US 20230053785A1

(19) **United States**

(12) **Patent Application Publication**

(10) **Pub. No.: US 2023/0053785 A1**

(43) **Pub. Date: Feb. 23, 2023**

**Carvalho et al.**

(54) **VISION-BASED MACHINE LEARNING MODEL FOR AGGREGATION OF STATIC OBJECTS AND SYSTEMS FOR AUTONOMOUS DRIVING**

(71) Applicant: **Tesla, Inc.**, Austin, TX (US)

(72) Inventors: **Micael Carvalho**, Austin, TX (US); **John Emmons**, Austin, TX (US); **Patrick Cho**, Austin, TX (US); **Bradley Emi**, Austin, TX (US); **Saachi Jain**, Austin, TX (US); **Nishant Desai**, Austin, TX (US); **Tony Duan**, Austin, TX (US)

(21) Appl. No.: **17/820,849**

(22) Filed: **Aug. 18, 2022**

**Related U.S. Application Data**

(60) Provisional application No. 63/260,439, filed on Aug. 19, 2021, provisional application No. 63/287,936, filed on Dec. 9, 2021.

**Publication Classification**

(51) **Int. Cl.**

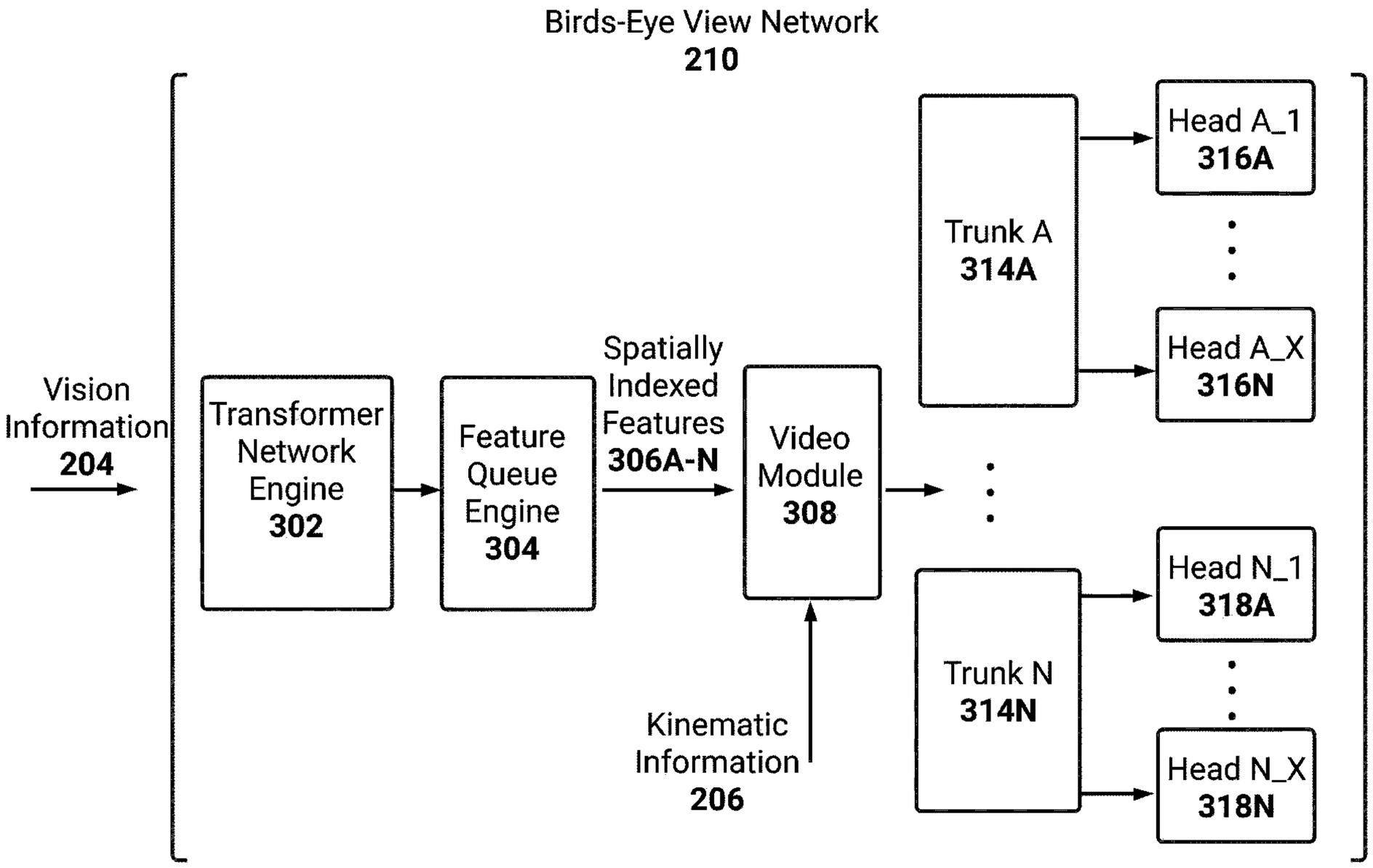
| | |
|---|---|
| ***B60W 60/00*** | (2006.01) |
| ***G06V 20/56*** | (2006.01) |

(52) **U.S. Cl.**

CPC ..... ***B60W 60/001*** (2020.02); ***G06V 20/588*** (2022.01); *B60W 2420/42* (2013.01); *B60W 2552/53* (2020.02)

(57) **ABSTRACT**

Systems and methods for a vision-based machine learning model for aggregation of static objects and systems for autonomous driving. An example method includes obtaining images from image sensors positioned about a vehicle. Features associated with the images are determined, with the features being output based on a forward pass through a machine learning model. The features are projected into a vector space associated with a birds-eye view based on the machine learning model. The projected features are aggregated with other projected features associated with prior images. Images depicting static objects in the birds-eye view are output.



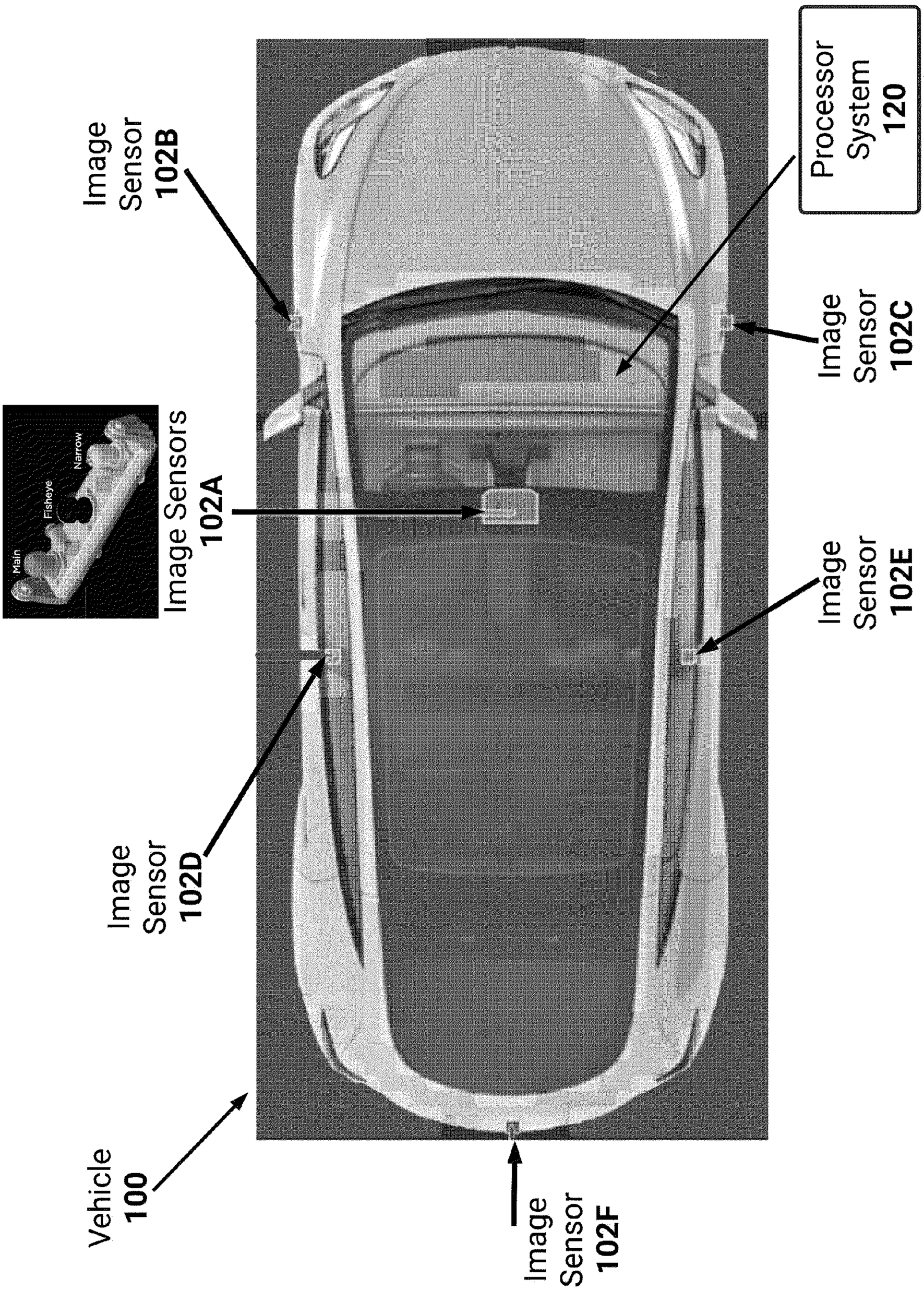


Image Sensor 102B
Processor System 120
Main
Fisheye
Narrow
Image Sensors 102A
Image Sensor 102C
Image Sensor 102D
Image Sensor 102E
Vehicle 100
Image Sensor 102F

FIG. 1A

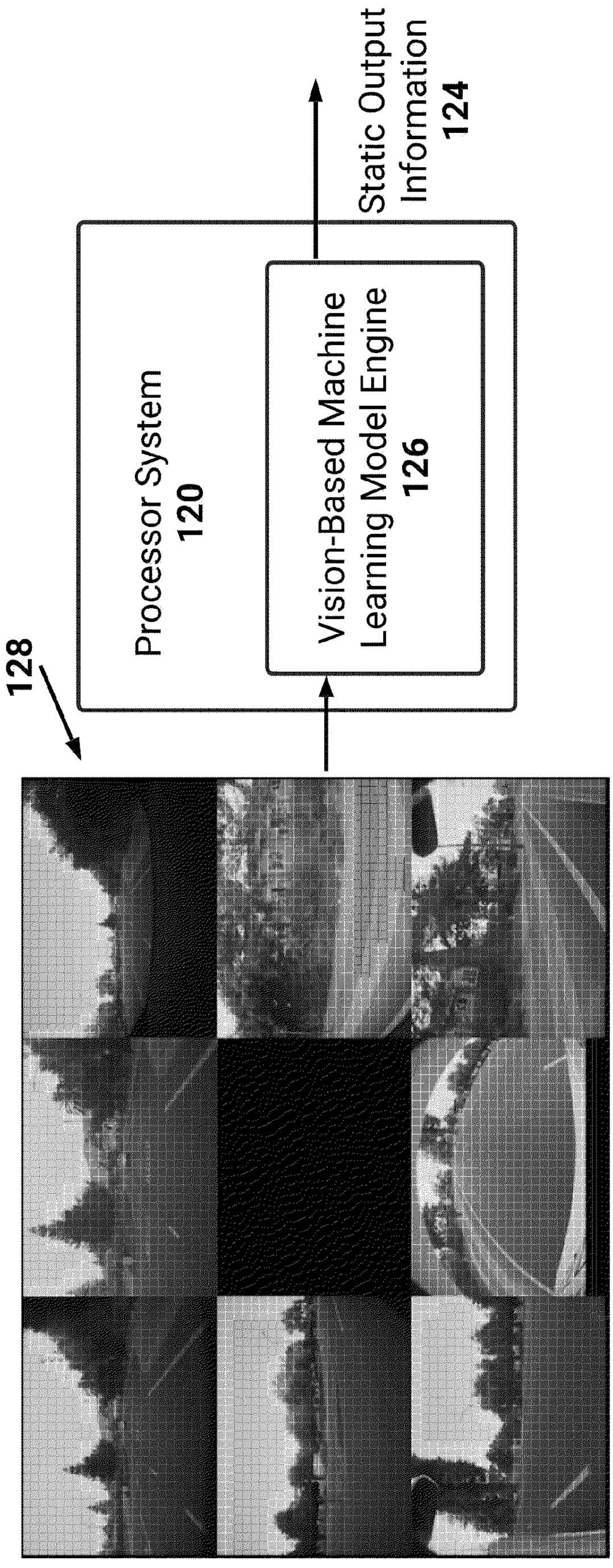


Image Information
122
128
Processor System
120
Vision-Based Machine
Learning Model Engine
126
Static Output
Information
124

FIG. 1B

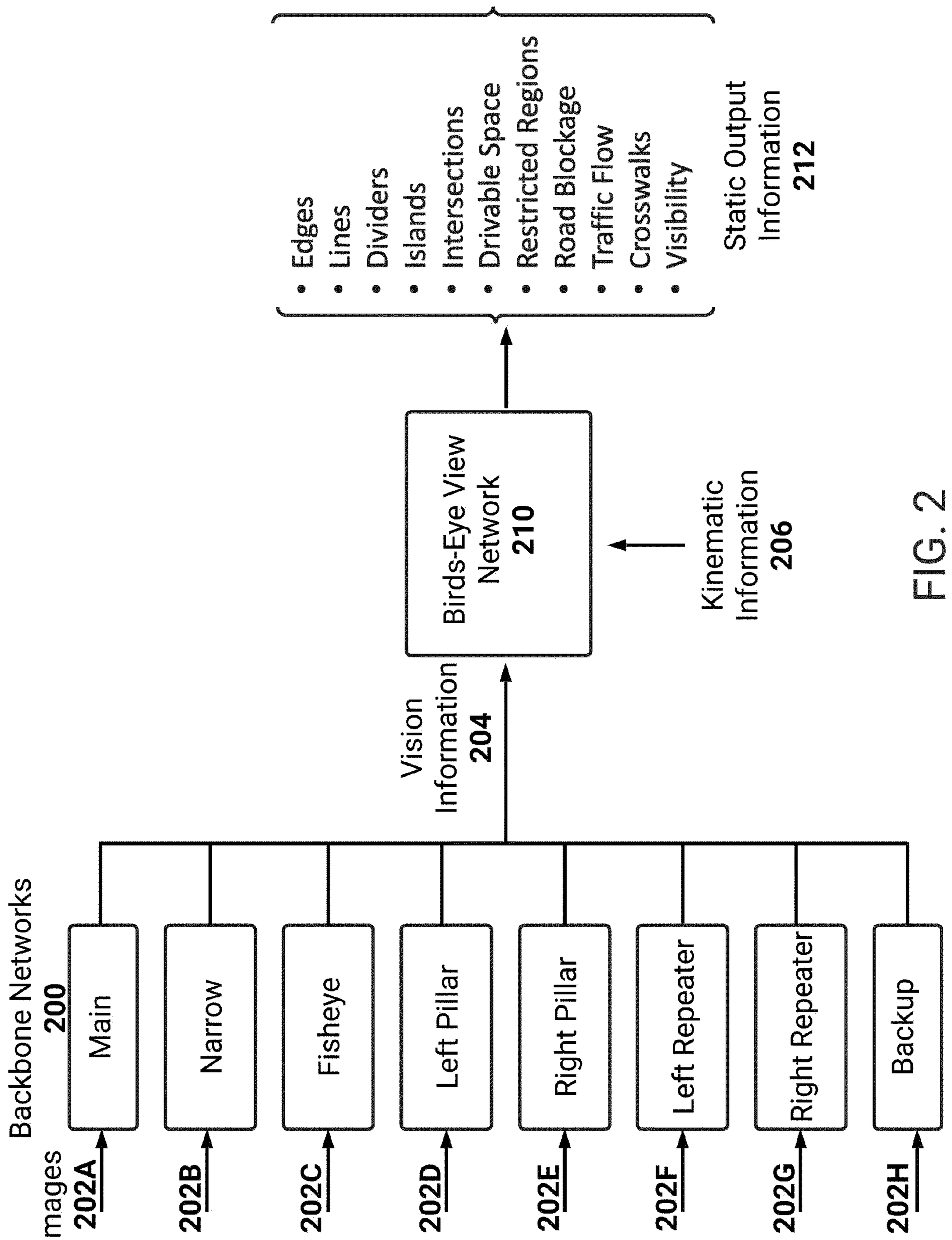

Backbone Networks
200
Images
202A
Main
202B
Narrow
202C
Fisheye
202D
Left Pillar
202E
Right Pillar
202F
Left Repeater
202G
Right Repeater
202H
Backup
Vision Information
204
Kinematic Information
206
Birds-Eye View Network
210
Static Output Information
212
Edges
Lines
Dividers
Islands
Intersections
Drivable Space
Restricted Regions
Road Blockage
Traffic Flow
Crosswalks
Visibility

FIG. 2

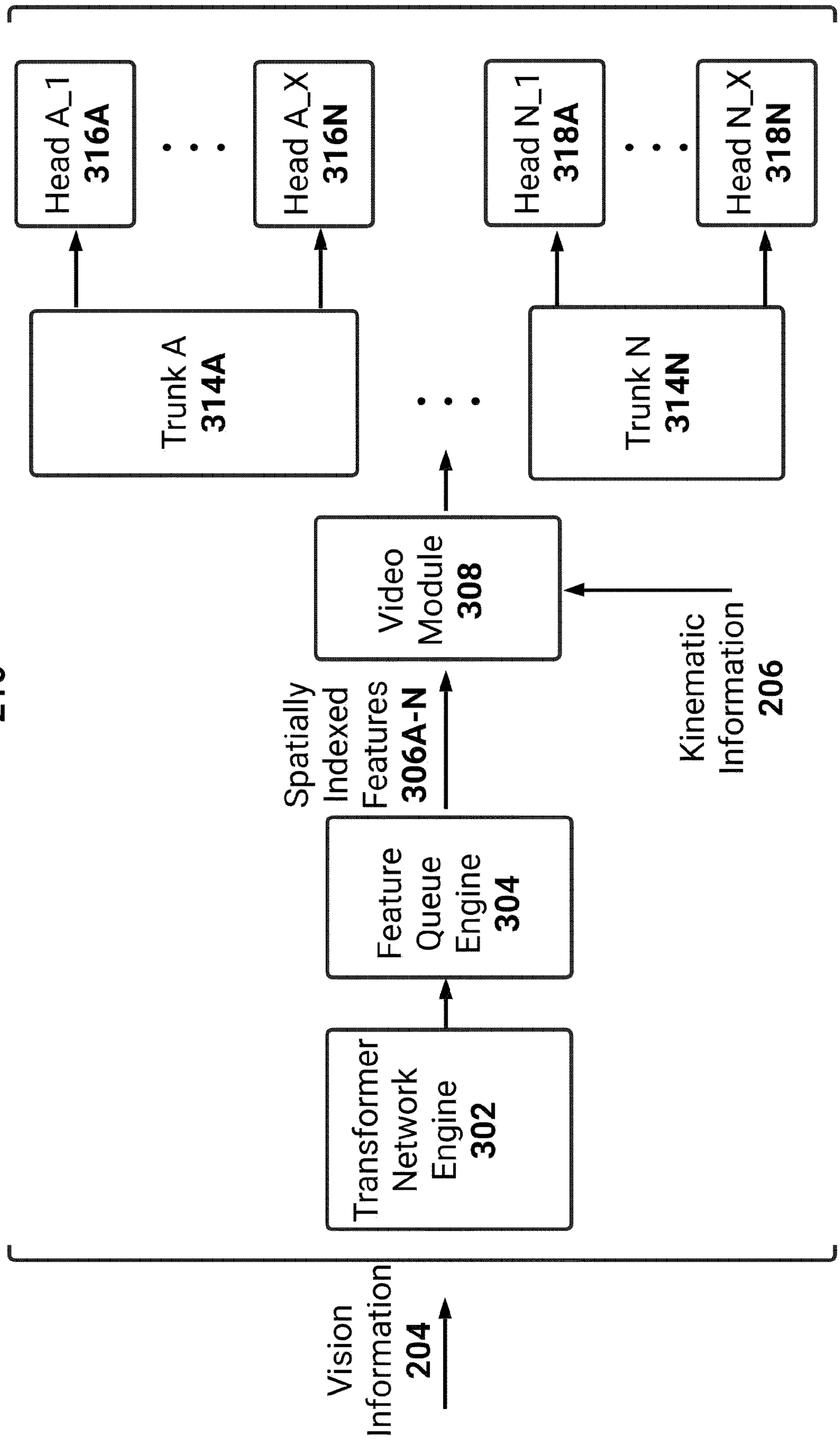

Birds-Eye View Network
210
Transformer Network Engine
302
Feature Queue Engine
304
Spatially Indexed Features
306A-N
Video Module
308
Kinematic Information
206
Trunk A
314A
Trunk N
314N
Head A_1
316A
Head A_X
316N
Head N_1
318A
Head N_X
318N
Vision Information
204

FIG. 3A

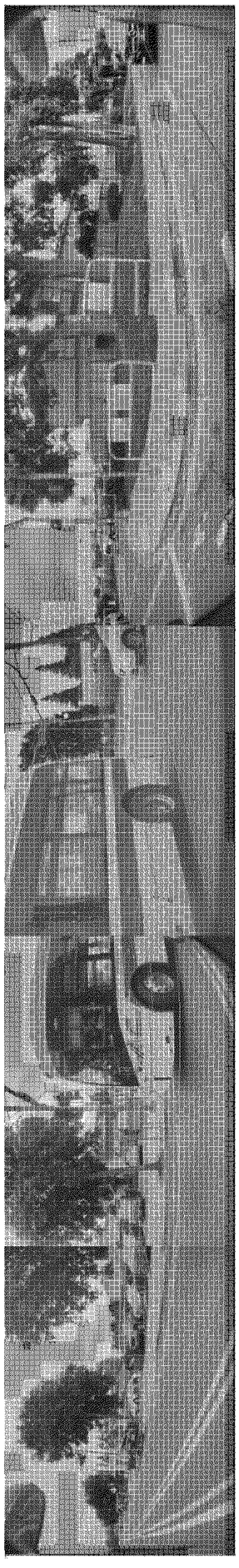


Image Information
320

322

Birds-Eye View Network
210

Output Information
324

Planning / Control
330

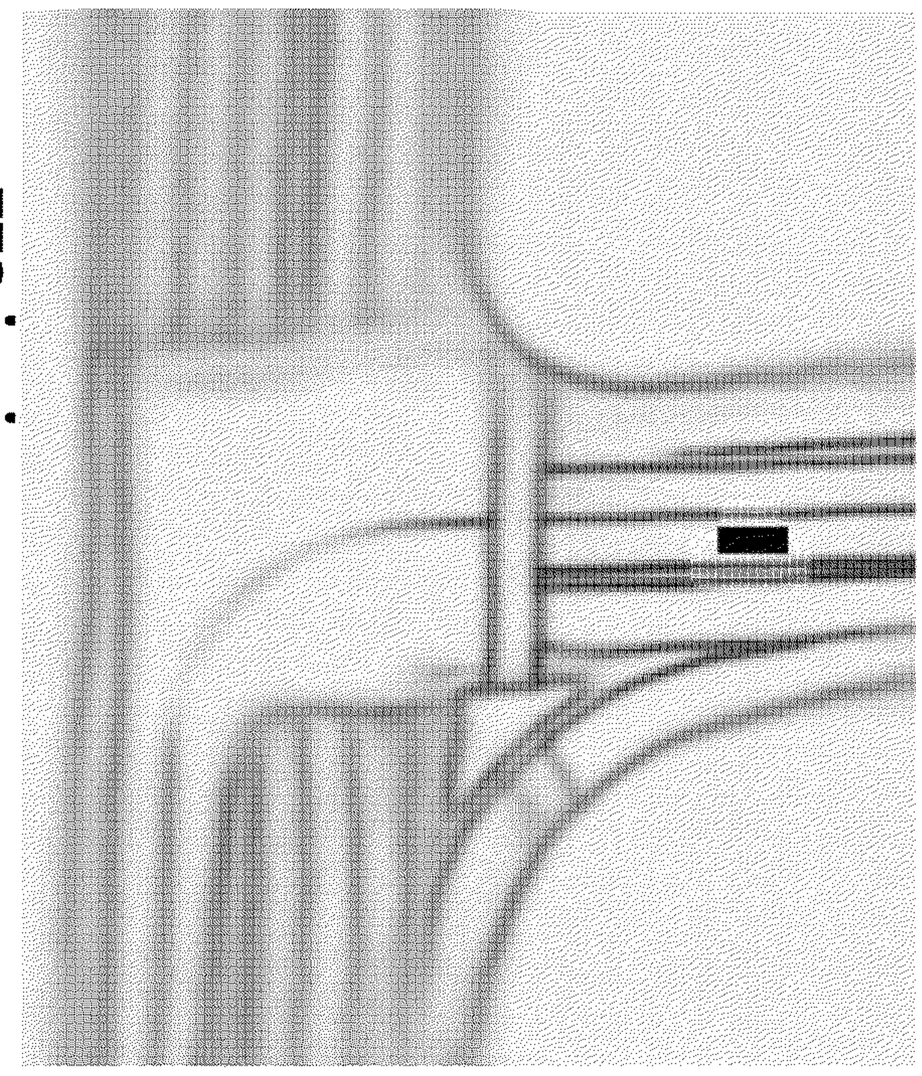


FIG. 3B

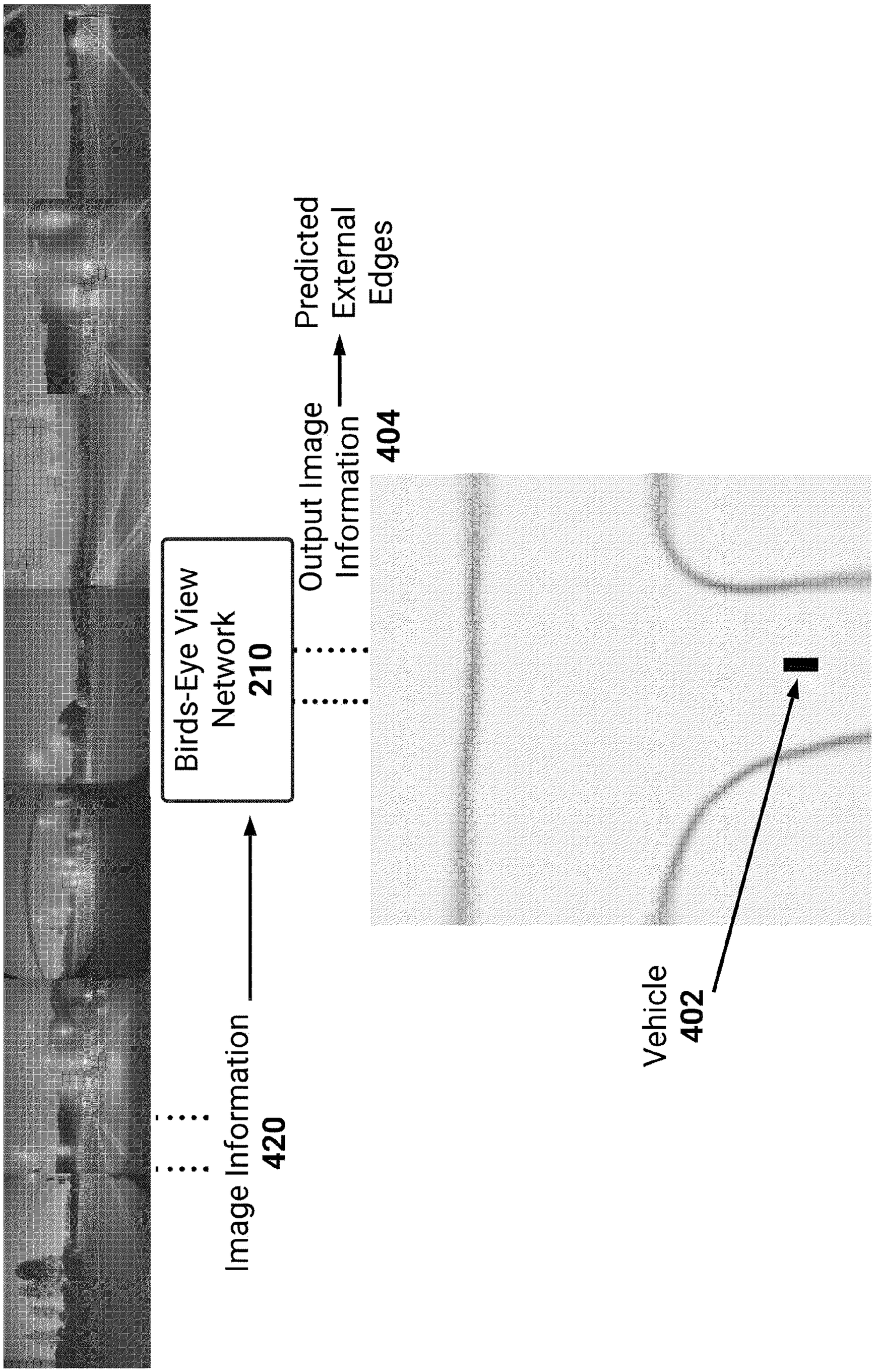


Image Information
420
Bird's-Eye View Network
210
Output Image Information
Predicted External Edges
404
Vehicle
402

FIG. 4A

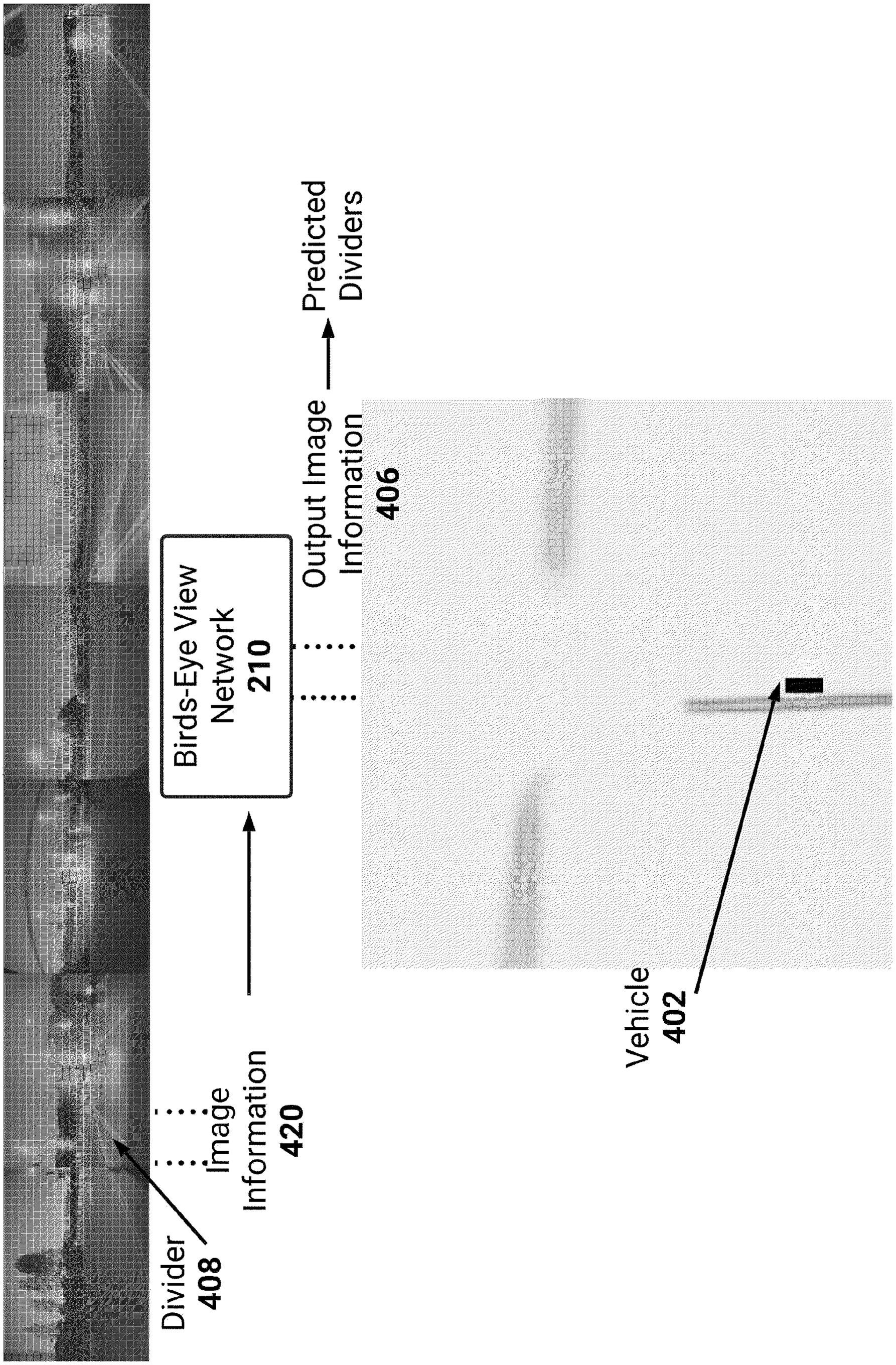


Image Information
404
Bird's-Eye View Network
210
Output Image Information
406
Predicted Dividers
Vehicle
402
Divider
408
Image Information
420

FIG. 4B

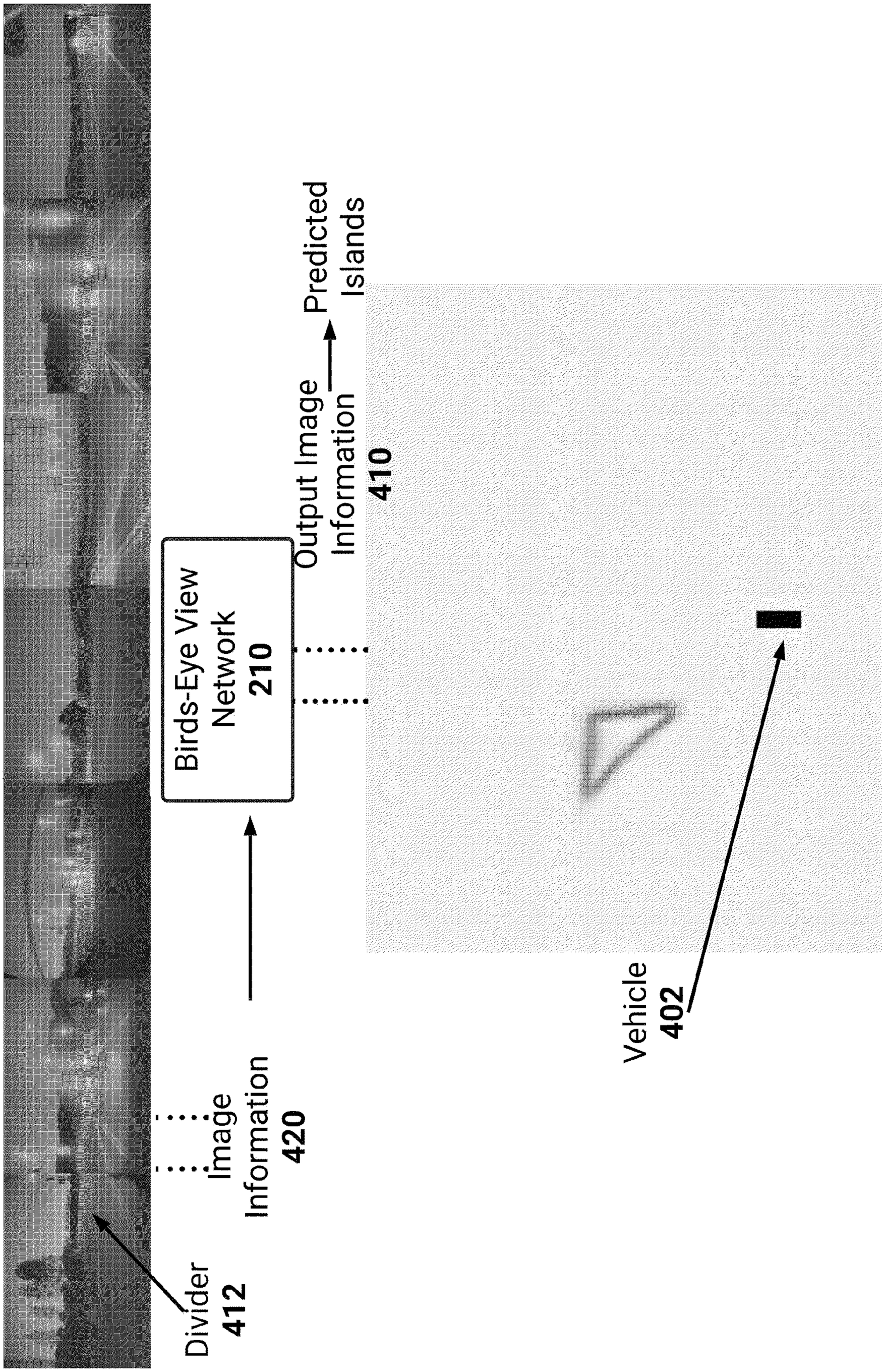

Bird's-Eye View Network
210
Output Image Information
Predicted Islands
410
Vehicle
402
Divider
412
Image Information
420

FIG. 4C

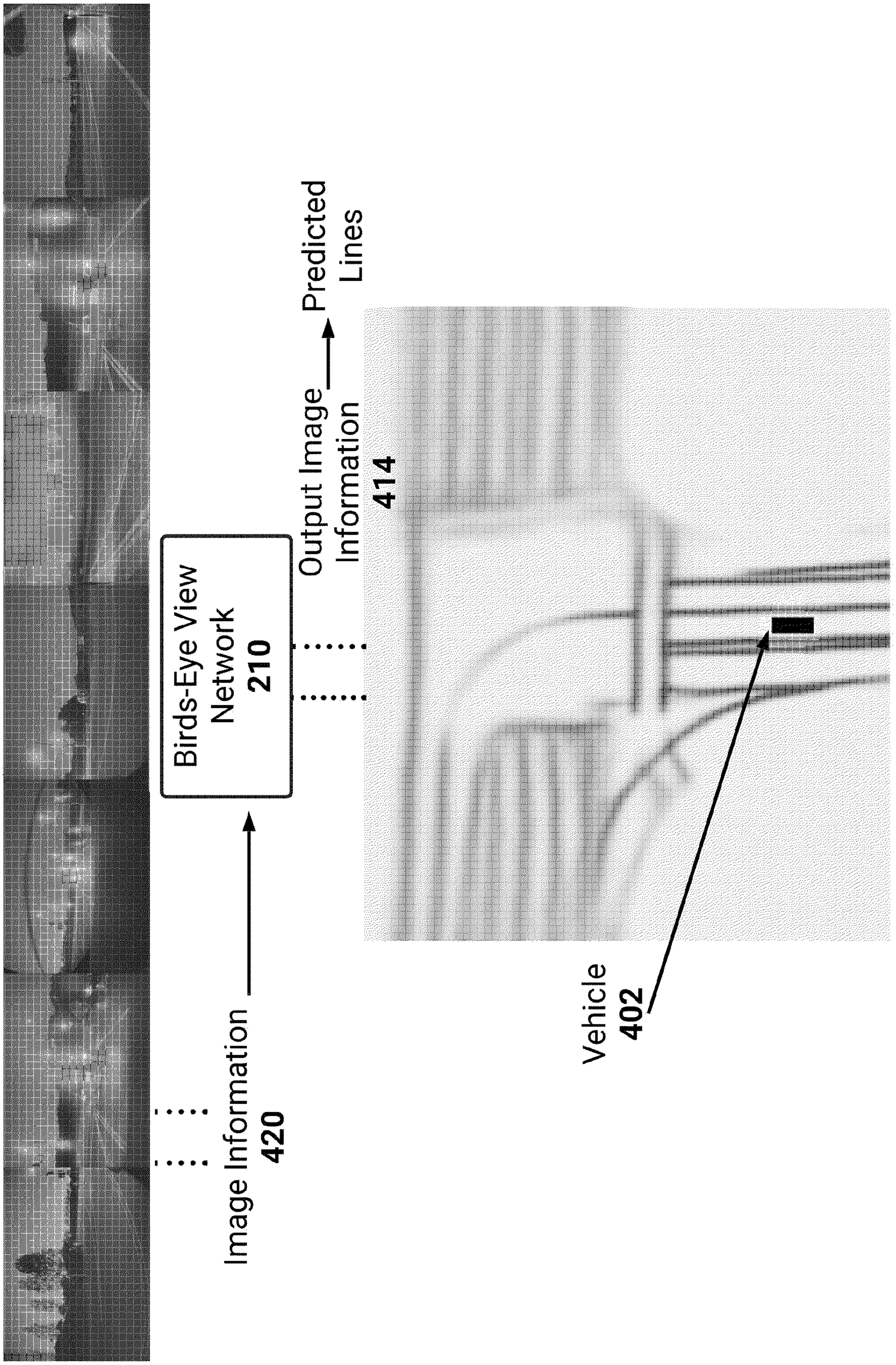


Image Information
420
Bird's-Eye View Network
210
Output Image Information
Predicted Lines
414
Vehicle
402

FIG. 4D

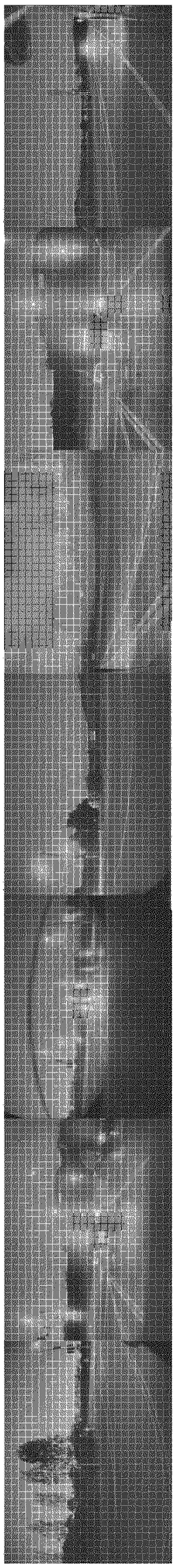


Image Information
420

Bird's-Eye View Network
210

Output Image Information

Superpixel Decoded Cores

416

Vehicle
402

FIG. 4E

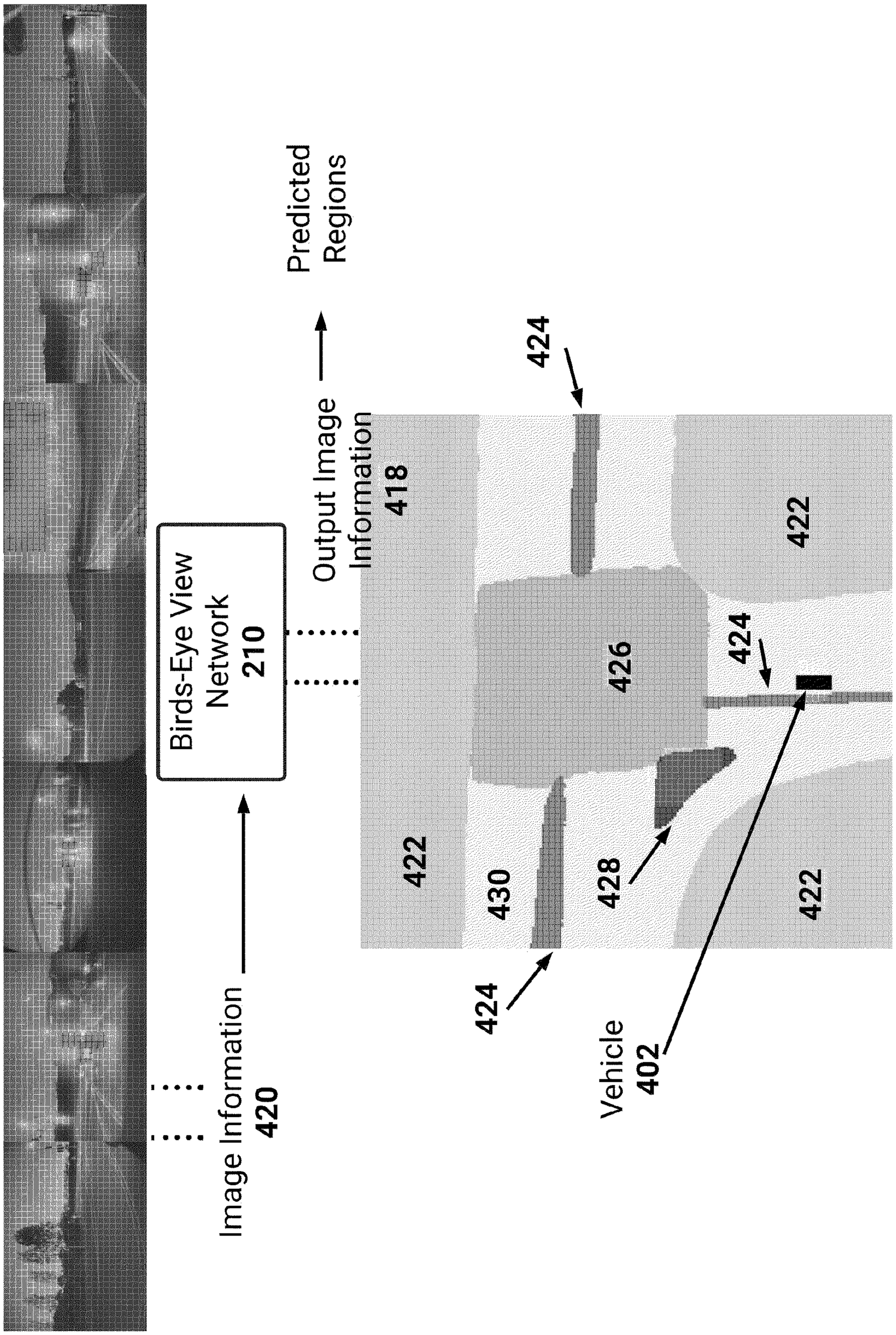

Predicted Regions
Output Image Information 418
Birds-Eye View Network 210
Image Information 420
424
422
426
424
422
430
428
422
424
Vehicle 402

FIG. 4F

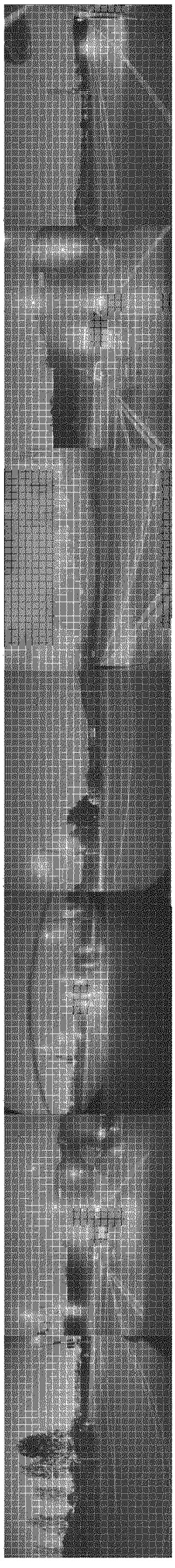
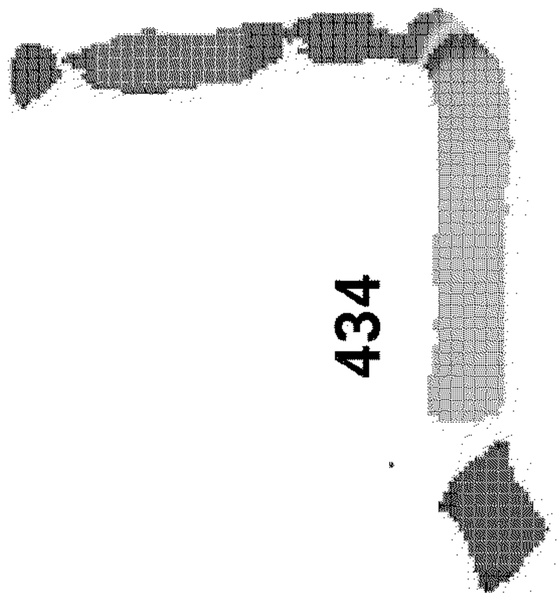


Image Information
**420**

Bird's-Eye View Network
**210**

Output Image Information
**432**

Predicted Crosswalk Region



**434**

Vehicle
**402**

FIG. 4G

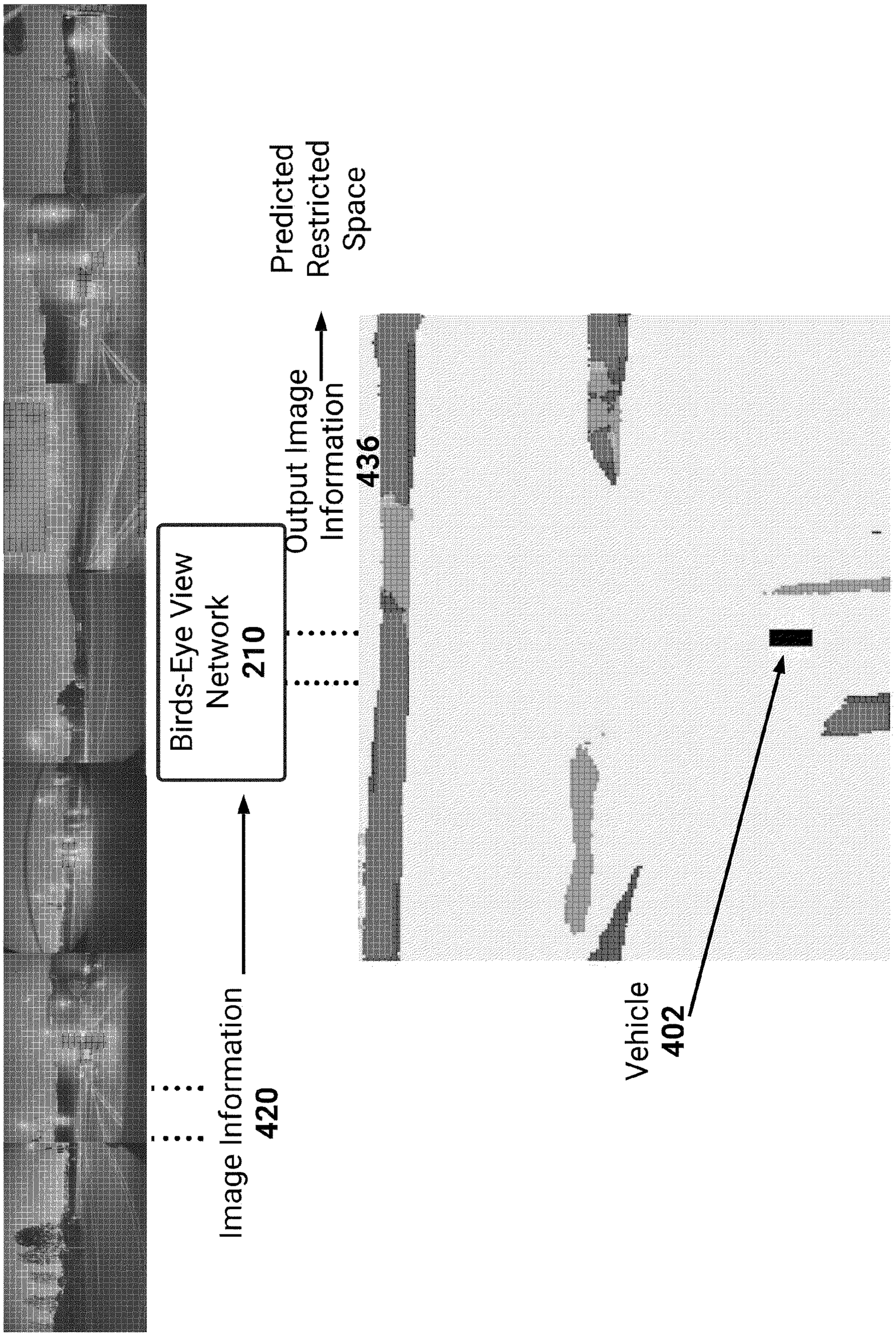

Predicted Restricted Space
Output Image Information
436
Birds-Eye View Network
210
Image Information
420
Vehicle
402

FIG. 4H

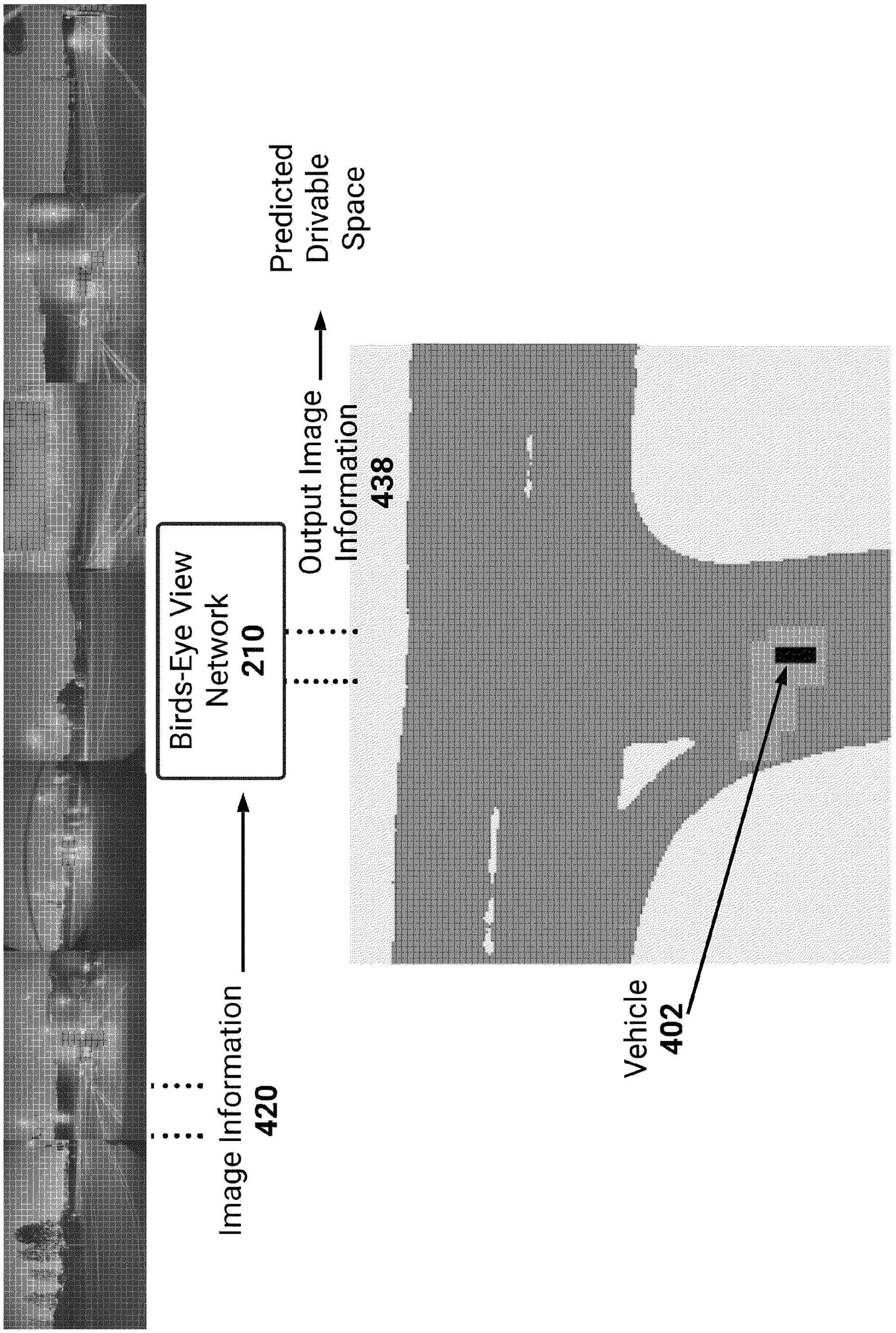

Predicted Drivable Space
Output Image Information
438
Birds-Eye View Network
210
Image Information
420
Vehicle
402

FIG. 4I

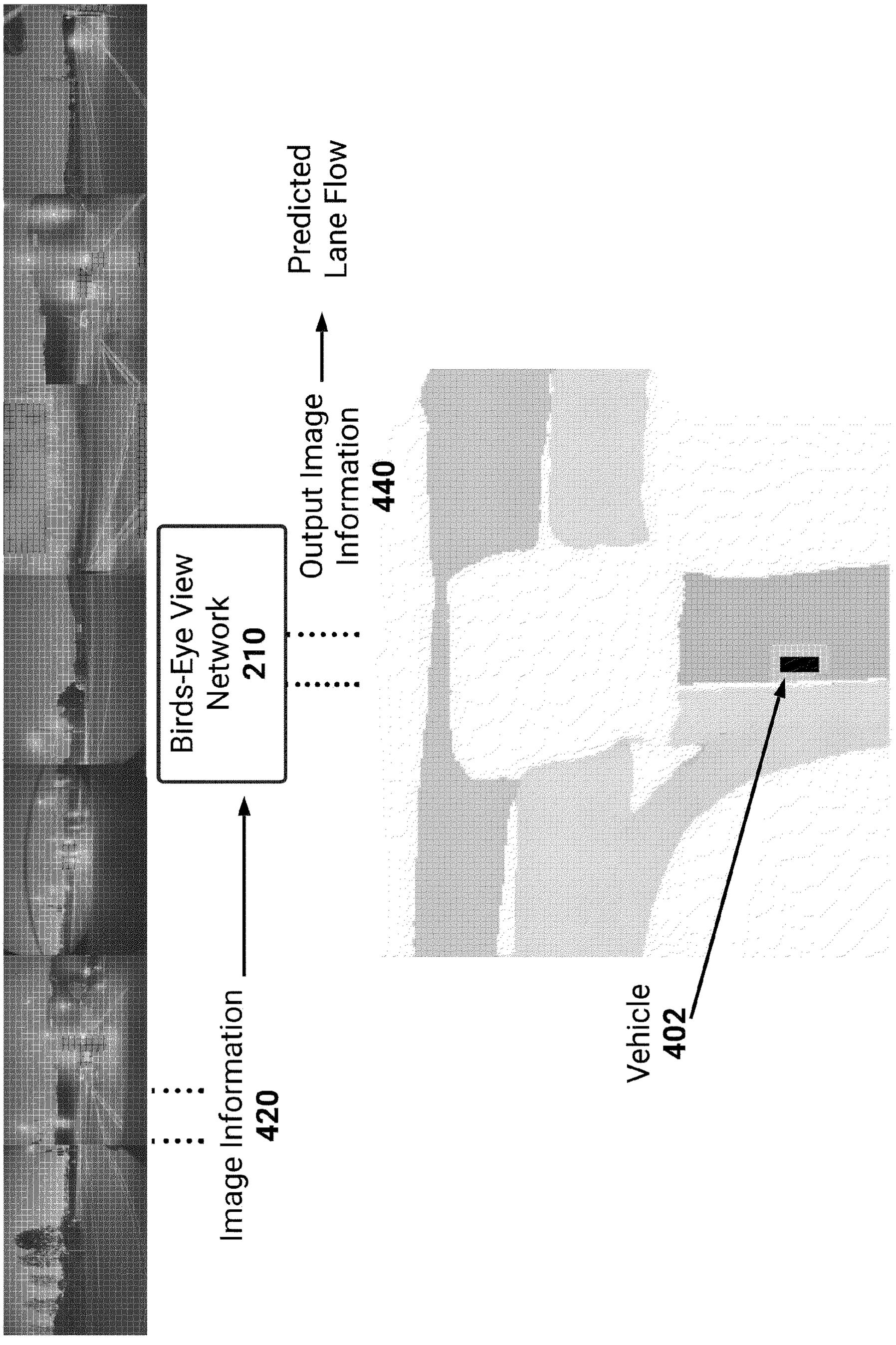

Predicted Lane Flow
Output Image Information
440
Birds-Eye View Network
210
Image Information
420
Vehicle
402

FIG. 4J

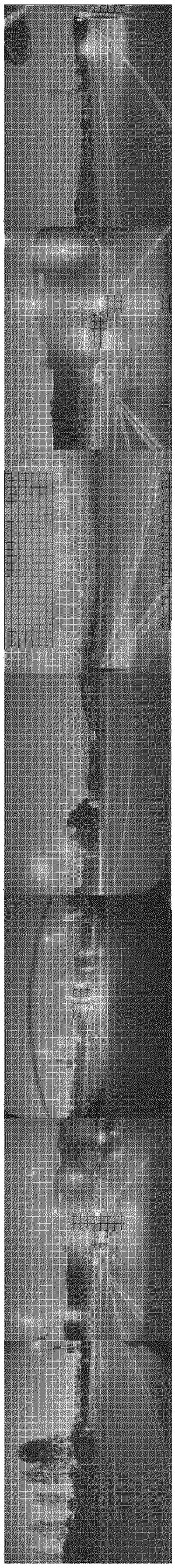


Image Information
**420**

Bird's-Eye View Network
**210**

Predicted Line Color / Apperance

Output Image Information

**442**

Vehicle
**402**

FIG. 4K

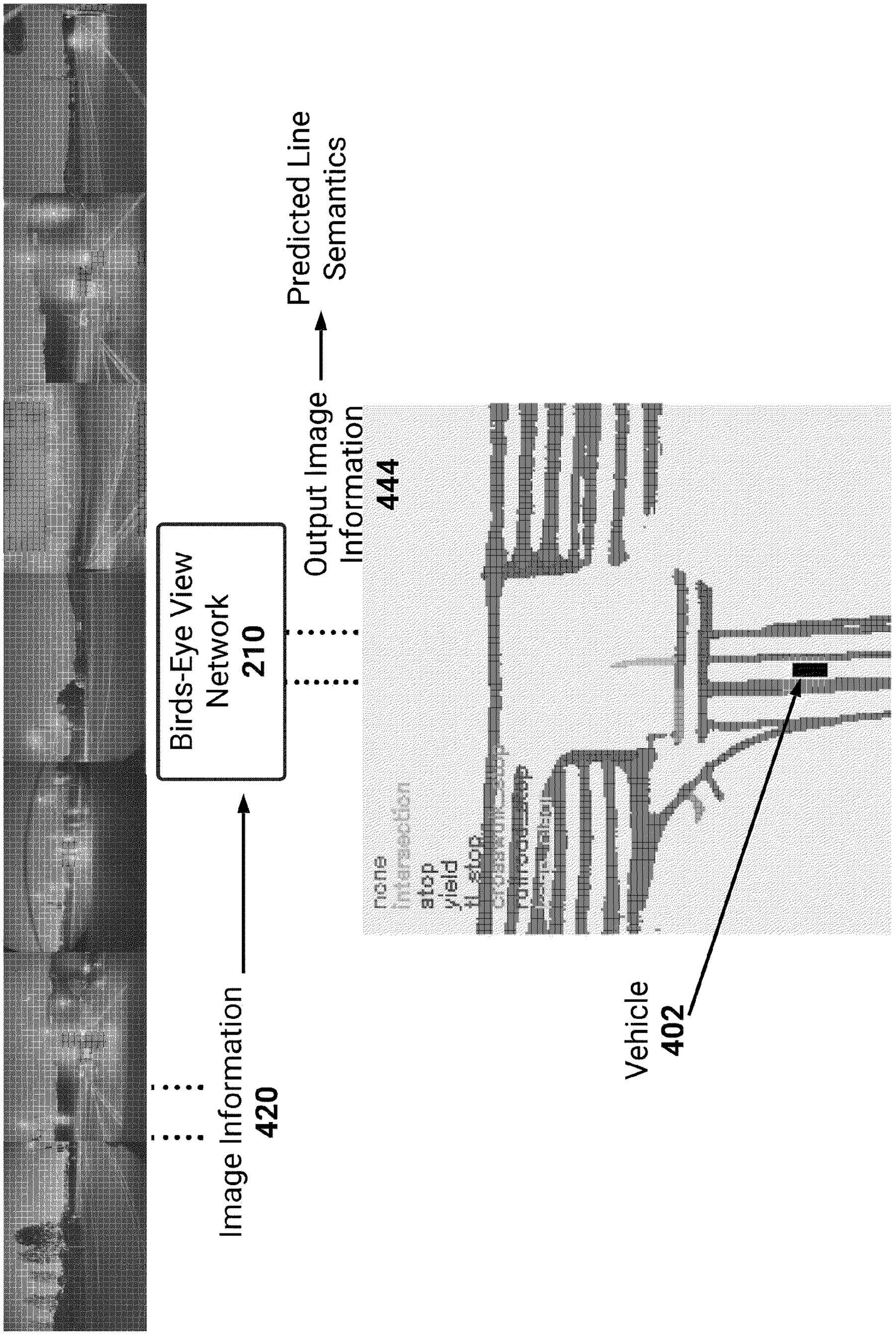


Image Information
420
Birds-Eye View
Network
210
Output Image
Information
444
Predicted Line
Semantics
Vehicle
402

FIG. 4L

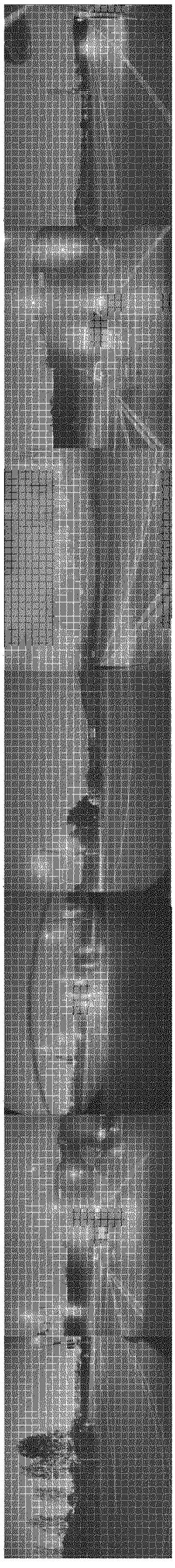


Image Information
420

Bird's-Eye View Network
210

Output Image Information
446

Predicted Line Graph



Vehicle
402

FIG. 4M

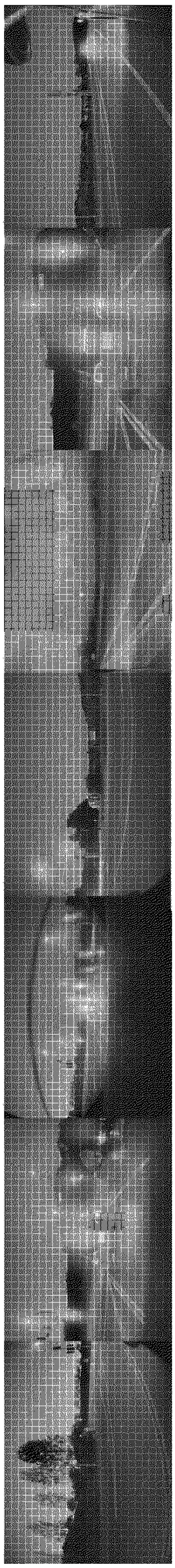


Image Information
420

Bird's-Eye View Network
210

Output Image Information
448

Predicted Vehicle Lane

450

Vehicle
402

FIG. 4N

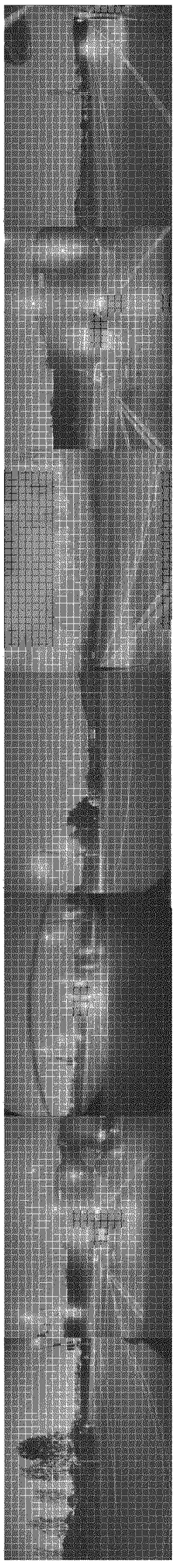


Image Information
420

Bird's-Eye View Network
210

Output Image Information
Predicted Vehicle Road
452




FIG. 4O

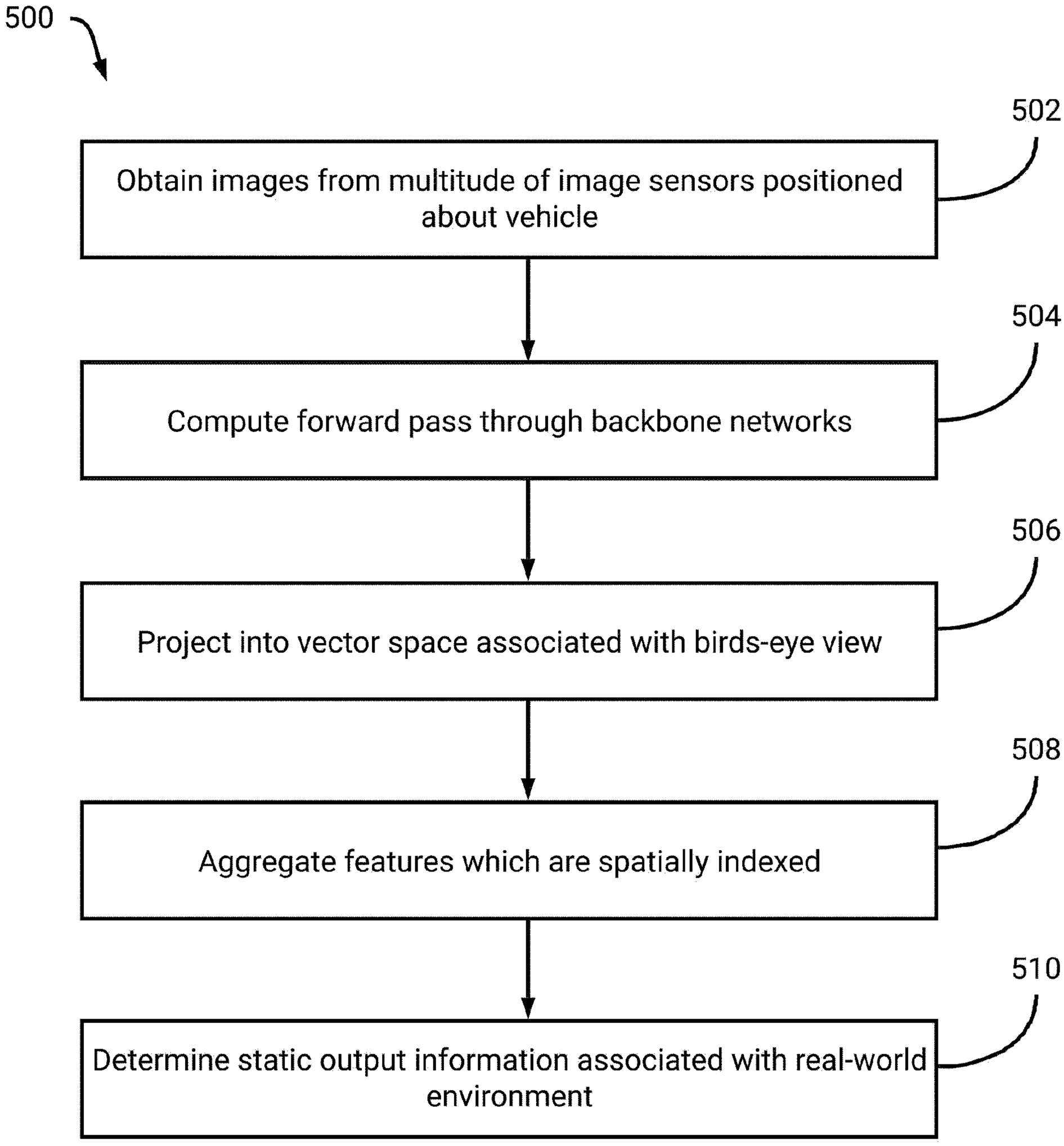

500
502
Obtain images from multitude of image sensors positioned about vehicle
504
Compute forward pass through backbone networks
506
Project into vector space associated with birds-eye view
508
Aggregate features which are spatially indexed
510
Determine static output information associated with real-world environment

FIG. 5

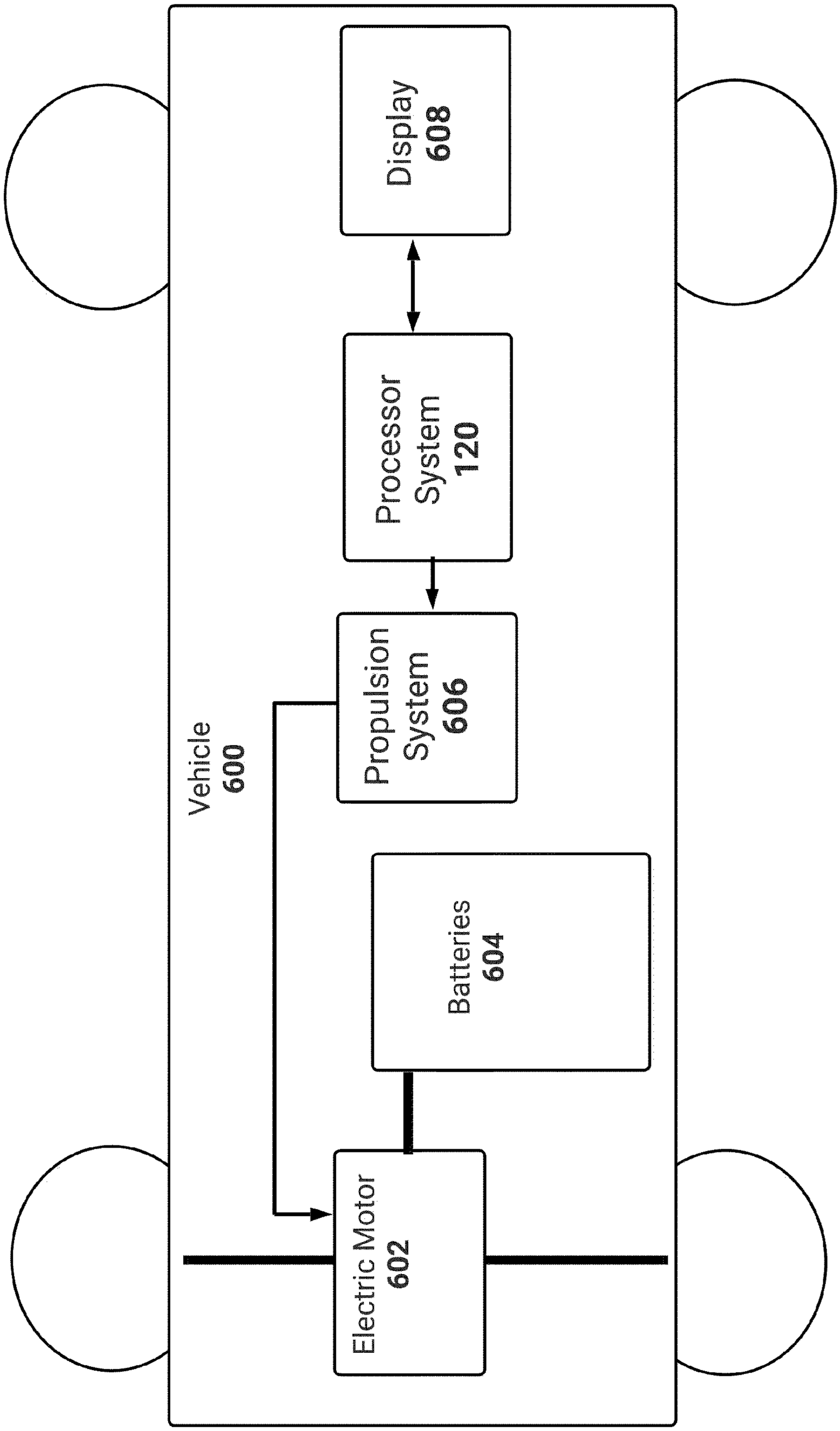

Vehicle
600
Display
608
Processor
System
120
Propulsion
System
606
Batteries
604
Electric Motor
602

FIG. 6

# VISION-BASED MACHINE LEARNING MODEL FOR AGGREGATION OF STATIC OBJECTS AND SYSTEMS FOR AUTONOMOUS DRIVING

## CROSS-REFERENCE TO RELATED APPLICATIONS

**[0001]** This application claims priority to U.S. Prov. Pat. App. No. 63/260439 titled "ENHANCED SYSTEMS AND METHODS FOR AUTONOMOUS VEHICLE OPERATION AND TRAINING" and filed on Aug. 19, 2021. This application claims priority to U.S. Prov. Pat. App. No. 63/287936 titled "ENHANCED SYSTEMS AND METHODS FOR AUTONOMOUS VEHICLE OPERATION AND TRAINING" and filed on Dec. 9, 2021. Each of the above-recited applications is hereby incorporated herein by reference in its entirety.

## BACKGROUND

### Technical Field

**[0002]** The present disclosure relates to machine learning models, and more particularly, to machine learning models using vision information.

### Description of Related Art

**[0003]** Neural networks are relied upon for disparate uses and are increasingly forming the underpinnings of technology. For example, a neural network may be leveraged to perform object classification on an image obtained via a user device (e.g., a smart phone). In this example, the neural network may represent a convolutional neural network which applies convolutional layers, pooling layers, and one or more fully-connected layers to classify objects depicted in the image. As another example, a neural network may be leveraged for translation of text between languages. For this example, the neural network may represent a recurrent-neural network.

**[0004]** Complex neural networks are additionally being used to enable autonomous or semi-autonomous driving functionality for vehicles. For example, an unmanned aerial vehicle may leverage a neural network, in part, to enable navigation about a real-world area. In this example, the unmanned aerial vehicle may leverage sensors to detect upcoming objects and navigate around the objects. As another example, a car or truck may execute neural network(s) to navigate about a real-world area. At present, such neural networks may rely upon costly, or error-prone, sensors. Additionally, such neural networks may lack accuracy with respect to detecting and classifying moving and stationary (e.g., fixed) objects causing deficient autonomous or semi-autonomous driving performance.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0005]** FIG. **1**A is a block diagram illustrating an example autonomous or semi-autonomous vehicle which includes a multitude of image sensors an example processor system.

**[0006]** FIG. **1**B is a block diagram illustrating the example processor system determining static information based on received image information from the example image sensors.

**[0007]** FIG. **2** is a block diagram of an example vision-based machine learning model which includes an example birds-eye view network.

**[0008]** FIG. **3**A is a block diagram illustrating detail of the example birds-eye view network.

**[0009]** FIG. **3**B is a block diagram illustrating an example birds-eye view associated with a virtual camera.

**[0010]** FIG. **4**A illustrates an example output associated with the birds-eye view network.

**[0011]** FIG. **4**B illustrates another example output associated with the birds-eye view network.

**[0012]** FIG. **4**C illustrates another example output associated with the birds-eye view network.

**[0013]** FIG. **4**D illustrates another example output associated with the birds-eye view network.

**[0014]** FIG. **4**E illustrates another example output associated with the birds-eye view network.

**[0015]** FIG. **4**F illustrates another example output associated with the birds-eye view network.

**[0016]** FIG. **4**G illustrates another example output associated with the birds-eye view network.

**[0017]** FIG. **4**H illustrates another example output associated with the birds-eye view network.

**[0018]** FIG. **4**I illustrates another example output associated with the birds-eye view network.

**[0019]** FIG. **4**J illustrates another example output associated with the birds-eye view network.

**[0020]** FIG. **4**K illustrates another example output associated with the birds-eye view network.

**[0021]** FIG. **4**L illustrates another example output associated with the birds-eye view network.

**[0022]** FIG. **4**M illustrates another example output associated with the birds-eye view network.

**[0023]** FIG. **4**N illustrates another example output associated with the birds-eye view network.

**[0024]** FIG. **4**O illustrates another example output associated with the birds-eye view network.

**[0025]** FIG. **5** is a flowchart of an example process for determining static information positioned about an autonomous or semi-autonomous vehicle using a vision-based machine learning model.

**[0026]** FIG. **6** is a block diagram illustrating an example vehicle which includes the example processor system.

## DETAILED DESCRIPTION

**[0027]** Embodiments of the present disclosure and their advantages are best understood by referring to the detailed description that follows. It should be appreciated that like reference numerals are used to identify like elements illustrated in one or more of the figures, wherein showings therein are for purposes of illustrating embodiments of the present disclosure and not for purposes of limiting the same.

### Introduction

**[0028]** This application describes enhanced techniques for autonomous or semi-autonomous (collectively referred to herein as autonomous) driving of a vehicle using image sensors (e.g., cameras) positioned about the vehicle. Thus, the vehicle may navigate about a real-world area using vision-based sensor information. As may be appreciated, humans are capable of driving vehicles using vision and a deep understanding of their real-world surroundings. For example, humans are capable of rapidly identifying objects (e.g.,

pedestrians, road signs, lane markings, vehicles) and using these objects to inform driving of vehicles. Increasingly, machine learning models are capable of identifying and characterizing objects positioned about vehicles. However, such machine learning models are prone to errors introduced through unsophisticated models and/or inconsistencies introduced through disparate sensors.

**[0029]** This application therefore describes a vision-based machine learning model which relies upon increased software complexity to enable a reduction in sensor-based hardware complexity while enhancing accuracy. For example, only image sensors may be used in some embodiments. Through use of image sensors, such as cameras, the described model enables a sophisticated simulacrum of human vision-based driving. As will be described, the machine learning model may obtain images from the image sensors and combine (e.g., stitch or fuse) the information included therein. For example, the information may be combined into a vector space which is then further processed by the machine learning model to extract objects, signals associated with the objects, and so on.

**[0030]** In contrast, another example technique may include identifying objects included in images from each image sensor. These objects may then be aggregated to determine a consistent set of objects in the images. For example, a first image sensor (e.g., a left image sensor) may depict a portion of a truck positioned to the left of a vehicle. In this example, a second image sensor (e.g., a front wide-angle sensor) may include another portion of the truck. Thus, this example technique may require that the portions of the truck be separately identified and then combined into a view of the truck. Such a combination may rely upon hand-tuned models and code which may introduce errors and be difficult to update. In contrast, the techniques described herein allow for the machine learning model to detect objects based on the vector space described above. For example, the machine learning model may effectuate a unified prediction, doing the job of stitching these different views internally by interpreting all images as one.

**[0031]** Furthermore, and as will be described, to limit occlusion of objects and ensure substantial range of visibility of objects, the information may be projected in a vector space according to a birds-eye view. As described herein, the birds-eye view represents a view of a real-world environment about a vehicle in which a virtual camera is pointing downwards at a particular height. Thus, objects positioned about the vehicle are projected into this birds-eye view vector space effectuating the accurate identification of certain types of objects.

**[0032]** As may be appreciated, the birds-eye view may include objects within a threshold distance of the vehicle. For certain types of objects, this type of projection view may be advantageous to an understanding of the real-world environment. In some embodiments, the described model may identify, or determine information regarding, static objects or static information associated with the real-world environment. Example static objects or information may include lane markings, crosswalks, bike lanes, direction of travel for a road or lane therein, intersections, connectivity between lanes which are separated via an intersection, and so on. Additionally, static information may include visibility information (e.g., measures associated with what the image sensors can see). This visibility information may include moving objects in some embodiments, for example if another vehicle is occluding a portion of a road the visibility information may indicate that this portion of not visible.

**[0033]** Without being constrained by way of example, a static object may represent a real-world object, marking, indicator, sign, feature, or characteristic of a real-world environment (e.g., direction of travel in a lane), which is expected to be substantially the same or generally unaltered as an autonomous vehicle navigates. A static object may also represent an object or information which is not a vulnerable or non-vulnerable road user (e.g., a vehicle, a person, a biker, a stroller, and so on). As an example, a lane may be expected to have traffic flow in a same direction as the autonomous vehicle navigates proximate to that lane. In this example, and as may be appreciated, a governmental officer (e.g., a police officer), or sign, may temporarily adjust the direction of traffic. This temporary adjustment may represent a feature or characteristic of the real-world environment and thus may be a static object or information which is detectable via the machine learning model described herein.

**[0034]** With respect to the above-example static objects, the birds-eye view may allow for a rapid understanding of important elements which are relied upon to effectuate autonomous driving. Indeed, stationary objects may inform the outlines of what is navigable in a real-world environment. For example, lane markings can be included in the birds-eye view as would be seen on a navigation map. In this example, the lane markings may be relied upon to inform future navigation options which are available to an autonomous vehicle. As another example, a bike-lane may be identified in the birds-eye view. For this example, the route of the bike-lane may be determined based on the image sensors positioned about the vehicle and updated as the autonomous vehicle navigates. In this way, and as one example, the vehicle may monitor for locations at which the bike-lane merges with vehicle lanes.

**[0035]** The birds-eye view may thus aggregate static objects which are detected proximate (e.g., in visual range) of an autonomous vehicle.

**[0036]** The birds-eye view network described herein may include disparate elements which, in some embodiments, may be end-to-end trained. As will be described, images from image sensors may be provided to respective backbone networks. In some embodiments, these backbone networks may be convolutional neural networks which output feature maps for use later in the network. A transformer network, such as a self-attention network, may receive the feature maps and transform the information into an output vector space. A feature queue may then push the output from the transformer network, optionally along with kinematics of a vehicle (e.g., an autonomous vehicle), into a queue which is optionally spatially indexed. Output from the feature queues may be provided to one or more video modules (e.g., video queues) for processing. In some embodiments, a video module may be a convolutional neural network, a recurrent neural network, or a transformer network. Trunks of the birds-eye view network may then obtain output, or a particular portion thereof, from the video module and generate output information using respective heads. Example output information is described in more detail below with respect to, at least, FIGS. **4**A-**4**O.

**[0037]** As described above, the feature queue may be spatially indexed. As may be appreciated, static information may not be expected to be temporally variable. For example,

a road lane may be expected to be the same as the vehicle moves spatially forward. In contrast, other vehicles, pedestrians, and so on, may be expected to vary in time due to their own movements. Thus, the feature queue may be spatially indexed such that information is aggregated over the previous threshold distance. In this way, if the vehicle is stuck in traffic and not moving then the vehicle may maintain its consistent view of the real-world environment as determined according to recent spatial movements.

**[0038]** Thus, the disclosed technology allows for enhancements to autonomous driving models while reducing sensor-complexity. For example, other sensors (e.g., radar, Lidar, and so on) may be removed during operation of the vehicles described herein. As may be appreciated, radar may introduce faults during operation of vehicles which may lead to phantom objects being detected. Additionally, lidar may introduce errors in certain weather conditions and lead to substantial manufacturing complexity in vehicles.

**[0039]** While description related to an autonomous vehicle (e.g., a car) is included herein, as may be appreciated the techniques may be applied to other autonomous vehicles. For example, the machine learning model described herein may be used, in part, to autonomously operate unmanned ground vehicles, unmanned aerial vehicles, and so on. Additionally, reference to an autonomous vehicle may, in some embodiments, represent a vehicle which may be placed into an autonomous driving mode. For example, the vehicle may autonomously drive or navigate on a highway, freeway, and so on. In some embodiments, the vehicle may autonomously drive or navigate on city roads.

## Block Diagram - Vehicle Processing System / Birds-Eye View Network

**[0040]** FIG. **1**A is a block diagram illustrating an example autonomous vehicle **100** which includes a multitude of image sensors **102**A-**102**F and an example processor system **120**. The image sensors **102**A-**102**F may include cameras which are positioned about the vehicle **100**. For example, the cameras may allow for a substantially 360-degree view around the vehicle **100**.

**[0041]** The image sensors **102**A-**102**F may obtain images which are used by the processor system **120** to, at least, determine information associated with objects positioned proximate to the vehicle **100**. The images may be obtained at a particular frequency, such as 30 Hz, 36 Hz, 60 Hz, 65 Hz, and so on. In some embodiments, certain image sensors may obtain images more rapidly than other image sensors. As will be described below, these images may be processed by the processor system **120** based on the vision-based machine learning model described herein.

**[0042]** Image sensor A **102**A may be positioned in a camera housing near the top of the windshield of the vehicle **100**. For example, the image sensor A **102**A may provide a forward view of a real-world environment in which the vehicle is driving. In the illustrated embodiment, image sensor A **102**A includes three image sensors which are laterally offset from each other. For example, the camera housing may include three image sensors which point forward. In this example, a first of the image sensors may have a wide-angled (e.g., fish-eye) lens. A second of the image sensors may have a normal or standard lens (e.g., 35 mm equivalent focal length, 50 mm equivalent, and so on). A third of the image sensors may have a zoom or narrow-view lens. In this way, three images of varying focal lengths may be obtained in the forward direction by the vehicle **100**.

**[0043]** Image sensor B **102**B may be rear-facing and positioned on the left side of the vehicle **100**. For example, image sensor B **102**B may be placed on a portion of the fender of the vehicle **100**. Similarly, Image sensor C **102**C may be rear-facing and positioned on the right side of the vehicle **100**. For example, image sensor C **102**C may be placed on a portion of the fender of the vehicle **100**.

**[0044]** Image sensor D **102**D may be positioned on a door pillar of the vehicle **100** on the left side. This image sensor **102**D may, in some embodiments, be angled such that it points downward and, at least in part, forward. In some embodiments, the image sensor **102**D may be angled such that it points downward and, at least in part, rearward. Similarly, image sensor E **102**E may be positioned on a door pillow of the vehicle **100** on the right side. As described above, image sensor E **102**E may be angled such that it points downwards and either forward or rearward in part.

**[0045]** Image sensor F **102**F may be positioned such that it points behind the vehicle **100** and obtains images in the rear direction of the vehicle **100** (e.g., assuming the vehicle **100** is moving forward). In some embodiments, image sensor F **102**F may be placed above a license plate of the vehicle **100**.

**[0046]** While the illustrated embodiments include image sensors **102**A-**102**F, as may be appreciated additional, or fewer, image sensors may be used and fall within the techniques described herein.

**[0047]** The processor system **120** may obtain images from the image sensors **102**A-**102**F and detect objects, and information associated with the objects, using the vision-based machine learning model described herein. Based on the objects, the processor system **120** may adjust one or more driving characteristics or features. For example, the processor system **120** may cause the vehicle **100** to turn, slow down, brake, speed up, and so on. While not described herein, as may be appreciated the processor system **120** may execute one or more planning and/or navigation engines or models which use output from the vision-based machine learning model to effectuate autonomous driving.

**[0048]** In some embodiments, the processor system **120** may include one or more matrix processors which are configured to rapidly process information associated with machine learning models. The processor system **120** may be used, in some embodiments, to perform convolutions associated with forward passes through a convolutional neural network. For example, input data and weight data may be convolved. The processor system **120** may include a multitude of multiply-accumulate units which perform the convolutions. As an example, the matrix processor may use input and weight data which has been organized or formatted to facilitate larger convolution operations.

**[0049]** For example, input data may be in the form of a three-dimensional matrix or tensor (e.g., two-dimensional data across multiple input channels). In this example, the output data may be across multiple output channels. The processor system **120** may thus process larger input data by merging, or flattening, each two-dimensional output channel into a vector such that the entire, or a substantial portion thereof, channel may be processed by the processor system **120**. As another example, data may be efficiently reused such that weight data may be shared across convolutions. With respect to an output channel, the weight data **106**

may represent weight data (e.g., kernels) used to compute that output channel.

**[0050]** Additional example description of the processor system, which may use one or more matrix processors, is included in U.S. Pat. No. 11,157,287, U.S. Pat. No. 11,409,692, and U.S. Pat. No. 11,157,441, which are hereby incorporated by reference in their entirety and form part of this disclosure as if set forth herein.

**[0051]** FIG. **1**B is a block diagram illustrating the example processor system **120** determining static information **124** based on received image information **122** from the example image sensors described above.

**[0052]** The image information **122** includes images from image sensors positioned about a vehicle (e.g., vehicle **100**). In the illustrated example of FIG. **1**A, there are 8 image sensors and thus 8 images are represented in FIG. **1**B. For example, a top row of the image information **122** includes three images from the forward-facing image sensors. As described above, the image information **122** may be received at a particular frequency such that the illustrated images represent a particular time stamp of images. In some embodiments, the image information **122** may represent high dynamic range (HDR) images. For example, different exposures may be combined to form the HDR images. As another example, the images from the image sensors may be pre-processed to convert them into HDR images (e.g., using a machine learning model).

**[0053]** In some embodiments, each image sensor may obtain multiple exposures each with a different shutter speed or integration time. For example, the different integration times may be greater than a threshold time difference apart. In this example, there may be three integration times which are, in some embodiments, about an order of magnitude apart in time. The processor system **120**, or a different processor, may select one of the exposures based on measures of clipping associated with images. In some embodiments, the processor system **120**, or a different processor may form an image based on a combination of the multiple exposures. For example, each pixel of the formed image may be selected from one of the multiple exposures based on the pixel not including values (e.g., red, green, blue) values which are clipped (e.g., exceed a threshold pixel value).

**[0054]** The processor system **120** may execute a vision-based machine learning model engine **126** to process the image information **122**. An example of the vision-based machine learning model is described in more detail below, with respect to FIG. **2**-**3**B. As described herein, the vision-based machine learning model may combine information included in the images. For example, each image may be provided to a particular backbone network. In some embodiments, the backbone networks may represent convolutional neural networks. Outputs of these backbone networks may then, in some embodiments, be combined (e.g., formed into a tensor) or may be provided as separate tensors to one or more further portions of the model. In some embodiments, an attention network (e.g., cross-attention) may receive the combination or may receive input tensors associated with each image sensor.

**[0055]** The combined output, as will be described, may then be used to determine disparate static output information **124** associated with a real-world environment. Example output information **124** may be provided as a birds-eye view of the real-world environment and include, for example, information related to one or more of edges, lines, dividers, islands, intersections, drivable space, restricted regions, road blockages, traffic flow (e.g., directions of travel for each lane proximate to an autonomous vehicle), crosswalks, visibility, and so on. In some embodiments, the output information may form respective images which embody or otherwise encode at least a portion of the above-described output information. For example, an image depicting islands (e.g., illustrated in FIG. **4**C) may be formed by the engine **126**. In this example, pixels may be assigned colors depending on a probability or likelihood value of the pixel being associated with an island.

**[0056]** Additionally, and as will be described, the vision-based machine learning model engine **126** may aggregate information which is spread across time and/or space. For example, a video module may be used to aggregate information which is determined as the autonomous vehicle navigates in a real-world environment. In this example, the information may be aggregated over a prior amount of drivable space. As an example, static information may be expected to remain substantially similar (e.g., a road line is not expected to suddenly change). Thus, a specific road line feature (e.g., a specific portion of a road line in the real-world environment) may be spatially indexed such that it is maintained via the video module until the autonomous vehicle is greater than a threshold distance away from the road line feature (e.g., 50 meters, 80 meters, 100 meters, 150 meters, and so on). In this say, static information (e.g., static objects) may be tracked such that the processor system **120** monitors their location even when temporarily occluded or if substantial time has passed (e.g., the autonomous vehicle is sitting in traffic, at a stop-light, and so on).

**[0057]** FIG. **2** is a block diagram of an example vision-based machine learning model which includes a birds-eye view network **210**. The example model may be executed by an autonomous vehicle, such as vehicle **100**. Thus, actions of the model may be understood to be performed by a processor system (e.g., system **120**) included in the vehicle. Example output **212** is included in FIG. **2**, and as may be appreciated, the output may be indicative of types of output from the network **210**. Example graphical representations of the output **212** is included in FIGS. **4**A-**4**O.

**[0058]** In the illustrated example, images **202**A-**202**H are received by the vision-based machine learning model. These images **202**A-**202**H may be obtained from image sensors positioned about the vehicle, such as image sensors **102**A-**102**F. The vision-based machine learning model includes backbone networks **200** which receive respective images as input. Thus, the backbone networks **200** process the raw pixels included in the images **202**A-**202**H. In some embodiments, the backbone networks **200** may be convolutional neural networks. For example, there may be 5, 10, 15, and so on, convolutional layers in each backbone network. In some embodiments, the backbone networks **200** may include residual blocks, recurrent neural network-regulated residual networks, and so on. Additionally, the backbone networks **200** may include weighted bi-directional feature pyramid networks (BiFPN). Output of the BiFPNs may represent multi-scale features determined based on the images **202**A-**202**H. In some embodiments, Gaussian blur may be applied to portions of the images at training and/or inference time. For example, road edges may be peaky in that they are sharply defined in images. In this example, a Gaussian blur may be applied to the road edges to allow for

bleeding of visual information such that they may be detectable by a convolutional neural network.

**[0059]** Additionally, certain of the backbone networks **200** may pre-process the images such as performing rectification, cropping, and so on. With respect to cropping, images **202**C from the fisheye forward-facing lens may be vertically cropping to remove certain elements included on a windshield (e.g., a glare shield).

**[0060]** With respect to rectification, the vehicles described herein may be examples of vehicles which are available to millions, or more, end-users. Due to tolerances in manufacturing and/or differences in use of the vehicles, the image sensors in the vehicles may be angled, or otherwise positioned, slightly differently (e.g., differences in roll, pitch, and/or yaw). Additionally, different models of vehicles may execute the same vision-based machine learning model. These different models may have the image sensors positioned and/or angled differently. The vision-based machine learning model described herein may be trained, at least in part, using information aggregated from the vehicle fleet used by end-users. Thus, differences in point of view of the images may be evident due to the slight distinctions between the angles, or positions, of the image sensors in the vehicles included in the vehicle fleet.

**[0061]** Thus, rectification may be performed via the backbone networks **200** to address these differences. For example, a transformation (e.g., an affine transformation) may be applied to the images **202**A-**202**H, or a portion thereof, to normalize the images. In this example, the transformation may be based on camera parameters associated with the image sensors (e.g., image sensors **102**A-**102**F), such as extrinsic and/or intrinsic parameters. In some embodiments, the image sensors may undergo an initial, and optionally repeated, calibrated step. For example, as a vehicle drives the cameras may be calibrated to ascertain camera parameters which may be used in the rectification process. In this example, specific markings (e.g., road lines) may be used to inform the calibration. The rectification may optionally represent one or more layers of the backbone networks **200**, in which values for the transformation are learned based on training data.

**[0062]** The backbone networks **200** may thus output feature maps (e.g., tensors) which are used by birds-eye view network **210**. In some embodiments, the output from the backbone networks **200** may be combined into a matrix or tensor. In some embodiments, the output may be provided as a multitude of tensors (e.g., 8 tensors in the illustrated example) to the birds-eye view net. In the illustrated example, the output is referred to as vision information **204** which is input into the network **210**. While the backbone networks **200** and birds-eye view network **210** are illustrated separately, in some embodiments they may form part of the same network or model (e.g., the vision-based model described herein). Additionally, in some embodiments the backbone networks **200** and birds-eye view network **210** may be end-to-end trained.

**[0063]** The output tensors from the backbone networks **200** may be combined (e.g., fused) together into a virtual camera space (e.g., a vector space) via the birds-eye view network **210**. The image sensors positioned about the autonomous vehicle may be at different heights of the vehicle. For example, the left and rear pillar image sensors may be positioned higher than the left and rear front bumper image sensors. Thus, to allow for a consistent view of objects positioned about the vehicle, the virtual camera space may be used. In the example described herein, the virtual camera space is a birds-eye view (e.g., top-down view) of static objects positioned about the autonomous vehicle. In some embodiments, the birds-eye view may extend laterally by about 70 meters, 80 meters, 100 meters, and so on. In some embodiments, the birds-eye view may extend longitudinally by about 80 meters, 100 meters, 120 meters, 150 meters, and so on. For example, the birds-eye view may include static objects which are positioned in a real-world environment in the lateral and/or longitudinal distance.

**[0064]** For certain information determined by the vision-based machine learning model, the autonomous vehicle's kinematic information **206** may be used. Example kinematic information **206** may include the autonomous vehicles velocity, acceleration, yaw rate, and so on. In some embodiments, the images **202**A-**202**H may be associated with kinematic information **206** determined for a time, or similar time, at which the images **202**A-**202**H were obtained. For example, the kinematic information **206**, such as velocity, yaw rate, acceleration, may be encoded (e.g., embedded into latent space), and associated with the images.

**[0065]** Example static output information **212** from the birds-eye view network **210** is indicated in FIG. **2**. The output information **212** may represent information associated with static objects in the real-world environment about the autonomous vehicle. For example, the output information **212** may include information associated with edges (e.g., road edge, such as sidewalk edge, curb edge, and so on). In this example, the information may indicate positions of the edges in the birds-eye view (e.g., positions with respect to the output vector space) of the edges. In some embodiments, the information **212** may be reflected in an image which is formed based on the output from the network **210**. For example, a pixel of an image may be assigned a color based on a likelihood, or value indicative of a probability, of the pixel being an edge. In this example, the color may be a grayscale color selected based on the likelihood or value.

**[0066]** Additional output **212** may include positions or locations associated with lines, dividers, islands, intersections, drivable space, restricted regions, road blockage, crosswalks, and so on. Furthermore, the output **212** may indicate traffic flow information. For example, and as illustrated in FIG. **4**J, the output **212** may indicate directions of traffic flow for road lanes proximate to the autonomous vehicle. In some embodiments, the output **212** may assign a color, or variations of a color, depending on the direction of traffic. Thus, traffic flowing east may be assigned a color different from traffic flowing west or south. The output **212** may additionally indicate visibility information, indicating portions of the real-world environment which are not visible to the vehicle (e.g., due to occlusions, weather, and so on).

**[0067]** As will be described, the output **212** may be generated via a forward pass through the birds-eye view network **210**. In some embodiments, forward passes may be computed at a particular frequency (e.g., 24 Hz, 30 Hz, and so on). In some embodiments, the output may represent 50, 100, 150, different outputs which indicate different static information associated with the real-world environment. This information may be used, for example, via a planning engine. As an example, the planning engine may determine driving actions to be performed by the autonomous vehicle

(e.g., accelerations, turns, braking, and so on) based on the birds-eye view of the real-world environment.

**[0068]** In some embodiments, map information may be provided as an input to the birds-eye view network **210**. For example, a raster or image of a map proximate to a location of the autonomous vehicle may be provided as an input. The birds-eye view network **210** may be trained to use this information to generate at least a portion of the outputs described herein. For example, the map information may include a representation of roads, lanes included in the roads, medians, islands, bike lanes, crosswalks, intersections, and so on, which are proximate to the vehicle. In this example, the birds-eye view network **210** may use the map information, for example, to determine which lanes connect to which other lanes (e.g., across an intersection).

**[0069]** Further detail regarding the birds-eye view network **210** is included below with respect to FIGS. **3**A-**3**B.

**[0070]** FIG. **3**A is a block diagram illustrating detail of the example birds-eye view network **210**. In the illustrated embodiment, vision information **204** is received by the birds-eye view network **210**. The vision information **204**, as described above, may represent output from the backbone networks **200**. Example output may include features (e.g., multi-scale features, feature maps, and so on) determined based on received images.

**[0071]** A transformer network engine **402** receives the vision information **204** as input. In some embodiments, the transformer network engine **402** is trained to project the information **204** into a virtual camera space. For example, the transformer network engine **402** may perform multi-camera fusion and project information into a birds-eye view camera space. For example, during training the engine **402** may be trained to associate objects detected in images as being positioned within the virtual camera space. In this example, the training data may indicate positions of objects as projected into the view desired (e.g., the birds-eye view). Thus, during training the loss function may cause updating of weights of the model such that the projection into the birds-eye view vector space is effectuated.

**[0072]** Output from the transformer network engine **302** is provided as input to the feature queue engine **304**. To ensure that objects can be tracked as an autonomous vehicle navigates, even while temporarily occluded, the feature queue engine **304** can store output from the engine **302**. For example, the output may be pushed into the queue **304** according to time and/or space. In this example, the time indexing may indicate that the queue **304** stores output from the engine **302** based on passage of time (e.g., information is pushed at a particular frequency). Spatial indexing may indicate that the queue **304** stores output from the engine **302** based on spatial movement of the vehicle. For example, as the vehicle moves in a direction the queue **304** may be updated after a threshold amount of movement (e.g., .2 meters, 1 meter, 3 meters, and so on). Optionally, the threshold amount of movement may be based on a location or speed of the vehicle. For example, navigation on city streets may allow for pushing information to the queue **304** after less movement than navigation on a freeway (e.g., at higher speed). In some embodiments, the queue **304** may store information determined based on images taken at 10 \-time stamps, 12 \-time stamps, 20 \-time stamps, and so on.

**[0073]** Output from the feature queue engine **304** may be combined to form a tensor which is then processed by the remainder of the birds-eye view network **210**. For example, the output 306A-N (e.g., spatially indexed features) may be provided to a video module **308**. The video module **308** may represent a convolutional neural network, which may cause the processor system **120** to perform three-dimensional convolutions. In this way, the video module **308** may allow for tracking of objects over space (e.g., as the vehicle moves). In some embodiments, the video module may represent an attention network (e.g., spatial attention), a recurrent neural network, and so on.

**[0074]** With respect to video module **308**, kinematic information **206** associated with the autonomous vehicle executing the vision-based machine learning model may optionally be input into the module **308**. As described above, the kinematic information **206** may represent one or more of acceleration, velocity, yaw rate, turning information, braking information, and so on. The kinematic information **206** may additionally be associated with features 306A-N from the feature queue engine **304**. Thus, the video module **308** may encode this kinematic information **206** for use in determining, as an example, positions of static objects.

**[0075]** For example, a particular road edge may be identified based on images processed by the birds-eye view network **210**. In this example, the vehicle may move a particular amount with respect in three-dimensions which is determinable based on the kinematic information. Thus, the position of the particular road edge may be adjusted in current images obtained by the vehicle. The birds-eye view network **210** may therefore adjust the position of the particular road edge, for example even if occluded in the current images, based on the kinematic information **206**.

**[0076]** Thus, the video module **308** may perform frame alignment. As described herein, frames may represent images (e.g., image frames) taken at a same time or substantially same time by the image sensors. Thus, the video module **308** may align frames taken at different times (e.g., the feature maps resulting from the frames). For example, frames may be selected according to their spatial index, and may be optionally aligned to correct for the autonomous vehicle's movement. For example, if the vehicle moved 20 meters ahead, then the video module **308** may select, or aggregate information which includes, frame(s) 20 meters earlier (e.g., in the past). In this example, the features of those earlier frame(s) may be spatially shifted to align with the current features which are 20 m ahead. This can be done longitudinally and laterally at the same time, to ensure views are consistent/aligned.

**[0077]** The birds-eye view network **210** includes one or more trunks **314**A-**314**N which obtain information from the video module **308**. For example, each trunk may obtain a portion, or all, of the output from the video module **308**. In this example, each trunk may be trained to generate specific types of output. In some embodiments, the trunks may relate to edges, lines, dividers, islands, intersections, drivable space, restricted regions, road blockage, traffic flow, crosswalks, visibility, and so on. The trunks **314**A-**314**N may be associated with one or more heads **316**A-**316**N, **318**A-**318**N, which output specific information for use by the autonomous vehicle. Example output from the heads **316**A-**316**N, **318**A-**318**N is illustrated in FIGS. **4**A-**4**O.

**[0078]** As known by those skilled in the art, these trunks or heads (collectively referred to herein as heads) may extend from a common portion of a neural network and be trained as experts in determining specific information. In addition to being experts in specific information, the separa-

tion into different heads allows for piecemeal training to quickly incorporate new training data. As new training information is obtained, portions of the machine learning model which would most benefit from the training information may be quickly updated. In this example, the training information may represent images or video clips of specific real-world scenarios gathered by vehicles in real-world operation. Thus, a particular head or heads may be trained, and the weights included in these portions of the network may be updated. For example, other portions (e.g., earlier portions of the network) may not have weights updated to reduce a training time and time to updating end-user autonomous vehicles.

**[0079]** In some embodiments, training data which is directed to one or more of the heads or trunks may be adjusted to focus on those heads or trunks. For example, images may be masked (e.g., loss masked) such that only certain pixels of the images are supervised while otherwise are not supervised. In this example, certain pixels may be assigned a value of zero while other pixels may maintain their values or be assigned a value of one. Thus, if training images depict a rarely seen static object (e.g., a relatively new form of bike lane) then the training images may optionally be masked to focus on that static object. During training, the error generated may be used to train for the loss in the pixels which a labeler has associated with the static object. Thus, only a head or trunk associated with this type of static object may be updated.

**[0080]** To ensure that sufficient training data is obtained, the autonomous vehicles may optionally execute classifiers which are triggered to obtain images which satisfy certain conditions. For example, vehicles operated by end-users may automatically obtain training images which depict, for example, tire spray, rainy conditions, snow, fog, fire soke, and so on. Further description related to use of classifiers is described in U.S. Pat. Pub. No. 2021/0271259 which is hereby incorporated herein by reference in its entirety as if set forth herein.

**[0081]** While the output described above, and in FIGS. **4**A-**4**O, may represent images, or information which may be included in images (e.g., pixel values), in some embodiments additional information may be generated as output from the network **210**. For example, road edges may be represented in an output image as values of specific pixels. In this example, a likelihood (e.g., a value between 0 and 1) of a pixel forming part of a road edge may be converted to grayscale. As may be appreciated, each pixel of the image may correspond to a portion of the real-world environment. For example, each pixel may correspond to an area of a threshold number of centimeters by a second threshold number of centimeters (e.g., 30 cm x 30 cm, 10 cm x 30 cm, 35 cm x 15 cm, 33 cm x 33 cm, and so on).

**[0082]** To reduce this coarseness, in some embodiments the network **210** may be trained to output an offset for each pixel. The offset may indicate how far from the center of a pixel the road edge is. The offset may additionally indicate a direction predicted which is associated with the offset. In this way, the finer detail may represent metrics or information which associated with the pixels of the output image. This information may be used to inform, at least, parking of the vehicle (e.g., the road edge may be curved such that finer accuracy is important). Additionally, the information may be used to determine how far from a road edge the vehicle is to ensure that maneuvers or driving actions are maintained to be smooth via use of enhanced clearance from the road edge.

**[0083]** Additional output may include connectivity information which is usable to indicate which lanes connect to which other lines (e.g., across an intersection). For example, a vehicle may be in the left-most lane in the vehicle's direction of travel (e.g., the right side of the road). In this example, the vehicle may be turning left across an intersection. Thus, the birds-eye view network **210** may determine which lanes across the intersection which connect to the vehicle's current lane. This connectivity may be generated as an image, for example the image of FIGS. **4**M-**4**O.

**[0084]** In some embodiments, however, the connectivity may additionally be represented as splines which connect lanes. For example, the birds-eye view network **230** may, at times, drop parts of the connectivity due to occlusion or other reasons. Thus, in some embodiments, the network **230** (e.g., a specific head) may connect lines with splines. In this way, if a portion of the connectivity represented in an image drops then a spline connecting the vehicle's current line to another lane across an intersection may be relied upon (e.g., in combination with the image or alone).

**[0085]** FIG. **3**B is a block diagram illustrating an example birds-eye view associated with a virtual camera. In the illustrated example, image information **320** is being processed by the processor system **120** based on the birds-eye view network **230**. As described in FIG. **3**A, the processor system **120** maps information included in the image information **320** into a virtual camera space. For example, a birds-eye view **322** is included in FIG. **3**B. Output **324** from the birds-eye view network **230** may be used by a planning / control engine **330** to navigate (e.g., autonomously navigate) the autonomous vehicle.

Block Diagram - Example Output

**[0086]** FIGS. **4**A-**4**O represent example output from the birds-eye view network described herein. As described above, the output may be generated by different heads of the network. The output may optionally be images in which pixels are used to indicate information. For example, the pixels may be assigned colors, or gradations of colors (e.g., to indicate likelihoods), to indicate information. FIG. **4**A-**4**) may thus depict examples of static objects which are detectable by the birds-eye view network.

**[0087]** FIG. **4**A illustrates an example output associated with the birds-eye view network. In the illustrated embodiment, image information **420** may be processed by the birds-eye view network **230** (e.g., by a processor system computing a forward pass through the network **230**) to generate output image information **404**. In FIG. **4**A, the output **404** represents predicted external edges. For example, a vehicle **402** which includes the processor system **120** is depicted in the output **404**. Proximate to the vehicle **402** are external road edges which are determined based on the image information **420** and projected into the birds-eye view vector space.

**[0088]** FIG. **4**B illustrates another example output associated with the birds-eye view network **230**. In FIG. **4**B, predicted dividers are represented in an output image **406** from the birds-eye view network **230**. These dividers are evident in portions of the image information **420** (e.g., portion **408**). As described above, the dividers, and other output information, may be determined based on information

which is aggregated as the vehicle drives such that temporary occlusions do not impact the inclusion in the output image **406**.

**[0089]** FIG. **4**C illustrates another example output associated with the birds-eye view network **230**. In FIG. **4**C, predicted islands are represented in an output image **410**. An example divider is illustrated in the output image **410** and corresponds with a portion **412** of the image information **420**.

**[0090]** FIG. **4**D illustrates another example output associated with the birds-eye view network **230**. In FIG. **4**D, predicted lines are included in an output image **414**. These predicted lines may represent, for example, road edges, road lines, bike lanes, and so on.

**[0091]** FIG. **4**E illustrates another example output associated with the birds-eye view network **230**. In FIG. **4**E, superpixel information is used to generate an output image **416**. As described above, in some embodiments the birds-eye view network **230** may determine metrics indicating respective extents to which lines or edges are offset from pixels in an image included in the output information (e.g., intra-pixel precision). In some embodiments, these metrics may be associated with the image and used, for example, to inform planning and/or navigation of the autonomous vehicle. In some embodiments, the information may be used to refine the image.

**[0092]** FIG. **4**F illustrates another example output associated with the birds-eye view network **230**. In FIG. **4**F, the birds-eye view network **230** has generated an image **418** indicative of predicted regions. For example, a first region **422** may indicate non-drivable space. As another example, a second region **424** may indicate lane dividers. As another example, a third region **426** may indicate an intersection. As another example, a fourth region **428** may indicate an island. As another example, a fifth region **430** may indicate drivable space. These different regions may be assigned different colors by the network **230**.

**[0093]** FIG. **4**G illustrates another example output associated with the birds-eye view network **230**. In FIG. **4**G, the birds-eye view network **230** has generated an image **432** which indicates crosswalk regions. In some embodiments, colors may be assigned to the crosswalk region **434** to indicate what angle (e.g., a vector) that pedestrians are expected to walk in. Thus, a first color may indicate that pedestrians are to walk between left and right while a second color may indicate that pedestrians are to walk between top and bottom.

**[0094]** FIG. **4**H illustrates another example output associated with the birds-eye view network **230**. In FIG. **4**H, the birds-eye view network **230** has generated an image **436** which indicates predicted restricted space. Restricted space may indicate, for example, portions which the autonomous vehicle is not able to drive in. For example, bike lanes, sidewalks, and so on, may be included in the image **436** optionally in distinct colors.

**[0095]** FIG. **4**I illustrates another example output associated with the birds-eye view network **230**. In FIG. **4**I, the birds-eye view network **230** has generated an image **438** which indicates predicted drivable space. Thus, the image **438** may indicate road portions.

**[0096]** FIG. **4**J illustrates another example output associated with the birds-eye view network **230**. In FIG. **4**J, the birds-eye view network **230** has generated an image **440** which indicates predicted lane flow. The image **440** may indicate, for each pixel, which direction cars are moving towards. For example, the image **440** may include different colors indicating different flow directions. As an example, a first color may indicate movement to the left, a second color may indicate movement to the right, a third color may indicate movement up, a fourth color may indicate movement down, and so on. The intersection may optionally not be assigned a color as the lane flow is invalid.

**[0097]** FIG. **4**K illustrates another example output associated with the birds-eye view network **230**. In FIG. **4**K, the birds-eye view network **230** has generated an image **442** which indicates road line color and/or appearance. With respect to color, the image **442** may indicate while road lines, yellow road lines, and so on. With respect to appearance, the image **442** may indicate dotted lines, double lines, single lines, and so on.

**[0098]** FIG. **4**L illustrates another example output associated with the birds-eye view network **230**. In FIG. **4**L, the birds-eye view network **230** has generated an image **444** which indicates line semantics. For example, lines associated with an intersection, stop sign, yield sign, crosswalk stop, railroad stop, keep clear, and so on, may be included in the image **444** optionally with different colors.

**[0099]** FIG. **4**M illustrates another example output associated with the birds-eye view network **230**. In FIG. **4**M, the birds-eye view network **230** has generated an image **446** which may aggregate disparate information associated with lines. For example, bike lanes, medians, islands, intersections, lane connectivity information, and so on, may be included in the image **446** optionally with different colors.

**[0100]** FIG. **4**N illustrates another example output associated with the birds-eye view network **230**. In FIG. **4**N, the birds-eye view network **230** has generated an image **448** which identifies potential connections between a current lane of the autonomous vehicle and one or more lanes across an intersection. For example, a particular color may indicate the connections (e.g., connection **450**).

**[0101]** FIG. **4**O illustrates another example output associated with the birds-eye view network **230**. In FIG. **4**O, the birds-eye view network **230** has generated an image **452** which identifies connections between a current road of the autonomous vehicle and proximate roads. For example, the connections include connections **452-454** to the road in the upper left part of the image **452**. The connections also include connections **456-458** in the lower right part of the image **452**.

## Example Flowchart

**[0102]** FIG. **5** is a flowchart of an example process **500** for determining static information positioned about an autonomous or semi-autonomous vehicle using a vision-based machine learning model. For convenience, the process **500** will be described as being performed by a system of one or more processors (e.g., the processor system **120**, which may be included in a vehicle).

**[0103]** At block **502**, the system obtains images from multitude of image sensors positioned about a vehicle. As described above, there may be 7, 8, 10, and so on, image sensors used to obtain images. At block **504**, the system computes a forward pass-through backbone networks. The backbone networks may represent convolutional neural networks which optionally pre-process the images (e.g., rectify the images, crop the images, and so on).

**[0104]** At block **506**, the system projects features determined from the images into a birds-eye view. For example, a transformer network included in a birds-eye view model may project the features into a consistent vector space. In this example, the transformer network may be trained to associate features extracted from the images into the birds-eye view projection. Optionally, a forced projection step may precede the transformer network to, at least in part, cause the projection into the birds-eye view.

**[0105]** At block **508**, the system aggregates spatially indexed features. As described above, a video module included in the birds-eye view network may be used to aggregate information which is determined within a threshold distance from the vehicle's location. For example, the video module may obtain features from a spatially indexed queue which were determined over the previous 50 meters, 75 meters, 100 meters, and so on.

**[0106]** At block **510**, the system determines static output information associated with a. real-world environment in which the vehicle is driving. For example, different trunks and heads included in the birds-eye view network may be used to generate the output information.

**[0107]** In some embodiments, the information (e.g., the images described herein) determined by the birds-view network may be presented in a display of the vehicle. For example, the information may be used to inform autonomous driving (e.g., used by a planning and/or navigation engine) and optionally presented as a visualization for a driver or passenger to view. In some embodiments, the information may be used only as a visualization. For example, the driver or passenger may toggle an autonomous mode off. The visualization may also represent a rendering based on the information. For example, three-dimensional graphics of objects (e.g., lane lanes, bike lanes) may be rendered based on positional information determined by the birds-eye view network.

## Vehicle Block Diagram

**[0108]** FIG. **6** illustrates a block diagram of a vehicle **600** (e.g., vehicle **100**). The vehicle **600** may include one or more electric motors **602** which cause movement of the vehicle **600**. The electric motors **602** may include, for example, induction motors, permanent magnet motors, and so on. Batteries **604** (e.g., one or more battery packs each comprising a multitude of batteries) may be used to power the electric motors **602** as is known by those skilled in the art.

**[0109]** The vehicle **600** further includes a propulsion system **606** usable to set a gear (e.g., a propulsion direction) for the vehicle. With respect to an electric vehicle, the propulsion system **606** may adjust operation of the electric motor **602** to change propulsion direction.

**[0110]** Additionally, the vehicle includes the processor system **120** which processes data, such as images received from image sensors **102**A-**102**F positioned about the vehicle **600**. The processor system **120** may additionally output information to, and receive information (e.g., user input) from, a display **608** included in the vehicle **600**. For example, the display may present graphical depictions of static objects positioned about the vehicle **600**.

## Other Embodiments

**[0111]** All of the processes described herein may be embodied in, and fully automated, via software code modules executed by a computing system that includes one or more computers or processors. The code modules may be stored in any type of non-transitory computer-readable medium or other computer storage device. Some or all the methods may be embodied in specialized computer hardware.

**[0112]** Many other variations than those described herein will be apparent from this disclosure. For example, depending on the embodiment, certain acts, events, or functions of any of the algorithms described herein can be performed in a different sequence or can be added, merged, or left out altogether (for example, not all described acts or events are necessary for the practice of the algorithms). Moreover, in certain embodiments, acts or events can be performed concurrently, for example, through multi-threaded processing, interrupt processing, or multiple processors or processor cores or on other parallel architectures, rather than sequentially. In addition, different tasks or processes can be performed by different machines and/or computing systems that can function together.

**[0113]** The various illustrative logical blocks, modules, and engines described in connection with the embodiments disclosed herein can be implemented or performed by a machine, such as a processing unit or processor, a digital signal processor (DSP), an application specific integrated circuit (ASIC), a field programmable gate array (FPGA) or other programmable logic device, discrete gate or transistor logic, discrete hardware components, or any combination thereof designed to perform the functions described herein. A processor can be a microprocessor, but in the alternative, the processor can be a controller, microcontroller, or state machine, combinations of the same, or the like. A processor can include electrical circuitry configured to process computer-executable instructions. In another embodiment, a processor includes an FPGA or other programmable device that performs logic operations without processing computer-executable instructions. A processor can also be implemented as a combination of computing devices, for example, a combination of a DSP and a microprocessor, a plurality of microprocessors, one or more microprocessors in conjunction with a DSP core, or any other such configuration. Although described herein primarily with respect to digital technology, a processor may also include primarily analog components. For example, some or all of the signal processing algorithms described herein may be implemented in analog circuitry or mixed analog and digital circuitry. A computing environment can include any type of computer system, including, but not limited to, a computer system based on a microprocessor, a mainframe computer, a digital signal processor, a portable computing device, a device controller, or a computational engine within an appliance, to name a few.

**[0114]** Conditional language such as, among others, "can," "could," "might" or "may," unless specifically stated otherwise, are understood within the context as used in general to convey that certain embodiments include, while other embodiments do not include, certain features, elements and/or steps. Thus, such conditional language is not generally intended to imply that features, elements and/or steps are in any way required for one or more embodiments or that one or more embodiments necessarily include logic for deciding, with or without user input or prompting, whether these features, elements and/or steps are included or are to be performed in any particular embodiment.

**[0115]** Disjunctive language such as the phrase "at least one of X, Y, or Z," unless specifically stated otherwise, is understood with the context as used in general to present that an item, term, etc., may be either X, Y, or Z, or any combination thereof (for example, X, Y, and/or Z). Thus, such disjunctive language is not generally intended to, and should not, imply that certain embodiments require at least one of X, at least one of Y, or at least one of Z to each be present.

**[0116]** Any process descriptions, elements or blocks in the flow diagrams described herein and/or depicted in the attached figures should be understood as potentially representing modules, segments, or portions of code which include one or more executable instructions for implementing specific logical functions or elements in the process. Alternate implementations are included within the scope of the embodiments described herein in which elements or functions may be deleted, executed out of order from that shown, or discussed, including substantially concurrently or in reverse order, depending on the functionality involved as would be understood by those skilled in the art.

**[0117]** Unless otherwise explicitly stated, articles such as "a" or "an" should generally be interpreted to include one or more described items. Accordingly, phrases such as "a device configured to" are intended to include one or more recited devices. Such one or more recited devices can also be collectively configured to carry out the stated recitations. For example, "a processor configured to carry out recitations A, B and C" can include a first processor configured to carry out recitation A working in conjunction with a second processor configured to carry out recitations B and C.

**[0118]** It should be emphasized that many variations and modifications may be made to the above-described embodiments, the elements of which are to be understood as being among other acceptable examples. All such modifications and variations are intended to be included herein within the scope of this disclosure.

What is claimed is:

**1**. A method implemented by a vehicle processor system, the method comprising:

obtaining images from a multitude of image sensors positioned about a vehicle;

determining features associated with the images, wherein the features are output based on a forward pass through a machine learning model;

projecting, based on the machine learning model, the features into a vector space associated with a birds-eye view;

aggregating, based on a video module, the projected features with other projected features associated with prior images; and

outputting, based on a plurality of heads of the machine learning model, images depicting static objects in the birds-eye view.

**2**. The method of claim **1**, wherein the birds-eye view represents a top-down view in which static objects are positioned about a location of the vehicle.

**3**. The method of claim **1**, wherein each image sensor is associated with a backbone network, and wherein the backbone network represents a portion of the machine learning model which determines a portion of the features which are associated with an individual image sensor.

**4**. The method of claim **3**, wherein the features are projected into the vector space based on an attention network, and wherein the attention network receives an aggregated input of the features from the backbone networks.

**5**. The method of claim **1**, wherein the video module obtains the projected features and other projected features from a feature queue, and wherein the feature queue spatially indexes information.

**6**. The method of claim **1**, wherein the machine learning model is a hydra network in which a plurality of trunks receives respective portions of output from the video module, and wherein each head is associated with an individual trunk.

**7**. The method of claim **1**, wherein a first image of the output images includes pixels which are assigned values based on likelihoods of the pixels depicting road edges.

**8**. The method of claim **1**, wherein a second image of the output images includes pixels which are assigned respective colors, and wherein each color is indicative of a direction of travel.

**9**. The method of claim **1**, wherein a third image of the output images depicts connections between a lane in which the vehicle is located with one or mother other lanes across an intersection.

**10**. The method of claim **9**, wherein the machine learning model determines splines which connect the lane with the other lanes.

**11**. A system comprising one or more processors and non-transitory computer storage media storing instructions that when executed by the one or more processors, cause the processors to perform operations, wherein the system is included in an autonomous or semi-autonomous vehicle, and wherein the operations comprise:

obtaining images from a multitude of image sensors positioned about a vehicle;

determining features associated with the images, wherein the features are output based on a forward pass through a machine learning model;

projecting, based on the machine learning model, the features into a vector space associated with a birds-eye view;

aggregating, based on a video module, the projected features with other projected features associated with prior images; and

outputting, based on a plurality of heads of the machine learning model, images depicting static objects in the birds-eye view.

**12**. The system of claim **11**, wherein the birds-eye view represents a top-down view in which static objects are positioned about a location of the vehicle.

**13**. The system of claim **11**, wherein each image sensor is associated with a backbone network, and wherein the backbone network represents a portion of the machine learning model which determines a portion of the features which are associated with an individual image sensor.

**14**. The system of claim **13**, wherein the features are projected into the vector space based on an attention network, and wherein the attention network receives an aggregated input of the features from the backbone networks.

**15**. The system of claim **11**, wherein the video module obtains the projected features and other projected features from a feature queue, and wherein the feature queue spatially indexes information.

**16**. The system of claim **11**, wherein a first image of the output images includes pixels which are assigned values based on likelihoods of the pixels depicting road edges.

**17**. The system of claim **11**, wherein a second image of the output images includes pixels which are assigned respective colors, and wherein each color is indicative of a direction of travel.

**18**. The system of claim **11**, wherein a third image of the output images depicts connections between a lane in which the vehicle is located with one or mother other lanes across an intersection.

**19**. The system of claim **18**, wherein the machine learning model determines splines which connect the lane with the other lanes.

**20**. Non-transitory computer storage media storing instructions that when executed by a system of one or more processors which are included in an autonomous or semi-autonomous vehicle, cause the system to perform operations comprising:

obtaining images from a multitude of image sensors positioned about a vehicle;

determining features associated with the images, wherein the features are output based on a forward pass through a machine learning model;

projecting, based on the machine learning model, the features into a vector space associated with a birds-eye view;

aggregating, based on a video module, the projected features with other projected features associated with prior images; and

outputting, based on a plurality of heads of the machine learning model, images depicting static objects in the birds-eye view.

* * * * *