocessing 808. For each selected patch, the labeler 816 labels patches of interest with edge or occlusion edge labels based on a dominating (i.e., most common) label for the selected patch. Labeling can simply indicate whether or not occlusion edges have been found. Once pre-training and preprocessing 808 are done, the model 804 uses the DCNN model 400 of FIG. 4 in performing supervised learning 802 with respect to a ground truth.

[0057] FIG. 9 illustrates a testing process 900 using the model 804 of FIG. 8 according to an embodiment. Multi-modal sensor data 902 from data source 901 (e.g., previously acquired data) having different modalities can be fused with a live stream of sensor data from either or both sensors 106 and 108 of sensor system 104 of FIG. 1 including, for instance, LiDAR and/or video data. Fusing of multi-modal sensor data 902 can also include creating one or more derived input channels for data derived from sensor data, such as optical flow. A region-of-interest detector 904 analyzes the multi-modal sensor data 902 to detect any regions of interest such as edges and/or occlusion edges. The region-of-interest

detector 904 can perform similar operations as the region-of-interest detector 812 of FIG. 8, for instance, using edge detection and occlusion edge techniques known in the art.

[0058] If no regions of interest (e.g., edges and/or occlusion edges) are detected by the region-of-interest detector 904, then block 906 continues processing of the multi-modal sensor data 902 as more samples/frames of data are acquired. If at least one region of interest is detected by the region-of-interest detector 904, then block 906 passes the region of interest information to a patch detector 907. The patch detector 907 can perform similar operations as the patch detector 814 of FIG. 8. For example, the patch detector 907 can detect patches (i.e., areas) of interest based on the regions of interest identified by the region-of-interest detector 904. Patches containing edges identified using data from video/image frames can be analyzed further using LiDAR data for depth perception to assign a confidence of the existence of edges and/or occlusion edges in patches of interest.

[0059] At block 908, if the patch detector 907 does not detect at least one patch of interest, then processing of the multi-modal sensor data 902 continues as more samples/frames of data are acquired. If the patch detector 907 detects one or more patches of interest in the multi-modal sensor data 902, then a check can be performed at block 909 to determine whether there are multiple sensors of different modalities available. For example, one or more of the sensors 106 and 108 may not be available due to damage or a fault condition. If multiple sensors are determined to be available at block 909, then at block 910 an alignment check is performed to determine whether sensors 106 and 108 are presently aligned or misaligned. If the sensors misaligned, then at block 911 a portion of process 500 of FIG. 5 (supervised) or process 700 of FIG. 7 (unsupervised) can be performed to correct sensor alignment prior to applying model 804. If at block 909 multiple sensors are unavailable or at block 910 the sensors are aligned, then model 804 is applied. The model 804 is applied to the one or more patches of interest using a deep convolutional neural network based on the training process 800 of FIG. 8. In the example of FIG. 9, a result of the model 804 includes classification data 912 (e.g., occlusion edge / no occlusion edge classification per patch) that are provided to post-processing 914.

[0060] The post-processing 914 can include aggregating 916 the classification data 912 from each patch as a classification result and smoothing 918 the classification result, for instance, using a Gaussian filter and/or temporal information, resulting in a post-processing result. A visualization / decision making block 920 outputs perception indication 922 that includes identification of occlusion edges. For example, the perception indication 922 can

include occlusion edge detection information used by a decision making process, such as autonomous navigation of the vehicle 100 of FIG. 1.

[0061] The terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting of the embodiments. While the present disclosure has been described in detail in connection with only a limited number of embodiments, it should be readily understood that the present disclosure is not limited to such disclosed embodiments. Rather, the present disclosure can be modified to incorporate any number of variations, alterations, substitutions or equivalent arrangements not heretofore described, but which are commensurate with the spirit and scope of the present disclosure. Additionally, while various embodiments of the present disclosure have been described, it is to be understood that aspects of the present disclosure may include only some of the described embodiments. Accordingly, the present disclosure is not to be seen as limited by the foregoing description, but is only limited by the scope of the appended claims.

CLAIMS

What is claimed is:

1. A method comprising:
fusing multi-modal sensor data from a plurality of sensors having different modalities;
detecting at least one region of interest in the multi-modal sensor data;
detecting one or more patches of interest in the multi-modal sensor data based on detecting the at least one region of interest;
applying a model comprising a deep convolutional neural network to the one or more patches of interest;
performing post-processing of a result of applying the model to produce a post-processing result for the one or more patches of interest; and
outputting a perception indication of the post-processing result.
2. The method of claim 1, wherein detecting the at least one region of interest comprises performing edge detection on one or more images in the multi-modal sensor data.
3. The method of claim 2, wherein the edge detection comprises occlusion edge detection.
4. The method of claim 1, wherein detecting the one or more patches of interest comprises applying a threshold on a percentage of pixels with edges in a given patch.
5. The method of claim 1, wherein the model is trained using a supervised learning process for auto-registration of the multi-modal sensor data and outputs classification values.
6. The method of claim 5, wherein the post-processing comprises aggregating the classification values and identifying a dominating label representing an alignment offset between at least two of the sensors to determine whether a misalignment exists.
7. The method of claim 1, wherein the model is trained using an unsupervised learning process using a deep auto-encoder and outputs reconstruction errors.
8. The method of claim 7, wherein the post-processing comprises aggregating the reconstruction errors and identifying a misalignment between at least two of the sensors based on the aggregation of the reconstruction errors.
9. The method of claims 6 or 8, wherein the perception indication is a misalignment indicator used to adjust at least one actuator of the at least two sensors having the misalignment.

10. The method of claim 1, wherein the model is trained using a supervised learning process for edge and occlusion edge labeling of the multi-modal sensor data and outputs classification values.

11. The method of claim 10, wherein the post-processing comprises aggregating the classification values and smoothing the classification, and further wherein the perception indication comprises identification of occlusion edges.

12. A system comprising:
a sensor system comprising a plurality of sensors having different modalities; and
a perception processor operable to fuse multi-modal sensor data from the sensors, detect at least one region of interest in the multi-modal sensor data, detect one or more patches of interest in the multi-modal sensor data based on detecting the at least one region of interest, apply a model comprising a deep convolutional neural network to at least one detected patch of interest, perform post-processing of a result of applying the model to produce a post-processing result for the at least one detected patch of interest, and output a perception indication of the post-processing result.

13. The system of claim 12, wherein the at least one region of interest is detected by performing edge detection on one or more images in the multi-modal sensor data.

14. The system of claim 12, wherein detection of the one or more patches of interest comprises applying a threshold on a percentage of pixels with edges in a given patch.

15. The system of claim 12, wherein the model is trained using a supervised learning process for auto-registration of the multi-modal sensor data and outputs classification values.

16. The system of claim 15, wherein the post-processing comprises aggregating the classification values and identifying a dominating label representing an alignment offset between at least two of the sensors to determine whether a misalignment exists.

17. The system of claim 12, wherein the model is trained using an unsupervised learning process using a deep auto-encoder and outputs reconstruction errors.

18. The system of claim 17, wherein the post-processing comprises aggregating the reconstruction errors and identifying a misalignment between at least two of the sensors based on the aggregation of the reconstruction errors.

19. The system of claims 16 or 18, wherein the perception indication is a misalignment indicator used to adjust at least one actuator of the at least two sensors having the misalignment.

20. The system of claim 12, wherein the model is trained using a supervised learning process for edge and occlusion edge labeling of the multi-modal sensor data and outputs classification values, the post-processing comprises aggregating the classification values and smoothing the classification, and further wherein the perception indication comprises identification of occlusion edges.

1/9

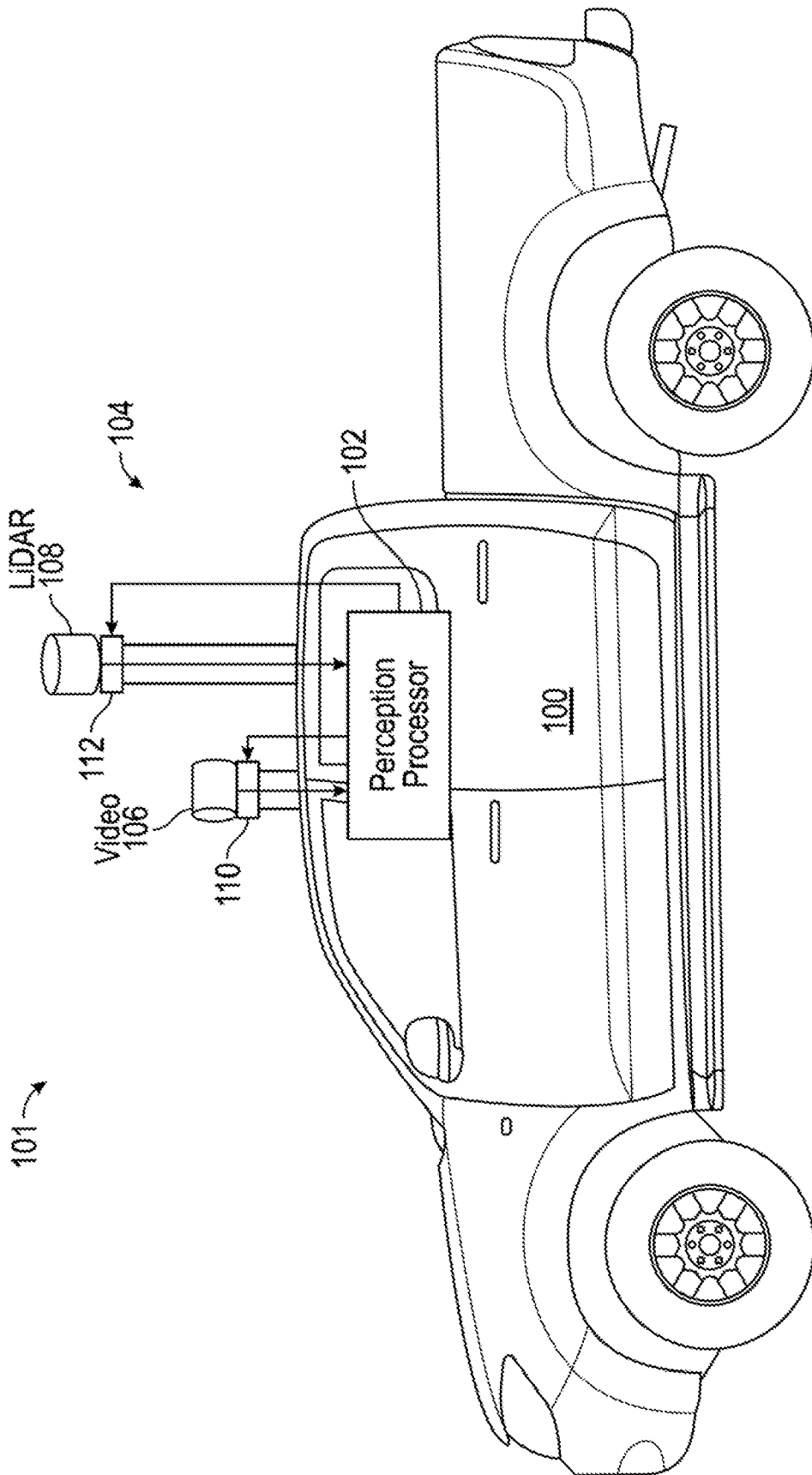


FIG. 1

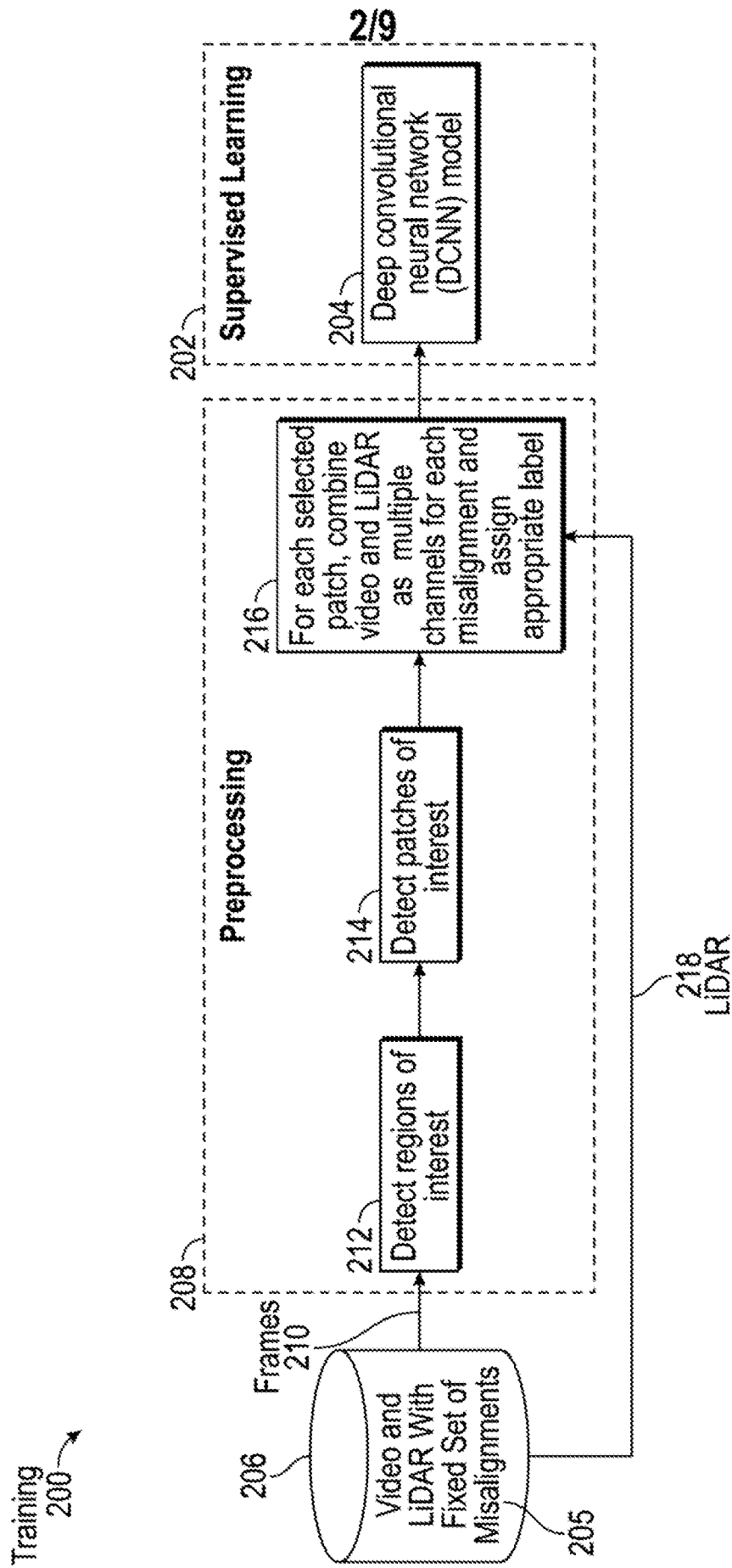


FIG. 2

3/9

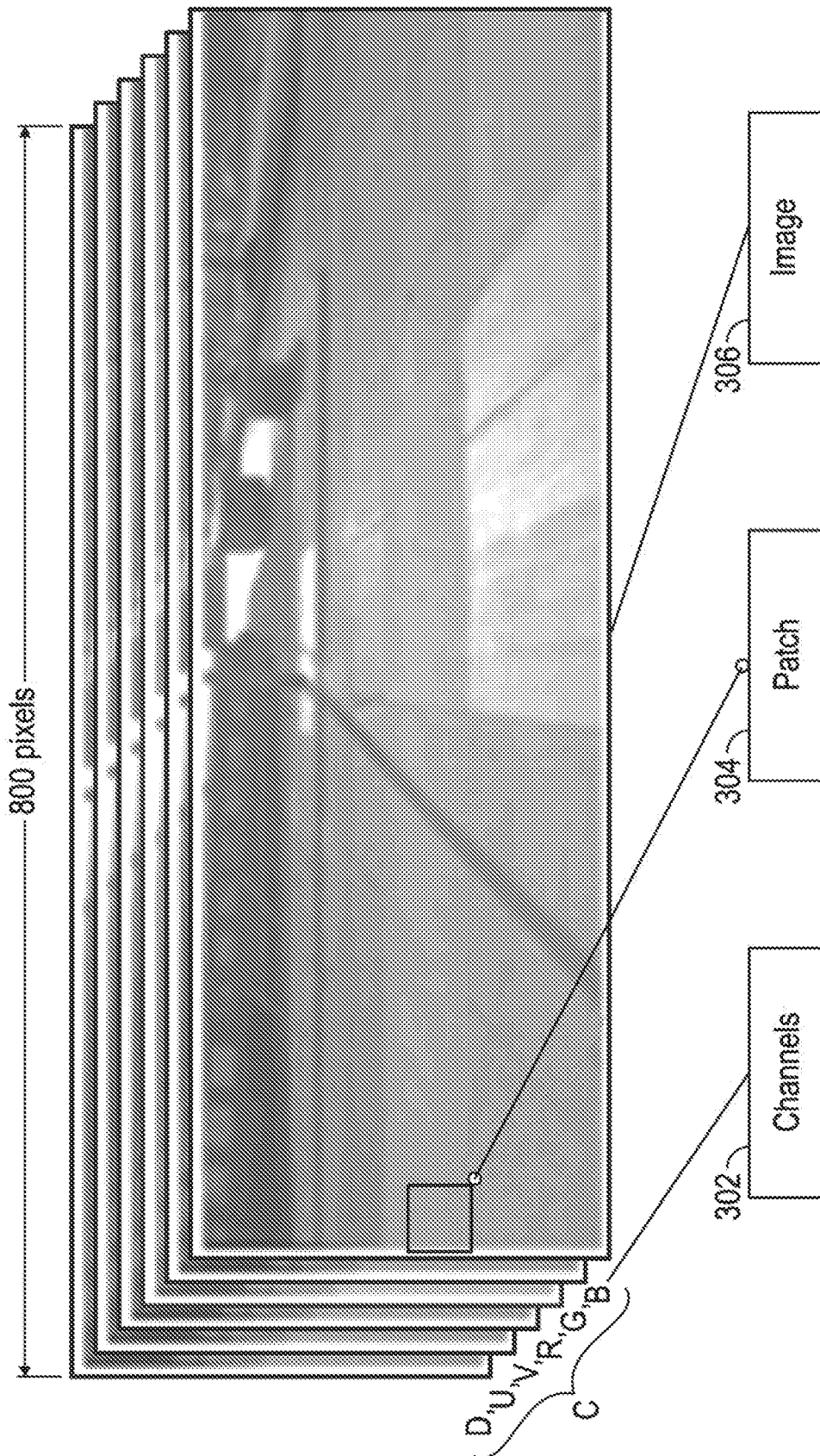


FIG. 3

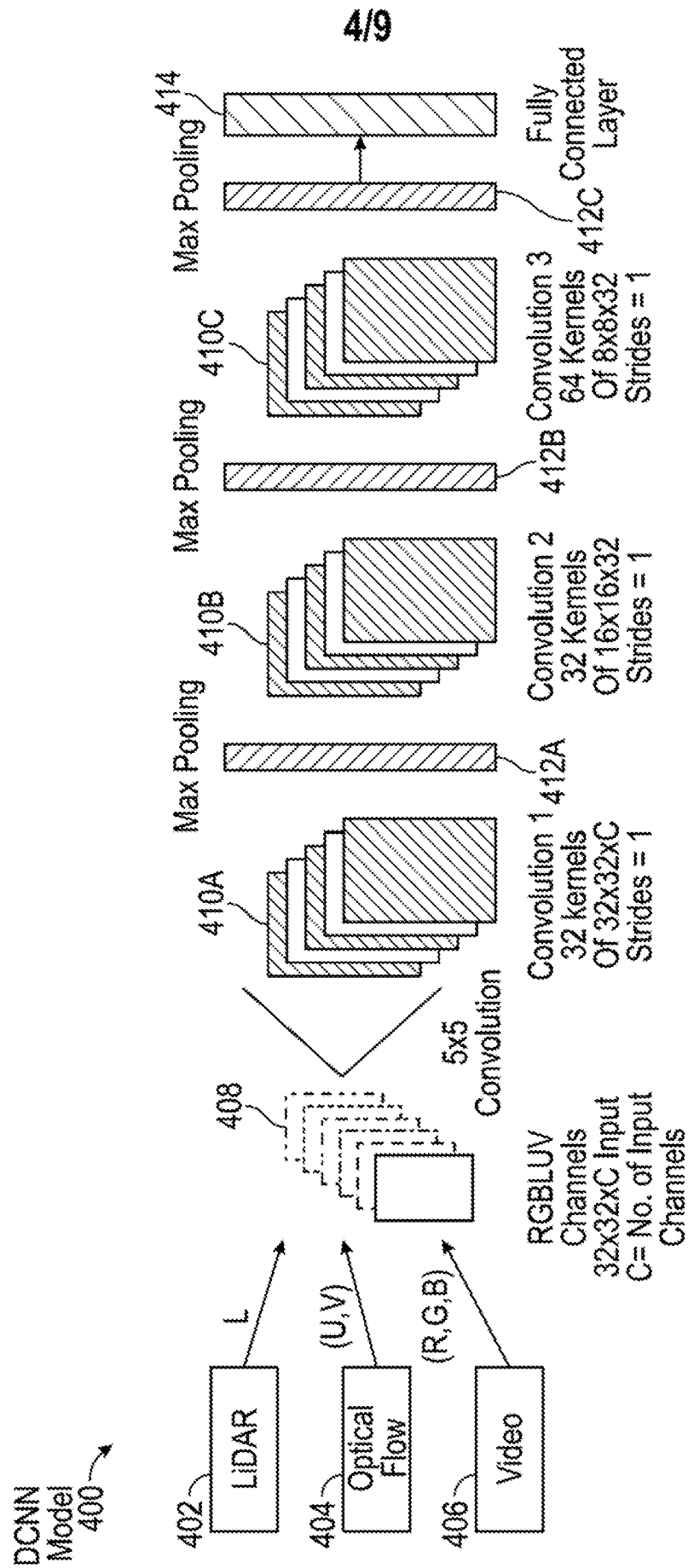
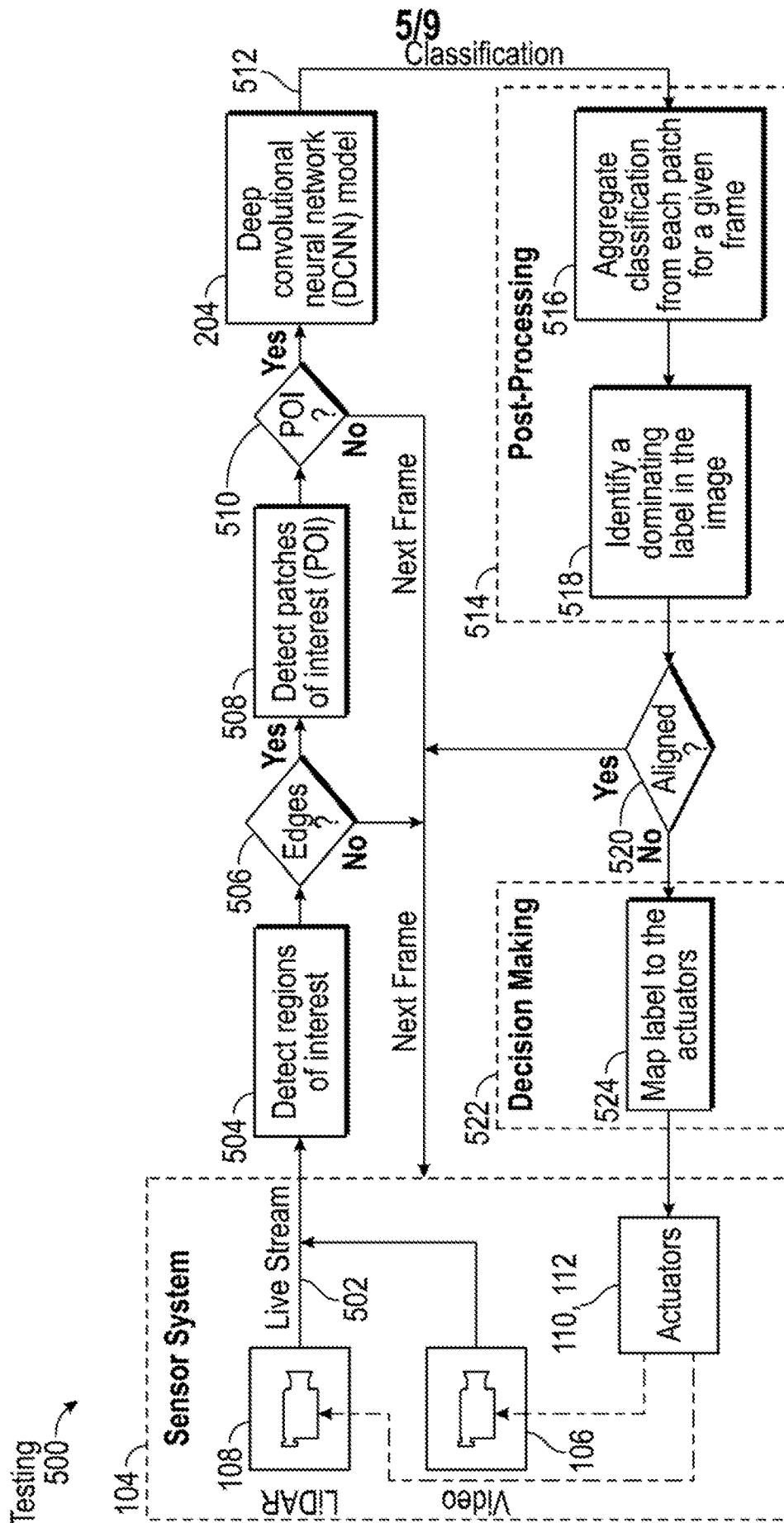


FIG. 4



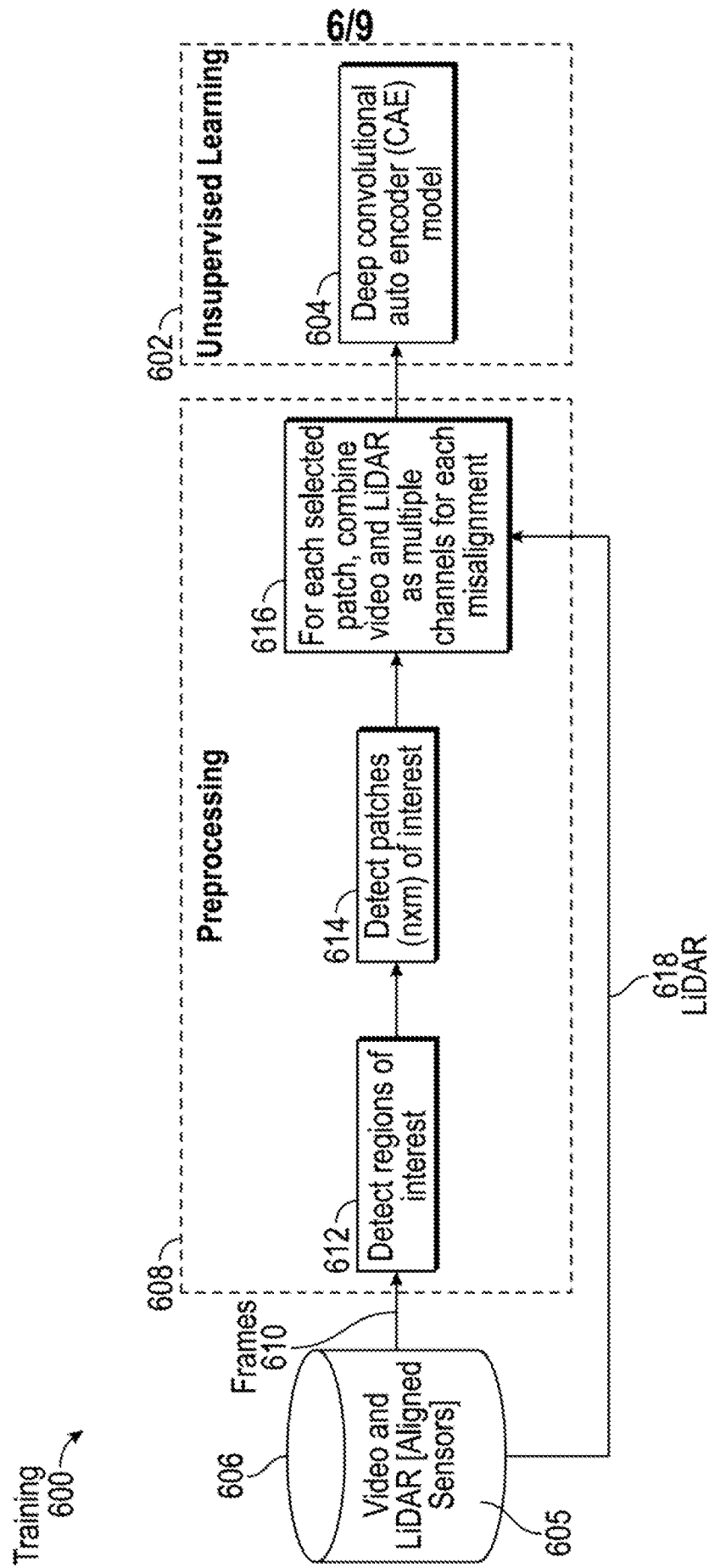


FIG. 6

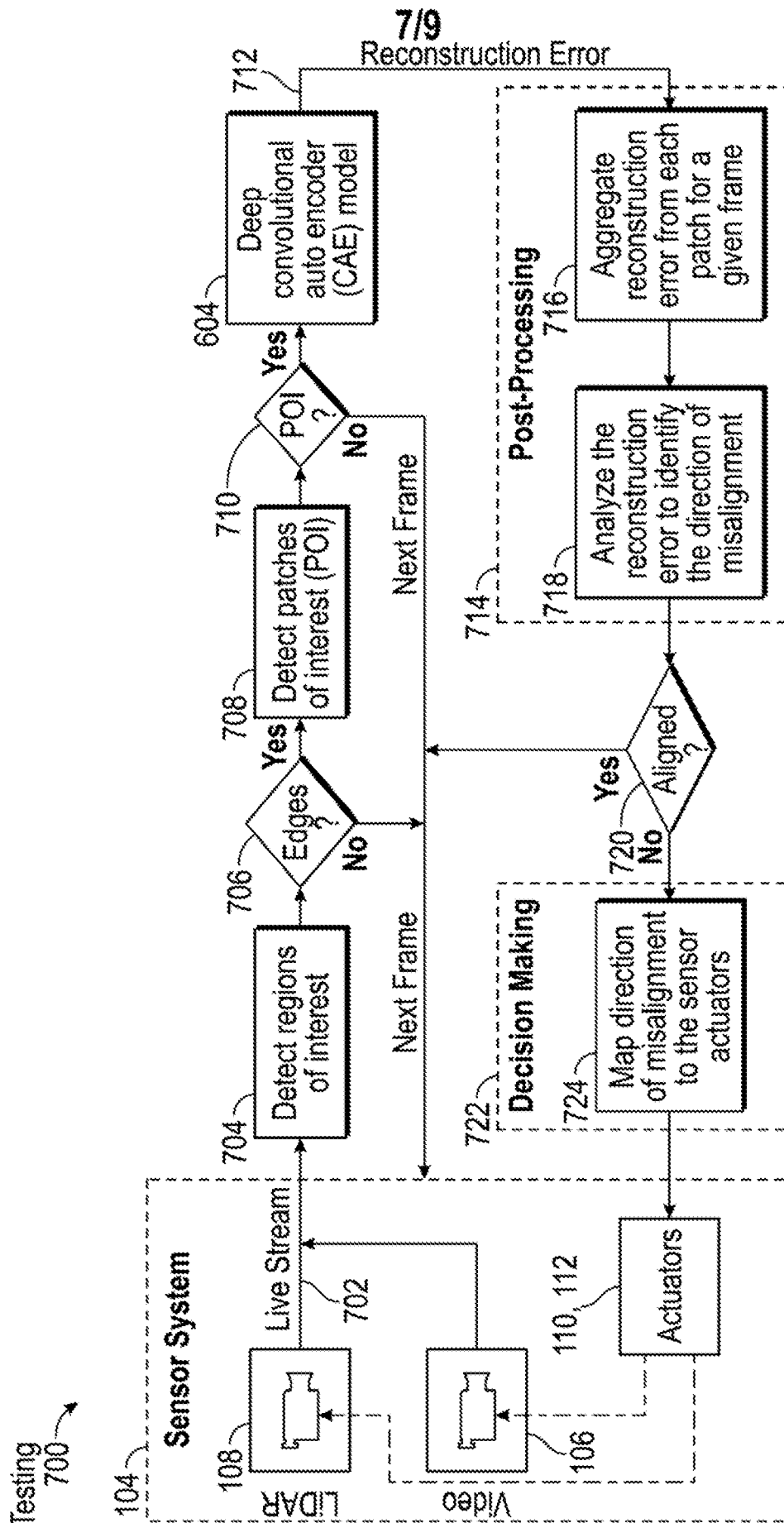


FIG. 7

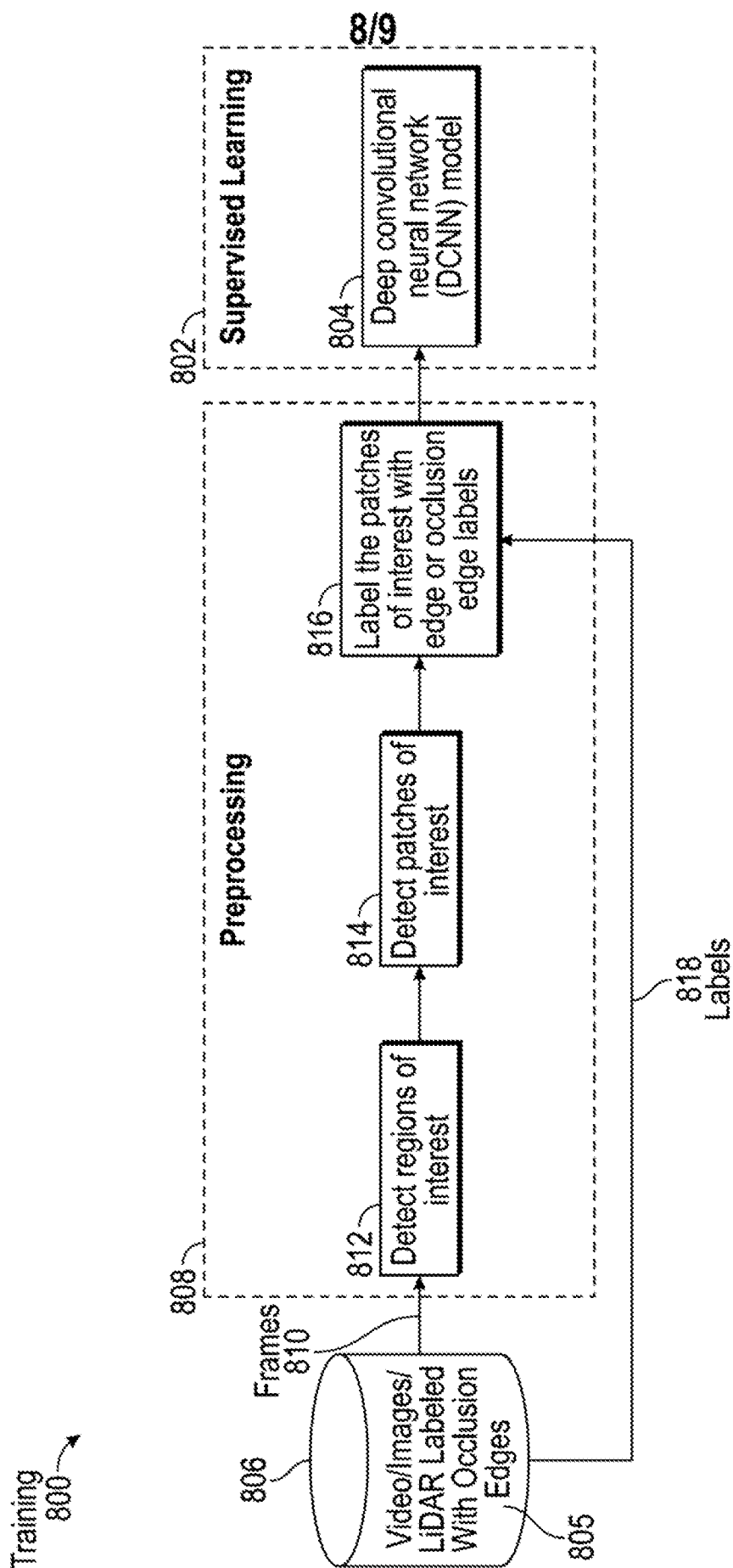


FIG. 8

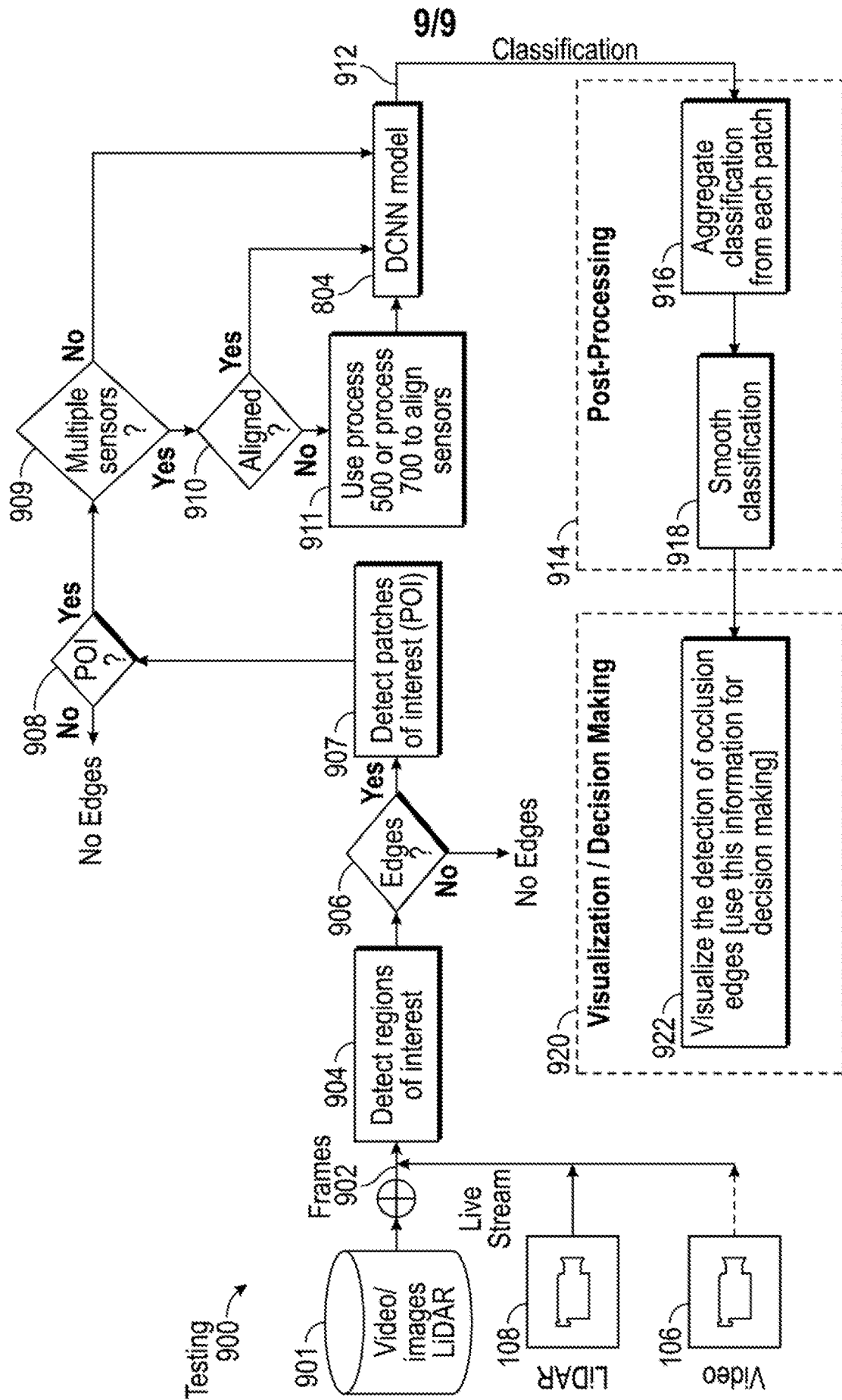


FIG. 9

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2015/066664

A. CLASSIFICATION OF SUBJECT MATTER

INV. G06K9/00

ADD. G06N3/04

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06N G06K

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	Z. KIRA ET AL: "Long-range pedestrian detection using stereo and a cascade of convolutional network classifiers", PROCEEDINGS OF THE 2012 IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS (IROS'12), 7 October 2012 (2012-10-07), pages 2396-2403, XP032287875, DOI: 10.1109/IROS.2012.6386029 sections III and V ----- -/--	1-20



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

8 April 2016

Date of mailing of the international search report

26/04/2016

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Douarche, Nicolas

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2015/066664

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	M. SZARVAS ET AL: "Real-time pedestrian detection using LIDAR and convolutional neural networks", PROCEEDINGS OF THE 2006 IEEE INTELLIGENT VEHICLES SYMPOSIUM (IVS'06), 13 June 2006 (2006-06-13), pages 213-218, XP010937015, DOI: 10.1109/IVS.2006.1689630 sections I-III -----	1-20
X	C. PREMEBIDA ET AL: "LIDAR and vision-based pedestrian detection system", JOURNAL OF FIELD ROBOTICS, vol. 26, no. 9, 1 September 2009 (2009-09-01), pages 696-711, XP055263851, DOI: 10.1002/rob.20312 figure 1 and table I sections 3-5 -----	1-20
X	J. NGIAM ET AL: "Multimodal deep learning", PROCEEDINGS OF THE 28TH INTERNATIONAL CONFERENCE IN MACHINE LEARNING (ICML'11), 28 June 2011 (2011-06-28), pages 689-696, XP055128071, ISBN: 978-1-4503-0619-5 sections 3 and 4 -----	1-20
A	B. WU ET AL: "Fast pedestrian detection with laser and image data fusion", PROCEEDINGS OF THE 6TH INTERNATIONAL CONFERENCE ON IMAGE AND GRAPHICS (ICIG'11), 12 August 2011 (2011-08-12), pages 605-608, XP032048726, DOI: 10.1109/ICIG.2011.107 sections I and II -----	1-20