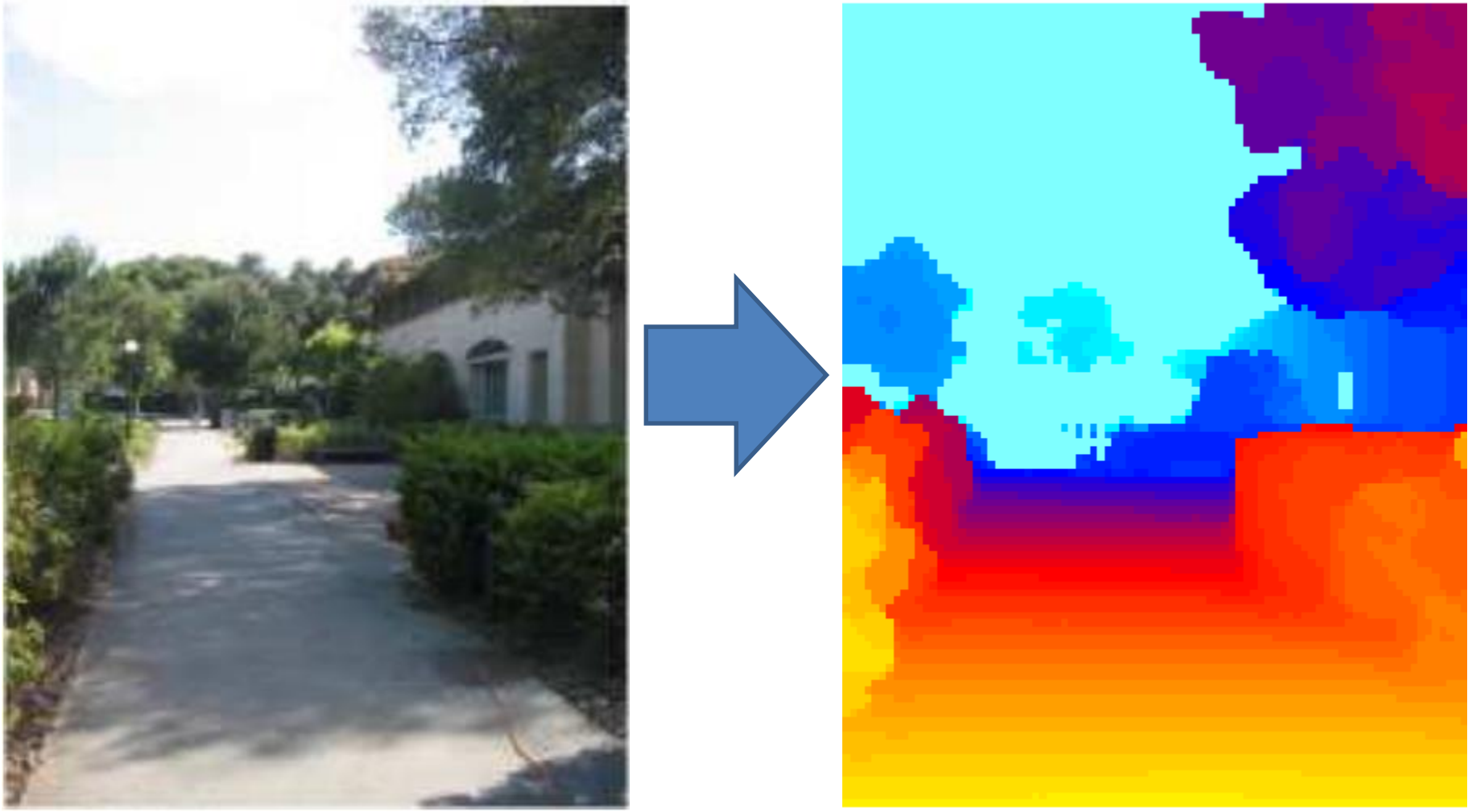
Monocular Depth Estimation

Jianping Fan
Department of Computer Science
UNC-Charlotte

Course Website:
http://webpages.uncc.edu/jfan/itcs5152.html

3-D Depth Reconstruction from a Single Still Image, IJCV



A single still image and its corresponding (ground-truth) depth map. Colors in the depth map indicate estimated distances from the camera.

3-D Depth Reconstruction from a Single Still Image, IJCV

By considering stereo or geometric (triangulation) differences between two images or even multiple images (multiple views), there are several approaches that can be used for depth estimation:

(a) binocular vision (stereopsis) by using two stereo images;

(b) structure from motion ;

(c) depth from defocus.

(d) beyond stereo/triangulation cues, there are also numerous monocular cues—such as texture variations and gradients, defocus, color/haze, etc.—that contain useful and important depth information.

Humans use numerous visual cues to perceive depth. Such cues are typically grouped into **four distinct categories**: monocular, stereo, motion parallax, and focus cues

3-D Depth Reconstruction from a Single Still Image, IJCV

Monocular Cues

Humans use monocular cues such as texture variations, texture gradients, interposition, occlusion, known object sizes, light and shading, haze, defocus, etc.

(a) many objects' texture will look different at different distances from the viewer;

(b) texture gradients, which capture the distribution of the direction of edges, also help to indicate depth. For example, a tiled floor with parallel lines will appear to have tilted lines in an image. The distant patches will have larger variations in the line orientations, and nearby patches with almost parallel lines will have smaller variations in line orientations. Similarly, a grass field when viewed at different distances will have different texture gradient distributions.

© Haze is another depth cue, and is caused by atmospheric light scattering.

(d) overall organization of the image can also be used to determine depths

Stereo Cues

Each eye receives a slightly different view of the world and stereo vision combines the two views to perceive 3-d depth.

An object is projected onto different locations on the two retinae (cameras in the case of a stereo system), depending on the distance of the object.

The retinal (stereo) disparity varies with object distance, and is inversely proportional to the distance of the object. Disparity is typically not an effective cue for estimating small depth variations of objects that are far away.

Motion Parallax

As an observer moves, closer objects appear to move more than further objects. By observing this phenomenon, called motion parallax, one can estimate the relative distances in a scene.

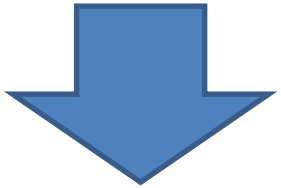
Focus Cues

Humans have the ability to change the focal lengths of the eye lenses by controlling the curvature of lens, thus helping them to focus on objects at different distances. The focus, or accomodation, cue refers to the ability to estimate the distance of an object from known eye lens configuration and the sharpness of the image of the object.

3-D Depth Reconstruction from a Single Still Image, IJCV

Depth estimation from a single still image is a difficult task, since depth typically remains ambiguous given only local image features. Thus the global structure of the image as well as use prior knowledge about the scene must be taken into account.

Modeling depths and relationships between depths at multiple spatial scales by using a hierarchical, multiscale Markov Random Field (MRF)



Modeling the conditional distribution of the depths given the monocular image features. Though learning in our MRF model is approximate, MAP inference is tractable via linear programming.

3-D Depth Reconstruction from a Single Still Image, IJCV

The image is first partitioned into small rectangular patches to estimate a single depth value for each patch.

Two types of features are used: (a) **absolute depth features**—used to estimate the absolute depth at a particular patch; (b) **relative features**, which we use to estimate relative depths (magnitude of the difference in depth between two patches).

These features try to capture two processes in the human visual system: (a) **local feature processing** (absolute features), such as that the sky is far away; and (b) **continuity features** (relative features), a process by which humans understand whether two adjacent patches are physically connected in 3-d and thus have similar depths.

3-D Depth Reconstruction from a Single Still Image, IJCV

Both absolute and relative features are selected to capture **three types of local cues**:

(a) texture variations;

(b) texture gradients;

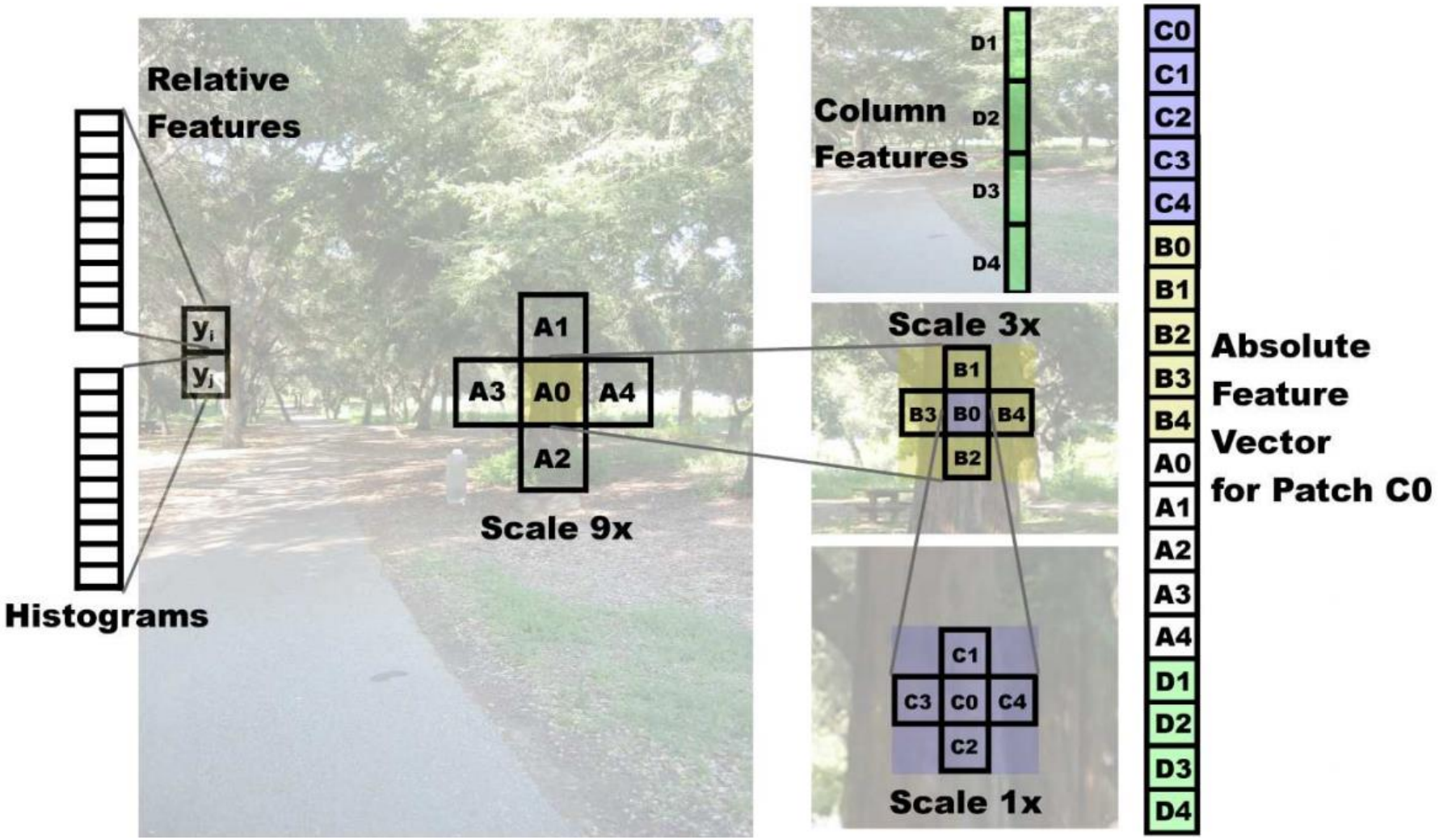
(c) color.

(1)Texture information is mostly contained within the image intensity channel, Laws' masks are applied to this channel to compute the texture.

(2)Haze is reflected in the low frequency information in the color channels, and we capture this by applying a local averaging filter (the first Laws' mask) to the color channels.

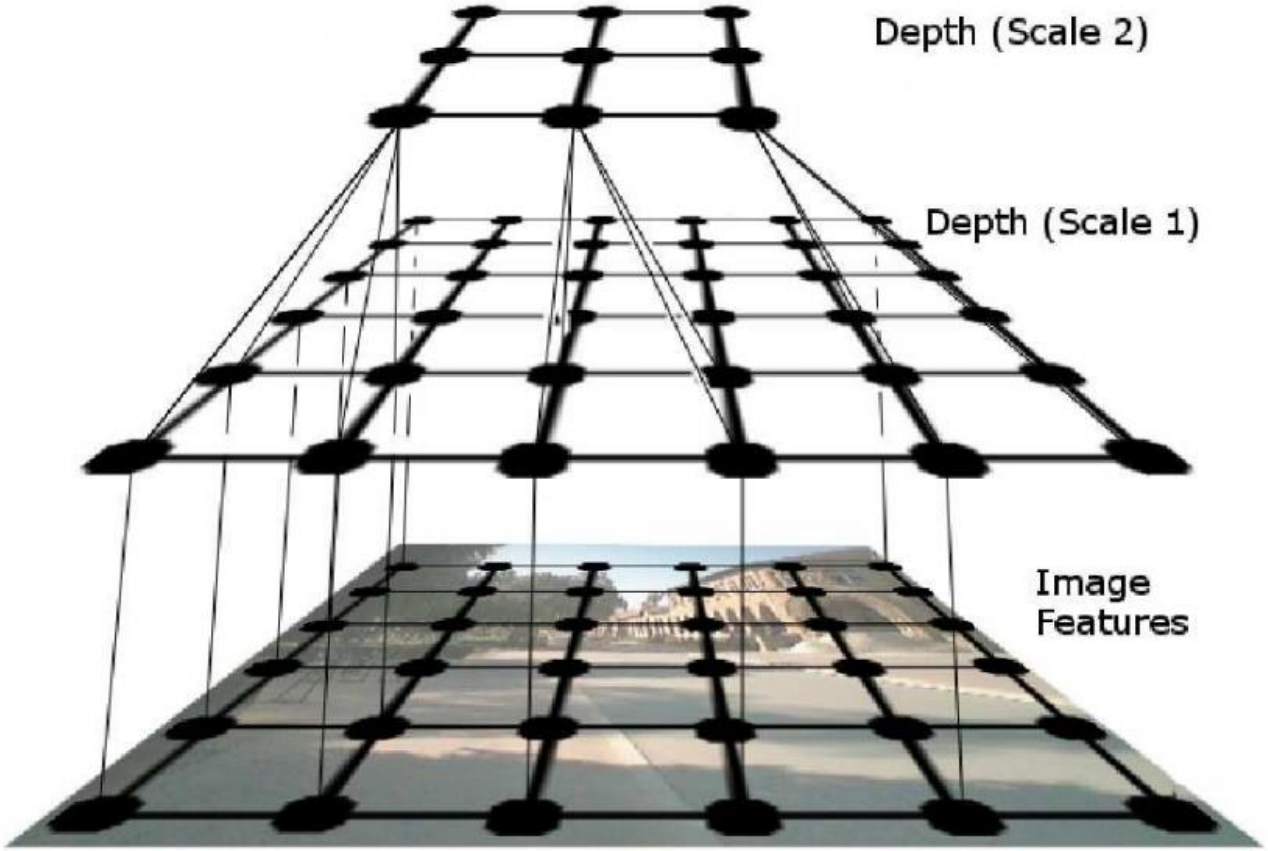
(3)Lastly, to compute an estimate of texture gradient that is robust to noise, we convolve the intensity channel with six oriented edge filters.

3-D Depth Reconstruction from a Single Still Image, IJCV



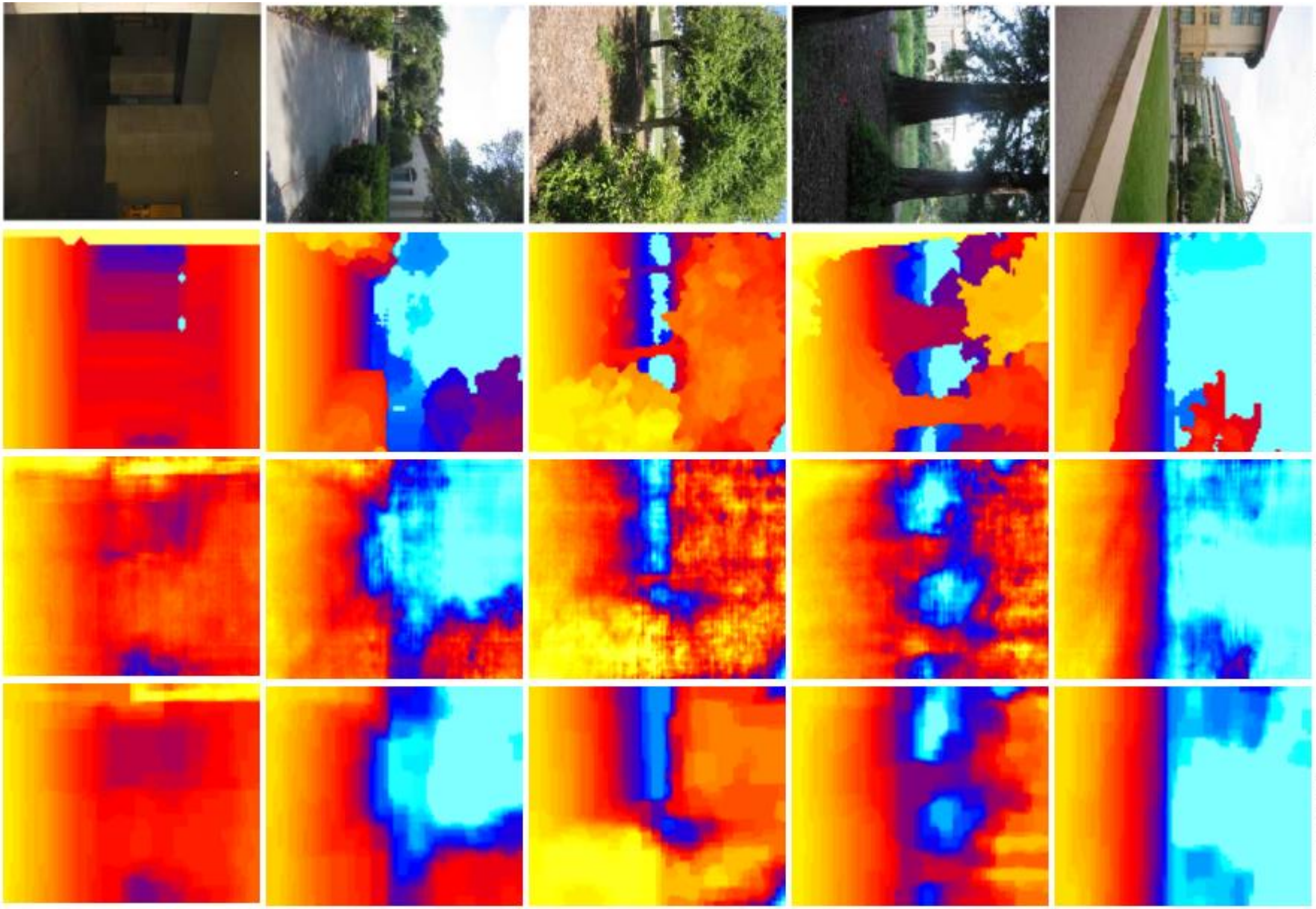
The absolute depth feature vector for a patch, which includes features from its immediate neighbors and its more distant neighbors (at larger scales). The relative depth features for each patch use histograms of the filter outputs.

3-D Depth Reconstruction from a Single Still Image, IJCV



The multiscale MRF model for modeling relation between features and depths, relation between depths at same scale, and relation between depths at different scales. (Only 2 out of 3 scales, and a subset of the edges, are shown.)

3-D Depth Reconstruction from a Single Still Image, IJCV



Depth Map Prediction from a Single Image using a Multi-Scale Deep Network

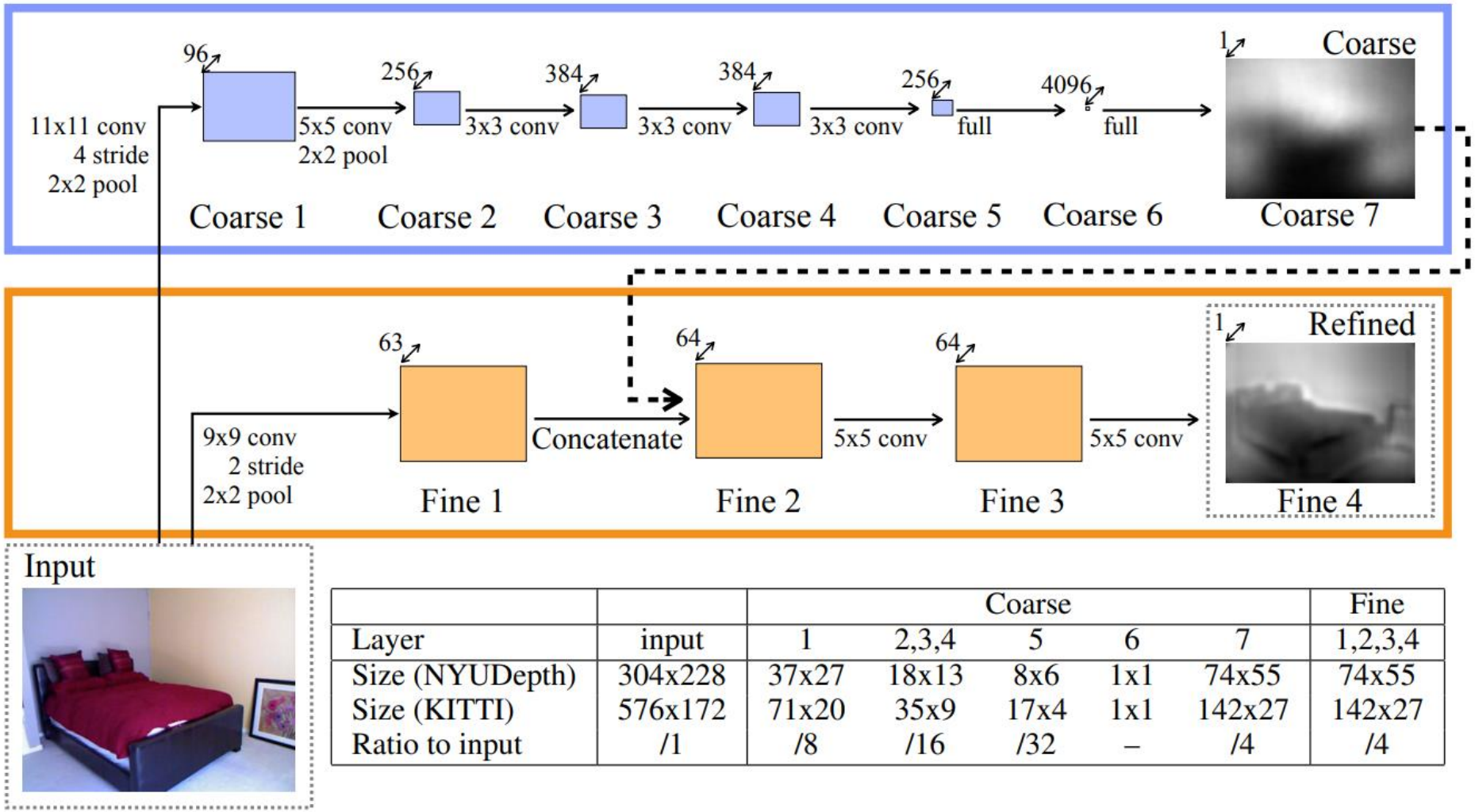
Finding depth relations from a single image is less straightforward, requiring integration of both global and local information from various cues. Moreover, the task is inherently ambiguous, with a large source of uncertainty coming from the overall scale.

Finding depth relations from a single image by employing two deep network stacks:

(a) one that makes a coarse global prediction based on the entire image

(b) another that refines this prediction locally.

Depth Map Prediction from a Single Image using a Multi-Scale Deep Network



		Coarse					Fine
Layer	input	1	2,3,4	5	6	7	1,2,3,4
Size (NYUDepth)	304x228	37x27	18x13	8x6	1x1	74x55	74x55
Size (KITTI)	576x172	71x20	35x9	17x4	1x1	142x27	142x27
Ratio to input	/1	/8	/16	/32	–	/4	/4

A **coarse-scale network** first predicts the depth of the scene at a global level. This is then refined within local regions by a **fine-scale network**. Both stacks are applied to the original input, but in addition, the coarse network's output is passed to the fine network as additional first-layer image features. In this way, the local network can edit the global prediction to incorporate finer-scale details.

Depth Map Prediction from a Single Image using a Multi-Scale Deep Network

The **task of the coarse-scale network** is to predict the overall depth map structure using a global view of the scene. The upper layers of this network are fully connected, and thus contain the entire image in their field of view. Similarly, the lower and middle layers are designed to combine information from different parts of the image through max-pooling operations to a small spatial dimension. In so doing, the network is able to integrate a global understanding of the full scene to predict the depth. Such an understanding is needed in the single-image case to make effective use of cues such as vanishing points, object locations, and room alignment. A local view (as is commonly used for stereo matching) is insufficient to notice important features such as these.

The global, coarse-scale network contains **five feature extraction layers** of convolution and max-pooling, followed by two fully connected layers. The final output is at 1/4-resolution compared to the input (which is itself down-sampled from the original dataset by a factor of 2), and corresponds to a center crop containing most of the input.

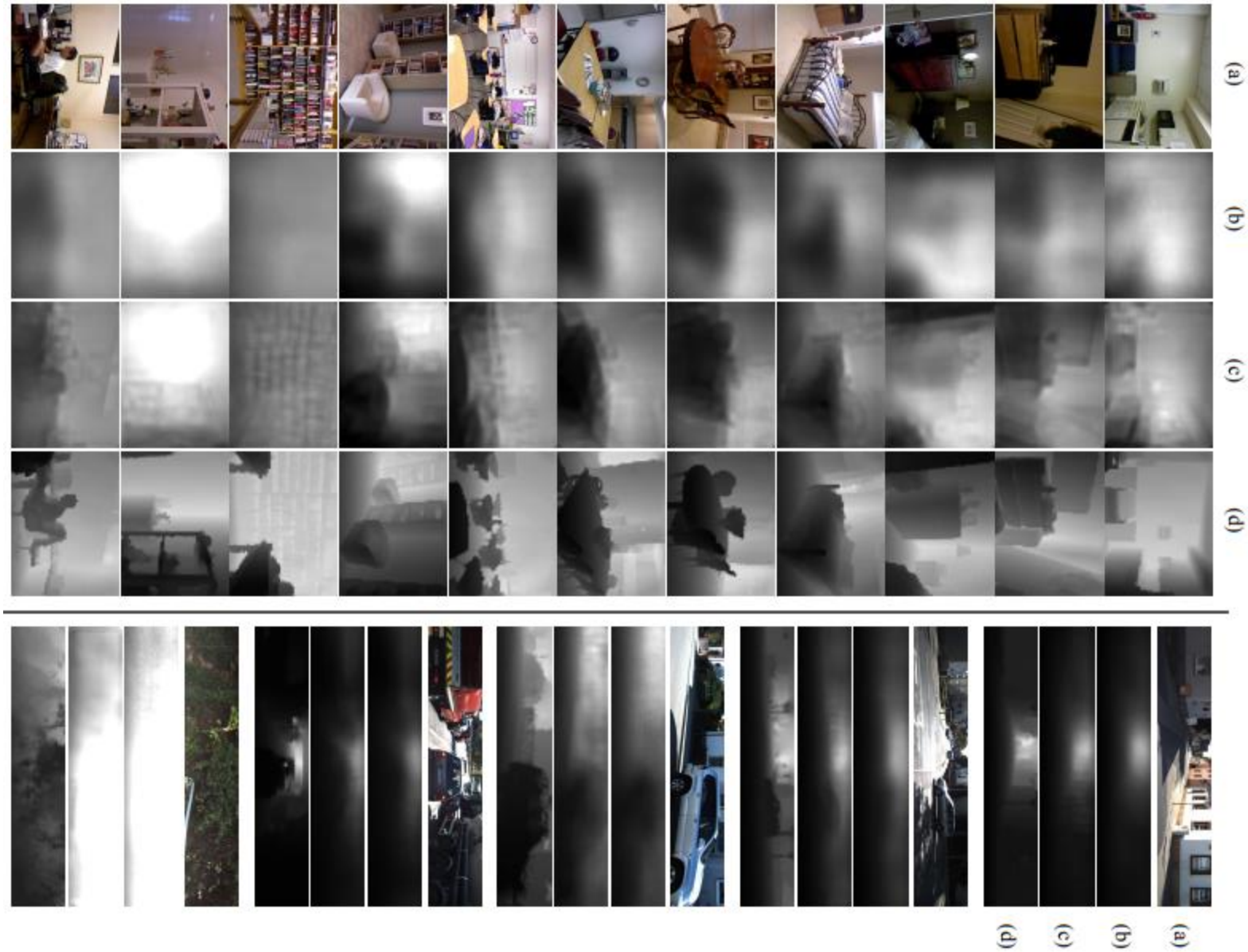
Depth Map Prediction from a Single Image using a Multi-Scale Deep Network

The task of **Local Fine-Scale Network** is to edit the coarse prediction it receives to align with local details such as object and wall edges. The fine-scale network stack consists of convolutional layers only, along with one pooling stage for the first layer edge features.

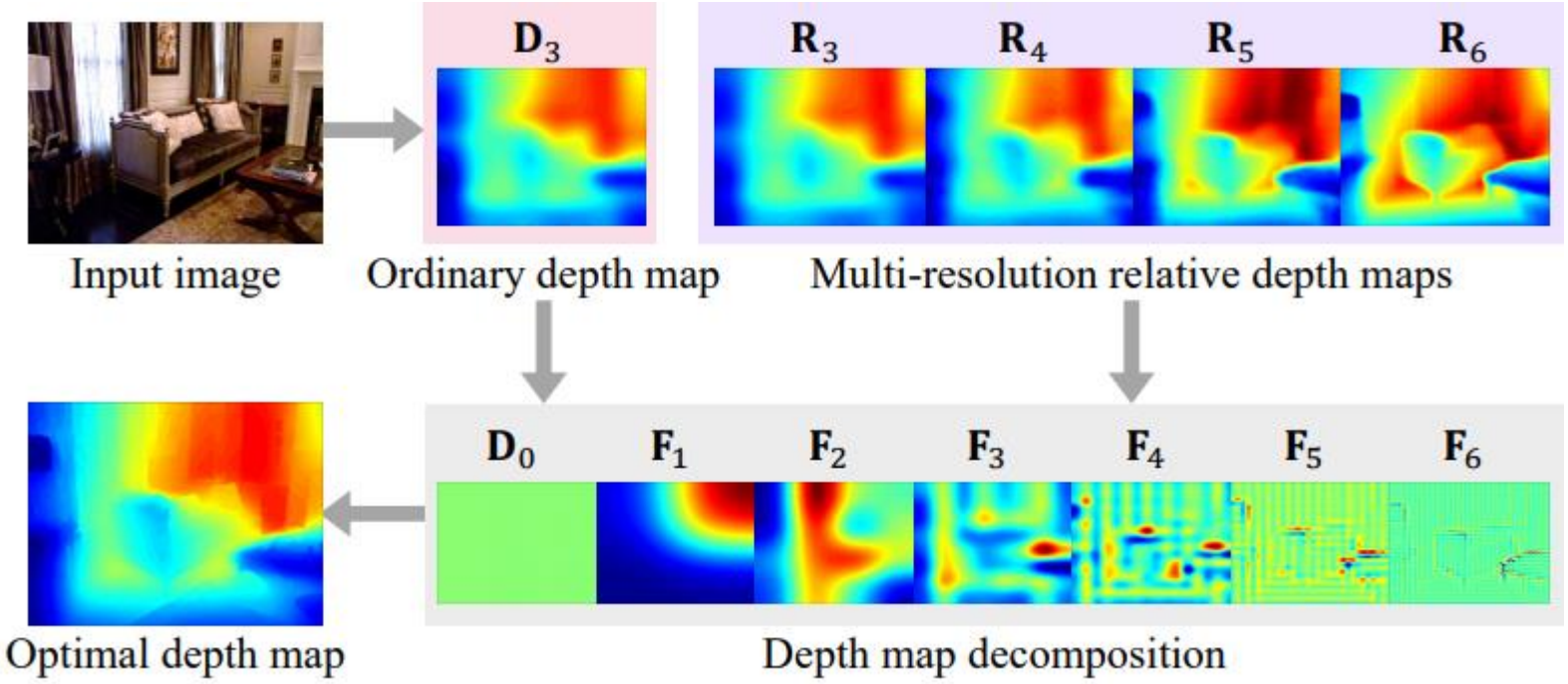
While the coarse network sees the entire scene, the field of view of an output unit in the fine network is 45x45 pixels of input. The convolutional layers are applied across feature maps at the target output size, allowing a relatively high-resolution output at 1/4 the input scale.

More concretely, the coarse output is fed in as an additional low-level feature map. By design, the coarse prediction is the same spatial size as the output of the first fine-scale layer (after pooling). Subsequent layers maintain this size using zero-padded convolutions.

Depth Map Prediction from a Single Image using a Multi-Scale Deep Network



Monocular Depth Estimation Using Relative Depth Maps, CVPR 2019



An overview of the proposed algorithm. First, one ordinary depth map and four relative depth maps are obtained from an image. Then, they are decomposed into depth components, which are, in turn, combined to reconstruct an optimal depth map.

A convolutional neural network is first used to estimate relative depths between pairs of regions, as well as ordinary depths, at various scales. The relative depth maps are then restored from selectively estimated data based on the rank-1 property of pairwise comparison matrices. The ordinary and relative depth maps are decomposed into components and they are recombined optimally to reconstruct a final depth map.

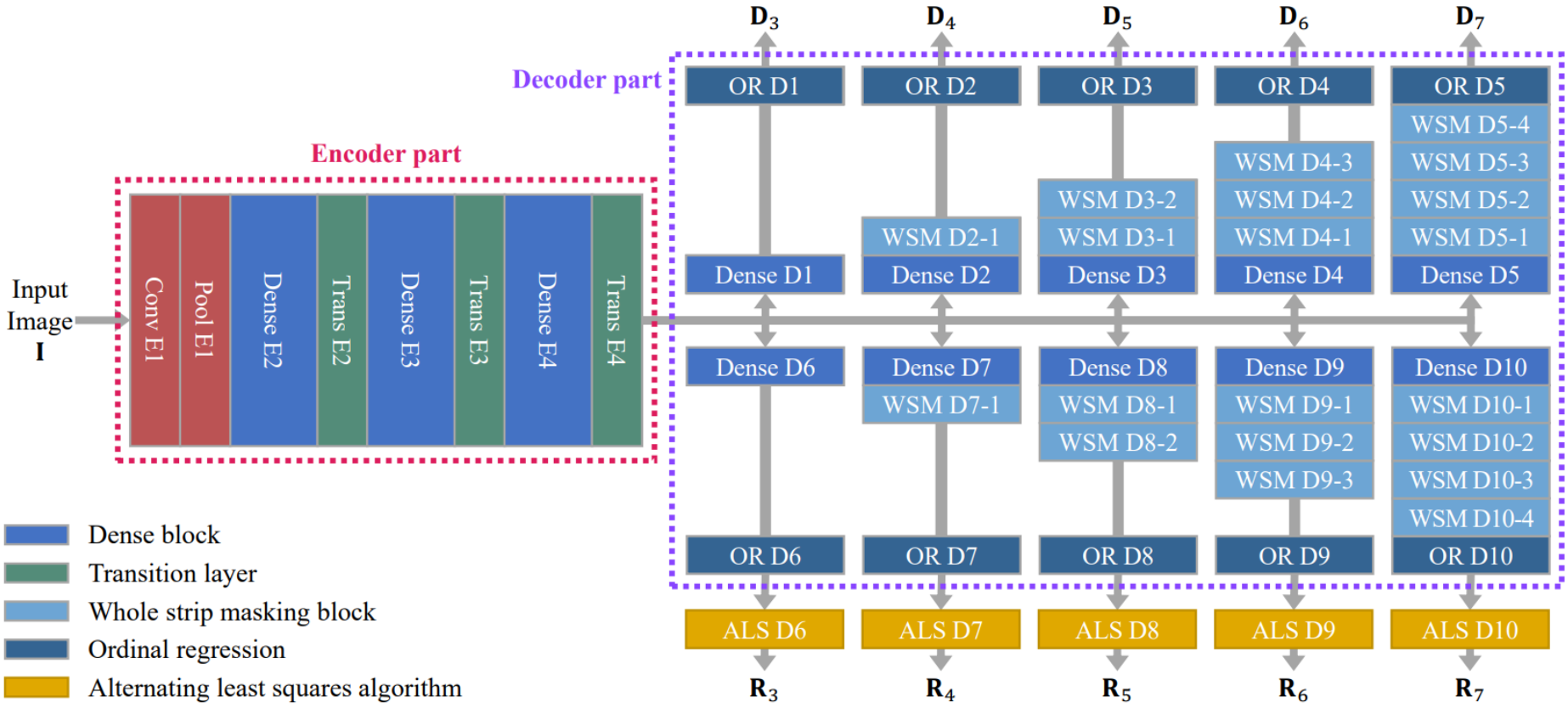
Monocular Depth Estimation Using Relative Depth Maps, CVPR 2019

First, a CNN in the encoder-decoder architecture is developed, which includes multiple decoder blocks for estimating relative depths, as well as ordinary depths, at various scales.

Second, a pairwise comparison matrix is formed, which is sparsely populated by the estimated relative depths. By exploiting the rank-1 property of the matrix, the entire matrix is restored by using the alternating least squares (ALS) algorithm, from which a relative depth map is obtained.

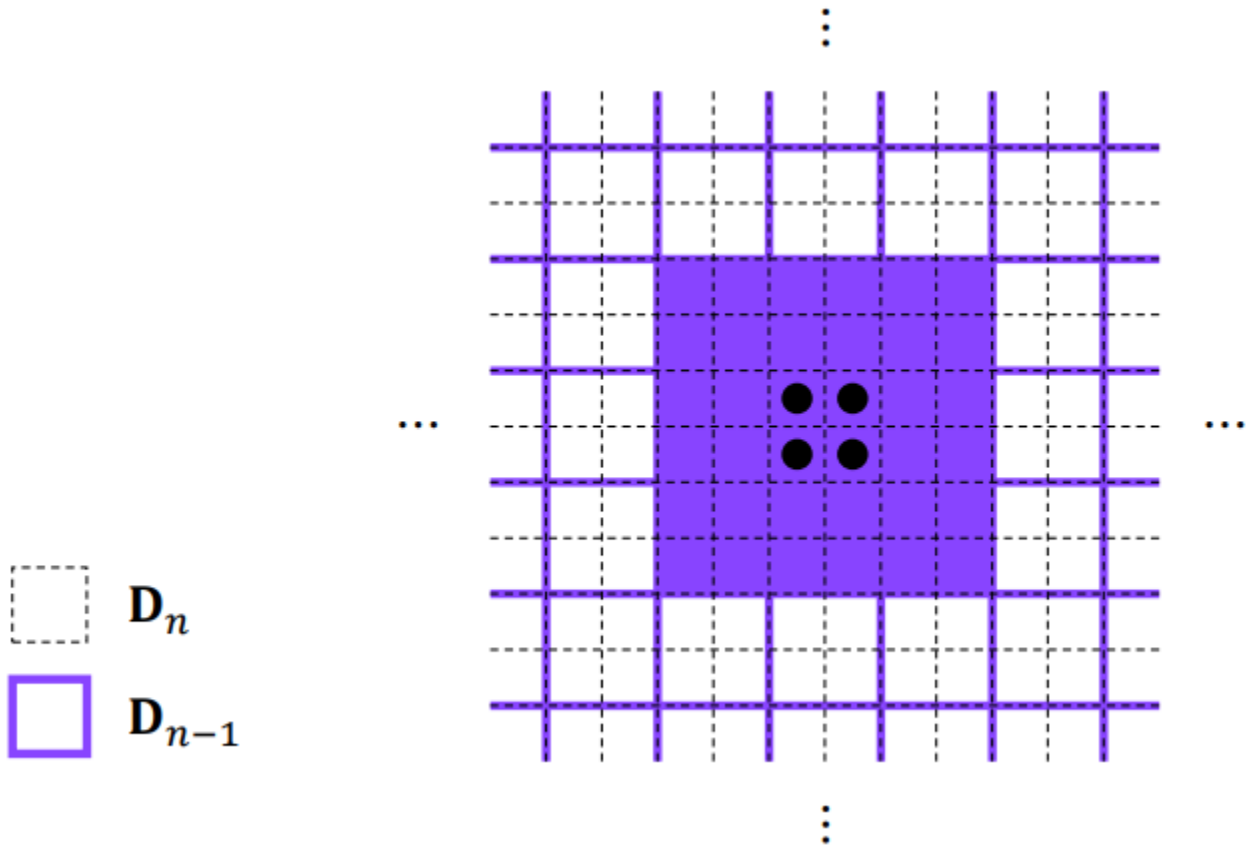
Third, each depth map is decomposed into components, which are re-combined to reconstruct a final depth map through a constrained optimization scheme.

Monocular Depth Estimation Using Relative Depth Maps, CVPR 2019



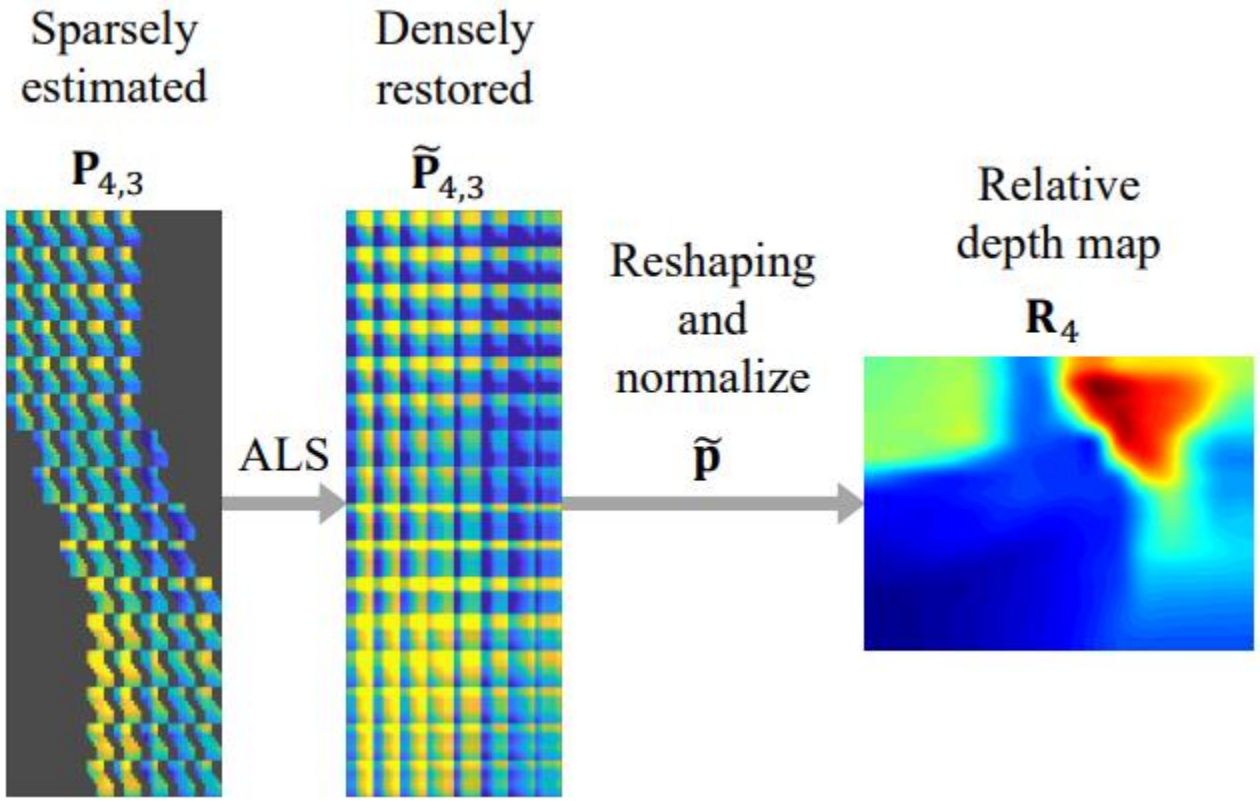
The structure of the proposed depth estimation network. As shown above, up to ten decoders can be used. In the default setting, the five decoders for (D3, R3, R4, R5, R6) are employed. WSM represents a whole strip masking block, OR an ordinal regression layer, and ALS an alternating least squares layer.

Monocular Depth Estimation Using Relative Depth Maps, CVPR 2019



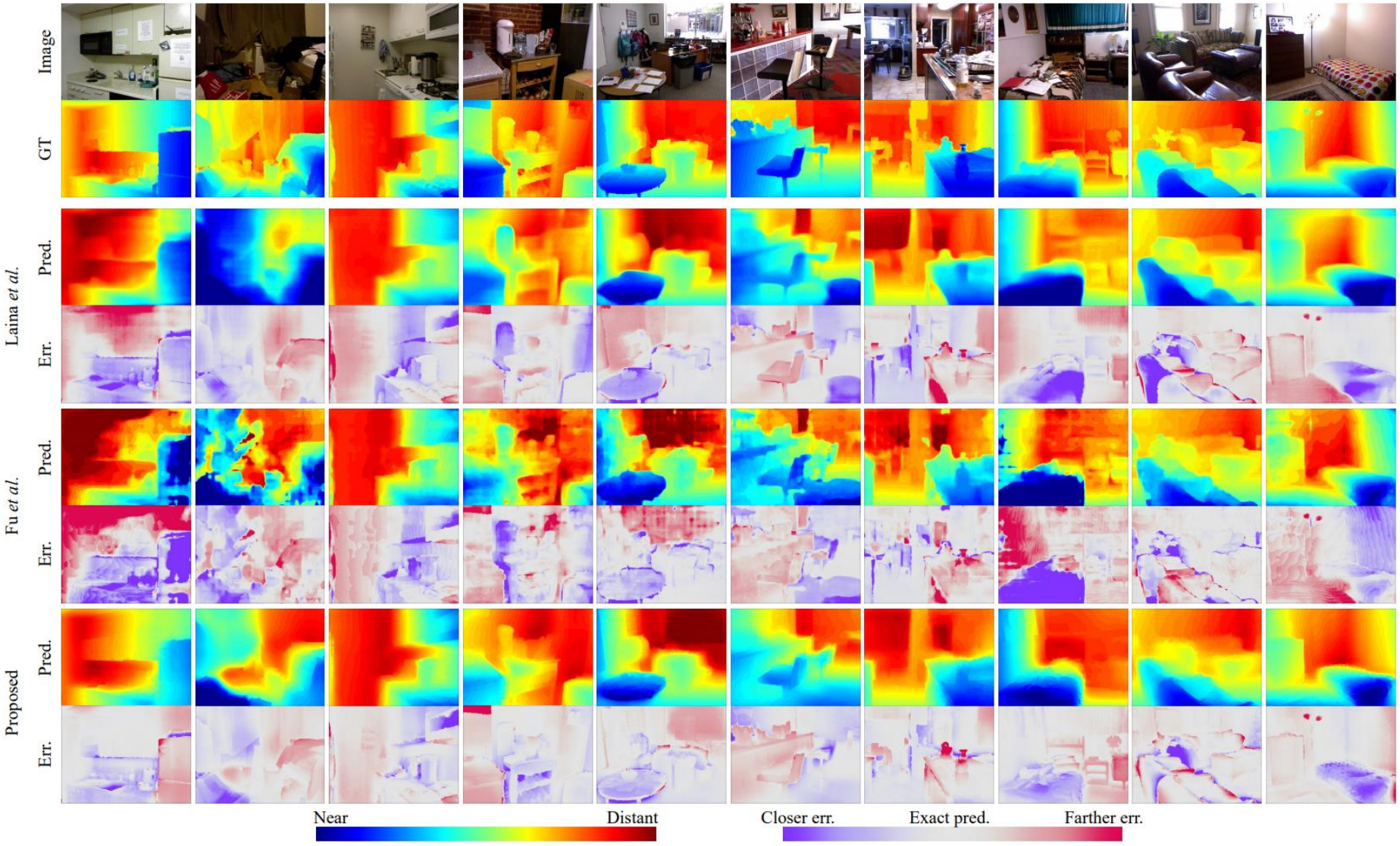
To estimate relative depths, each depth in Dn, depicted by a dot, is compared with the depths of the 3×3 nearest pixels in Dn−1, which are depicted by purple squares. For the illustration, Dn is overlaid with Dn−1.

Monocular Depth Estimation Using Relative Depth Maps, CVPR 2019

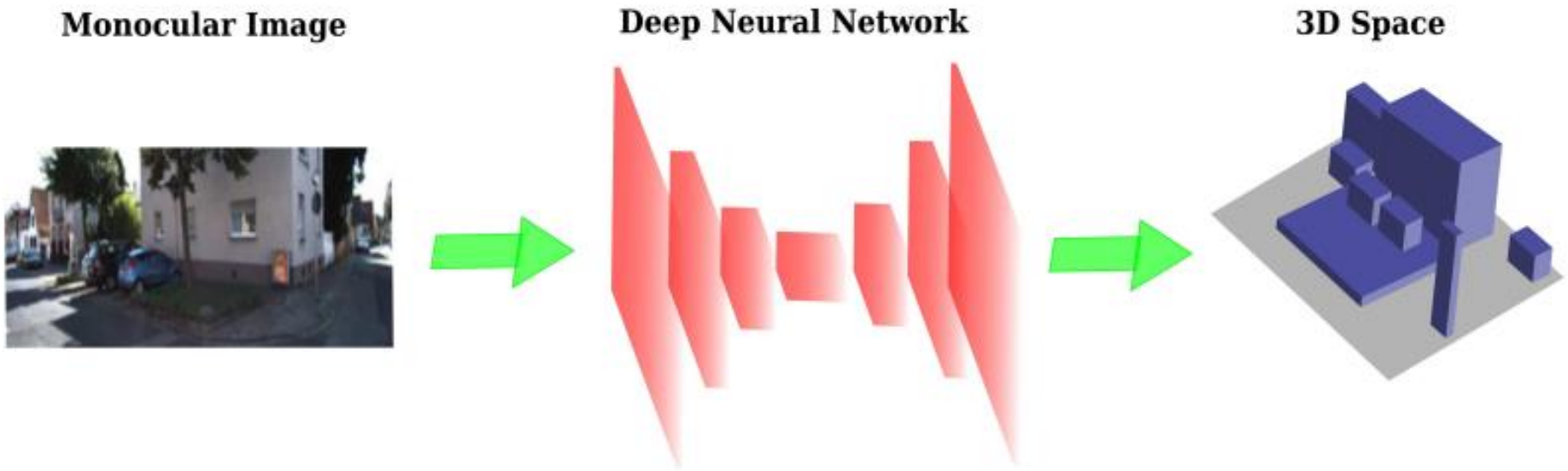


A sparse comparison matrix $\mathbf{P}_{4,3}$ is restored to a dense matrix $\tilde{\mathbf{P}}_{4,3}$ by the ALS algorithm. Then, $\tilde{\mathbf{P}}_{4,3}$ is reshaped and normalized to a relative depth map R4.

Monocular Depth Estimation Using Relative Depth Maps, CVPR 2019

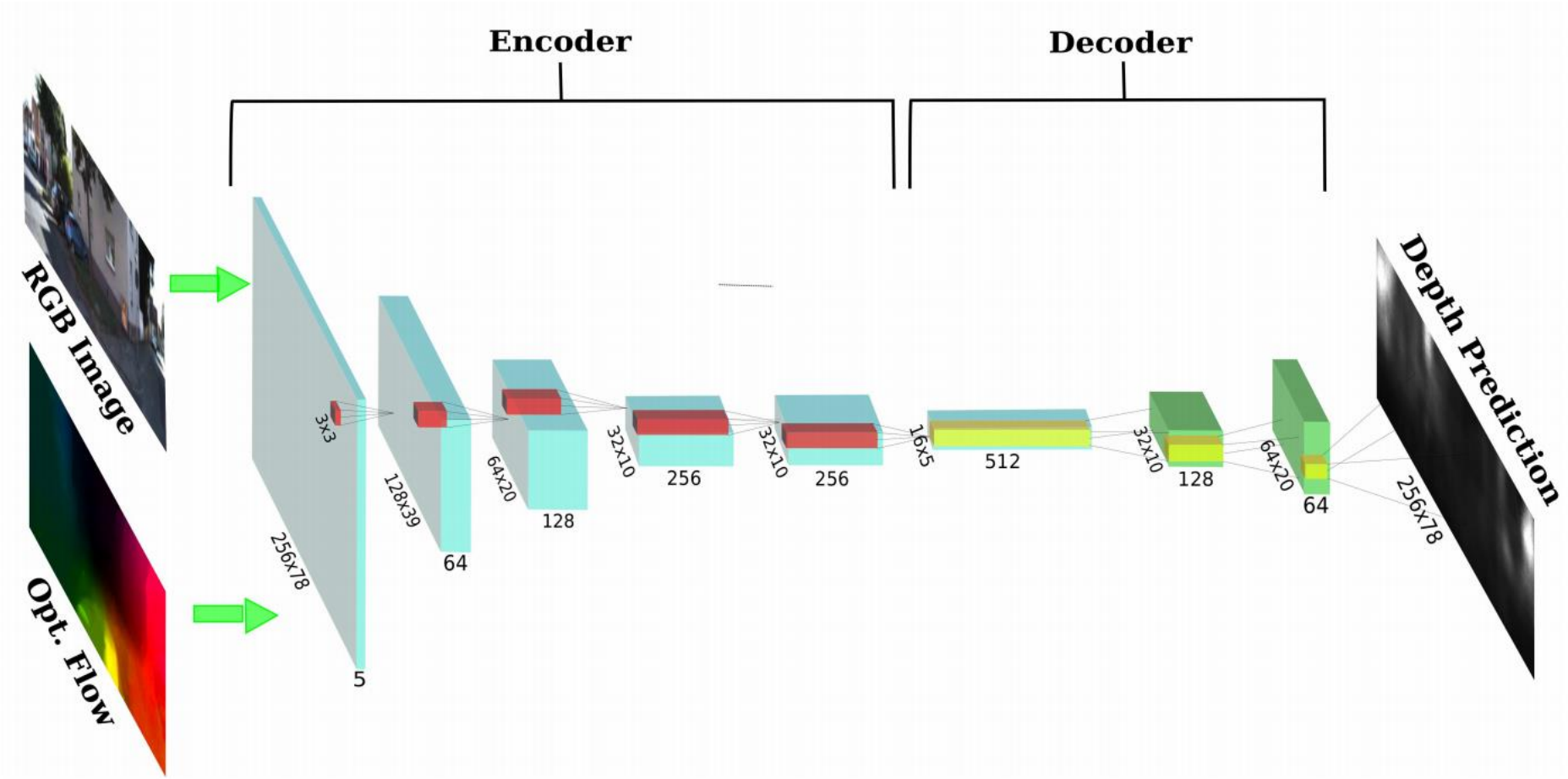


Fast Robust Monocular Depth Estimation for Obstacle Detection with Fully Convolutional Networks, ICRA 2018



We propose a fully convolutional network fed with both images and optical flows to obtain fast and robust depth estimation, with a robotic applications-oriented design.

Fast Robust Monocular Depth Estimation for Obstacle Detection with Fully Convolutional Networks, ICRA 2018



Network architecture. Blue boxes: Encoder feature maps. Green boxes: Decoder feature maps. Convolutional filters are reported in red, deconvolutional filters in yellow.

Fast Robust Monocular Depth Estimation for Obstacle Detection with Fully Convolutional Networks, ICRA 2018

In order to choose appropriate network input, two possible strategies are compared:
(a) feeding the network with a single image, currently captured by the camera;
(b) concatenate current image with optical flow information between current frame and the previous one.

Optical flow has been used previously as raw feature for obstacle detectors. It is known how relative motion information between each pixel in two consecutive frames contains some implicit information about object dimensions and locations in 3D space. As previous works stated, optical flow alone is not sufficient to obtain a complete and long-range depth estimation.

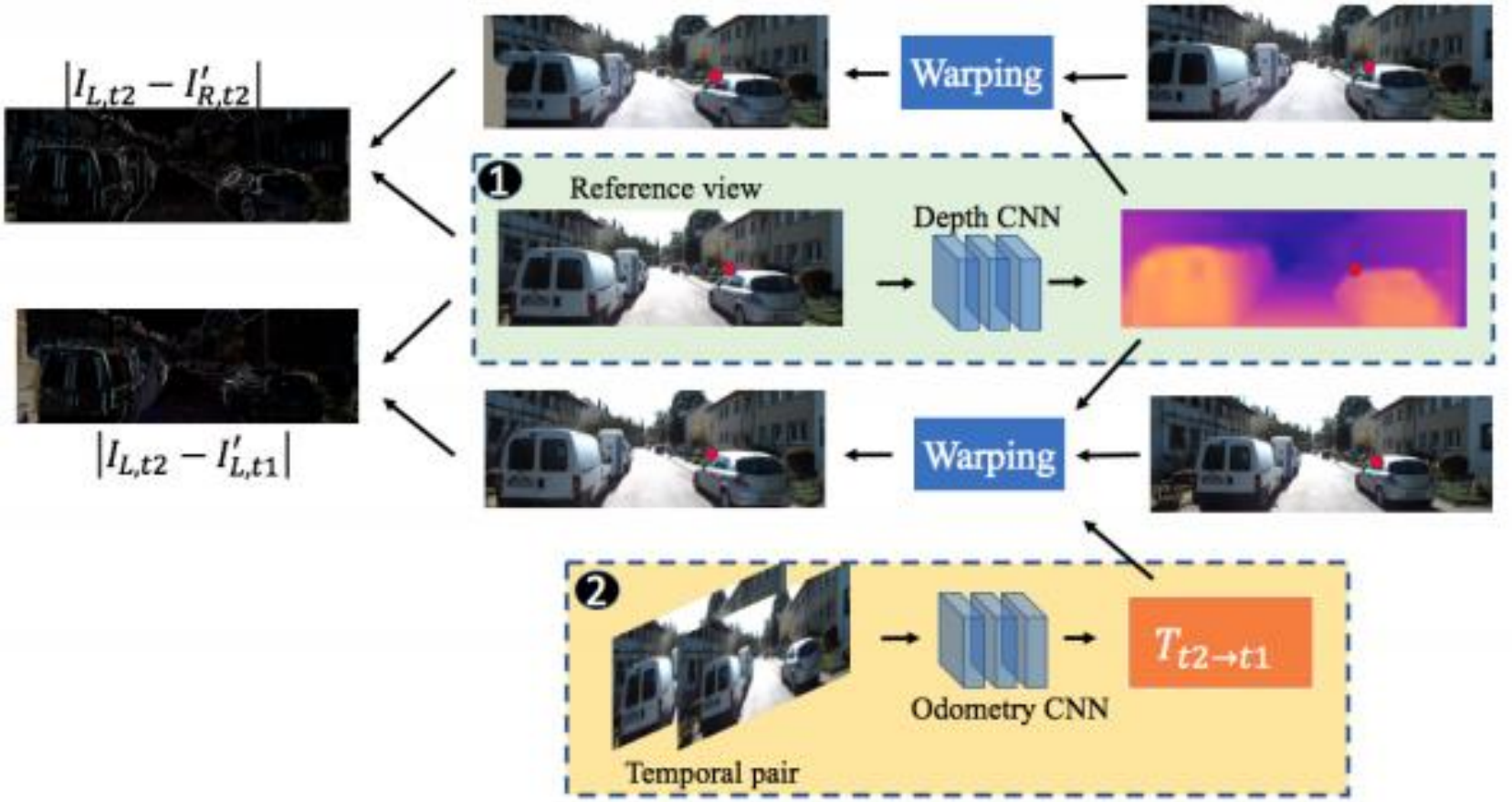
Unsupervised Learning of Monocular Depth Estimation and Visual Odometry with Deep Feature Reconstruction, CVPR 2018

The use of stereo sequences is explored for learning depth and visual odometry. The use of stereo sequences enables the use of both spatial (between left-right pairs) and temporal (forward backward) photometric warp error, and constrains the scene depth and camera motion to be in a common, real-world scale. In addition, a standard photometric warp loss isimproved by considering a warp of deep features.

(i) jointly training for single view depth and visual odometry improves depth prediction because of the additional constraint imposed on depths and achieves competitive results for visual odometry;

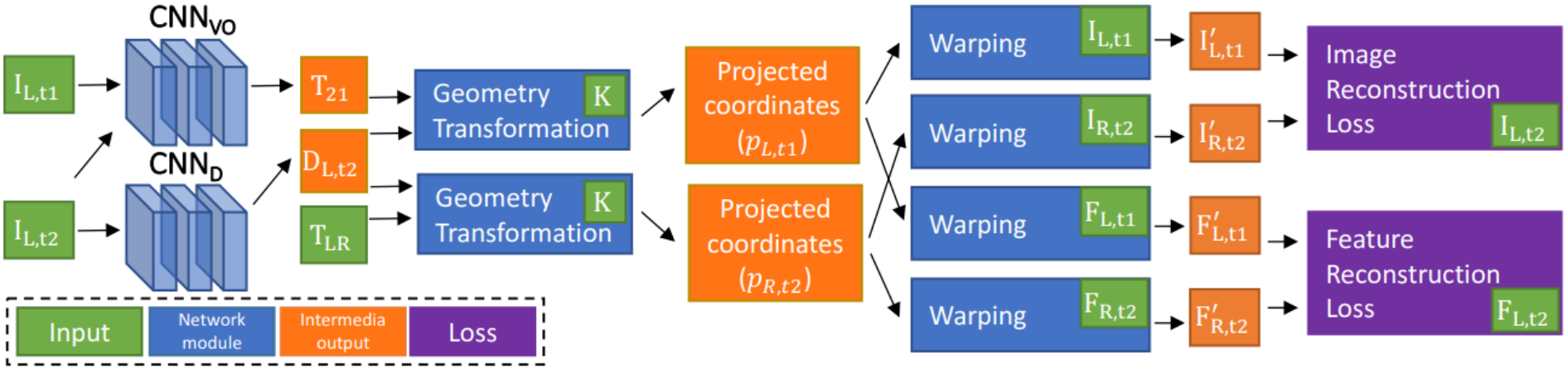
(ii) deep feature-based warping loss improves upon simple photometric warp loss for both single view depth estimation and visual odometry.

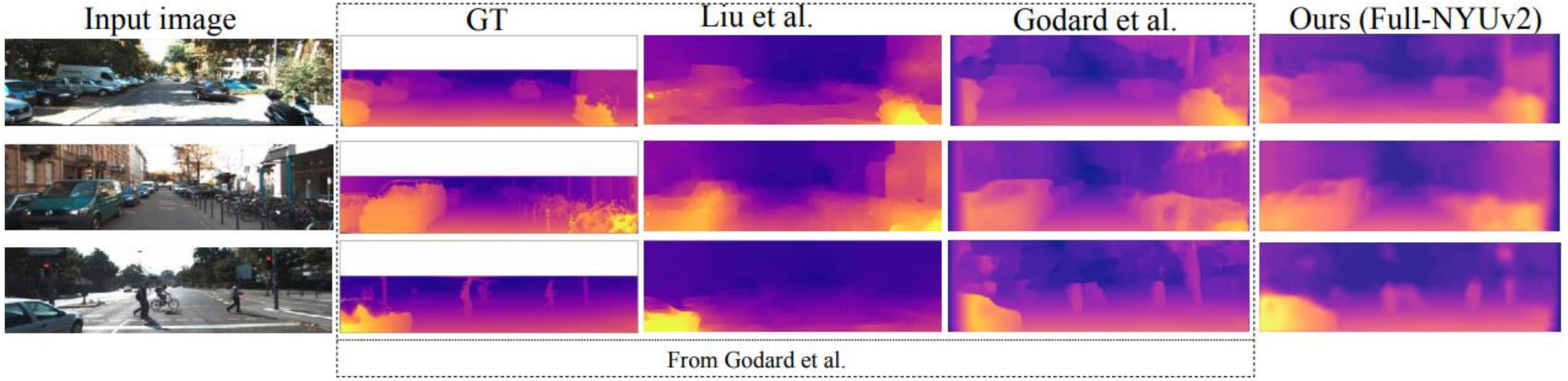
Unsupervised Learning of Monocular Depth Estimation and Visual Odometry with Deep Feature Reconstruction, CVPR 2018



The known camera motion between stereo cameras $T_{L\to R}$ constrains the Depth CNN and Odometry CNN to predict depth and relative camera pose with actual scale.

Unsupervised Learning of Monocular Depth Estimation and Visual Odometry with Deep Feature Reconstruction, CVPR 2018





Learning monocular depth estimation infusing traditional stereo knowledge, ICCV 2019

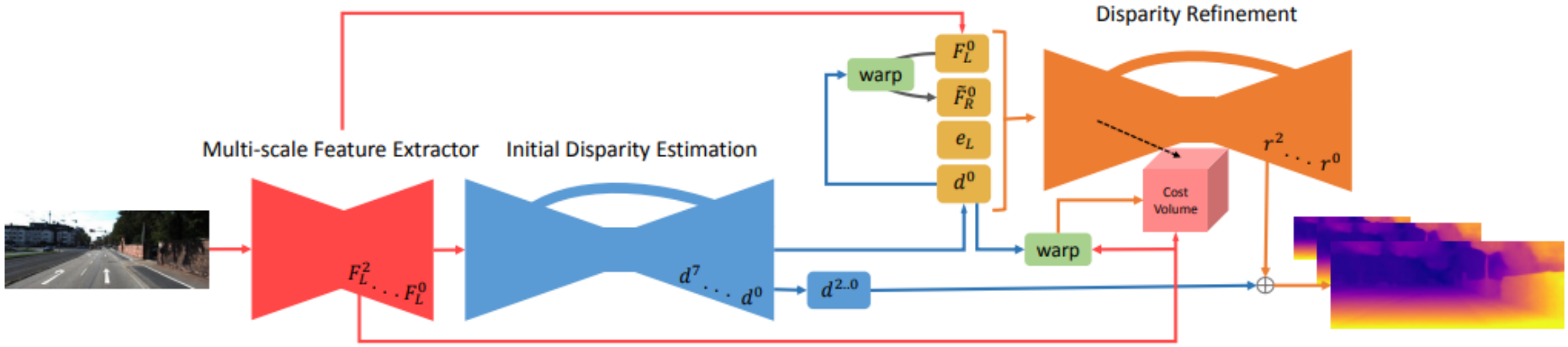
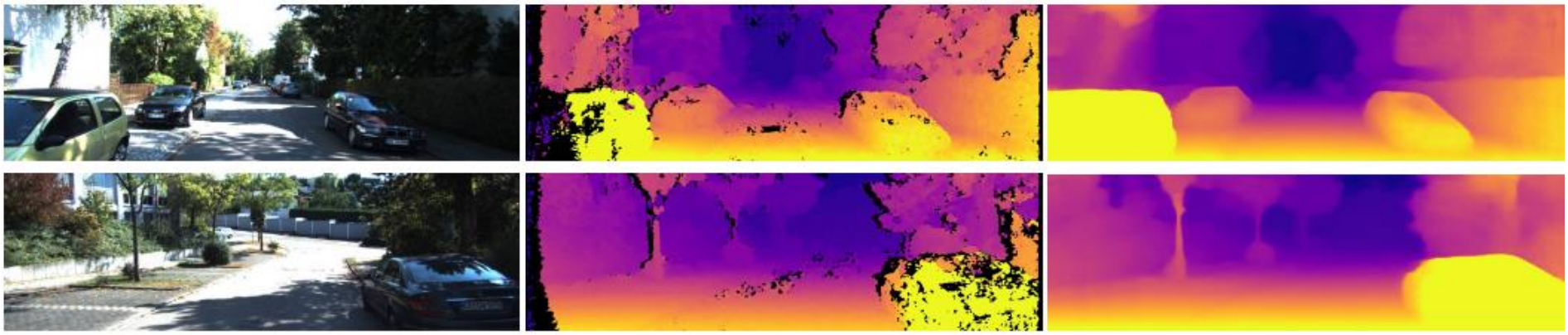


Illustration of our monoResMatch architecture. Given one input image, the multi-scale feature extractor (in red) generates high-level representations in the first stage. The initial disparity estimator (in blue) yields multi-scale disparity maps aligned with the left and right frames of a stereo pair. The disparity refinement module (in orange) is in charge of refining the initial left disparity relying on features computed in the first stage, disparities generated in the second stage

Learning monocular depth estimation infusing traditional stereo knowledge, ICCV 2019

First, a multi-scale feature extractor takes as input a single raw image and computes deep learnable representations at different scales from quarter resolution to full-resolution in order to toughen the network to ambiguities in photometric appearance.
Second, deep high-dimensional features at input image resolution are processed to estimate, through an hourglass structure with skip-connections, multi-scale inverse depth (i.e., disparity) maps aligned with the input and a virtual right view learned during training. By doing so, our network learns to emulate a binocular setup, thus allowing further processing in the stereo domain.
Third, a disparity refinement stage estimates residual corrections to the initial disparity. In particular, we use deep features from the first stage and back-warped features of the virtual right image to construct a cost volume that stores the stereo matching costs using a correlation layer.

Learning monocular depth estimation infusing traditional stereo knowledge, ICCV 2019



(a) (b) (c)

Examples of proxy labels computed by SGM. Given the source image (a), the network exploits the SGM supervision filtered with left-right consistency check (b) in order to train monoResMatch to estimate the final disparity map (c). No post-processing is performed on (c) in this example.