Single-View Height Map Prediction for Satellite Imagery

Abstract

We explore the problem of predicting height maps for man-made and natural terrain when seen from a single image captured by a satellite. This is a very challenging, ill-posed problem, and single-view depth prediction models trained for indoor and outdoor scenes do not immediately apply to the satellite domain. This paper explores this problem for flat and sloped terrain. We propose a baseline model using a deep network that regresses directly from the input image to a height map and is trained using a MAE loss designed specifically for our domain. Predicting the height for images with sloped terrain proves to be a particularly hard challenge. To solve for slopes, we train a separate network to predict dense surface normal maps and combine the surface normals and the predicted height in a global optimization step to improve results on sloped terrain. However, the standard optimization strategy using a constant weight for all pixels in an image introduces a regression on the performance on flat terrain. We propose a novel weight prediction model that predicts per-pixel weights as input for the optimization step. We show that this proposed solution improves height prediction for sloped terrain without regressing on the flat terrain and perform evaluations on a range of loss functions, data sets, fusion strategies and training strategies.

1. Introduction

Timely access to geo-spatially accurate 3D data has become a need for enabling rapid response to events such as disasters and humanitarian crises. 3D reconstruction using satellite imaging can provide this data model. Further, this data can be used for a variety of tasks, including urban planning, landscape and environmental monitoring [13, 29], and change detection [23]. When multiple images are available from overlapping viewpoints, 3D reconstruction can be achieved using triangulation-based stereo methods [7, 13, 9]. However, in many cases only a single view of a region may be available. For example, situations where you want up-to-the-minute 3D or when multiple images are available but only one is useable, for instance due to cloud cover. Such cases can be handled by single-view 3D reconstruction models. Single-view 3D reconstruction also has applications in historical aerial photography where overlapping stereo pairs cannot be acquired. Thus, there is a rise in interest in Single-view 3D reconstruction and it is currently a contest (track #1) in a U.S. government-sponsored data fusion contest [24].

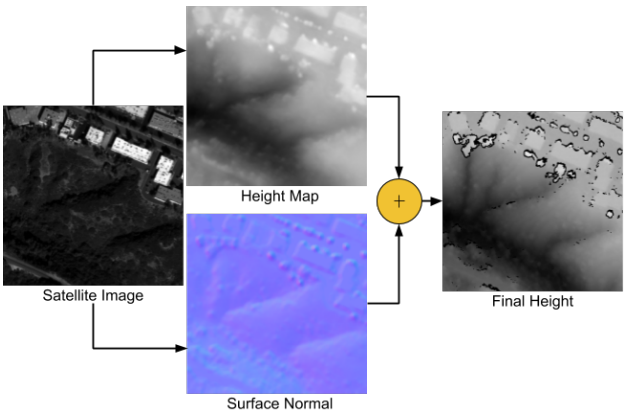

Figure 1. **Height map prediction for satellite images.** The focus of our work is predicting the height of man-made and ground objects given a satellite image. First we train a deep network to predict dense height maps for input satellite images and then, we use a separate network to predict the surface normal map and fuse the height and normal maps to improve on sloped terrain. The gray gradient in the height maps represents the increase in height along the slope.

In this work, we focus on the problem of learning single-view height map prediction for satellite images [27] using convolutional neural networks. The goal is to predict the height of objects relative to a ground plane in the image, which is different from predicting depth (i.e., the distance to the object from the satellite camera). Predicting absolute height of objects on the ground seen in an image is an ill-posed problem. Beyond the mathematical ill-posedness of predicting 2D from 3D, height map prediction for satellite imagery suffers from a translation ambiguity along the axis of the height. This is because the height of the satellite is essentially infinitely larger than the height of the objects on the ground and objects at an arbitrary translation of the ground plane along the height axis (i.e., an arbitrary choice of "sea level") map to the same input image. Accordingly, only relative height differences between the objects are learnable and we use a new translation-invariant loss for training the height regression for this purpose.

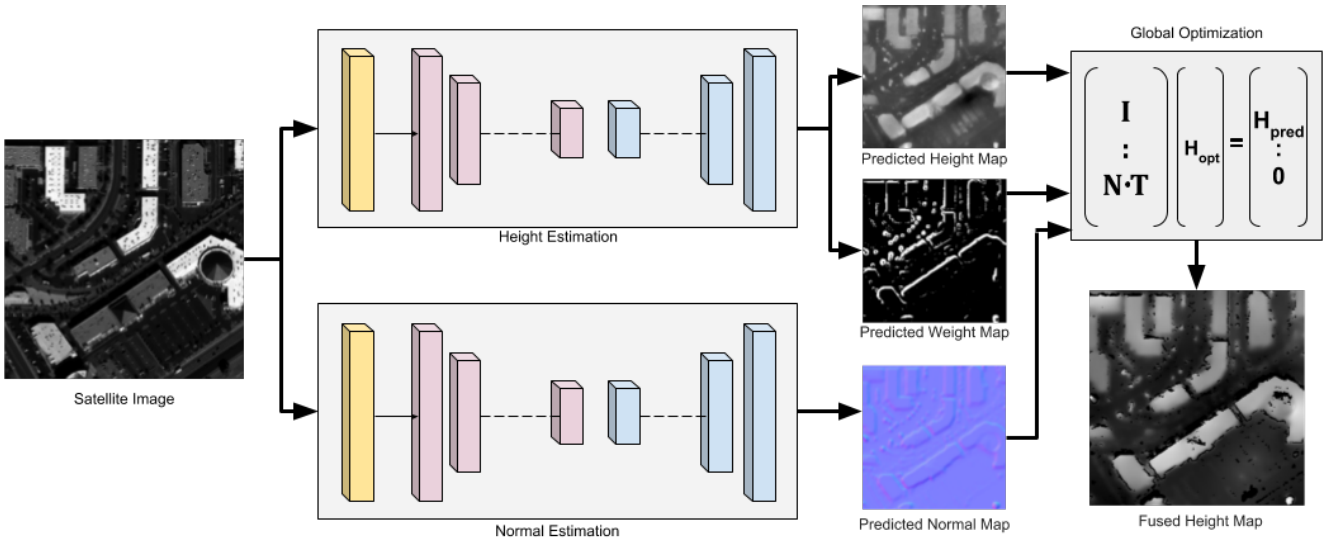

Figure 2. **Overview of our approach.** We learn to predict dense height maps from the input satellite images using a single-stack hourglass network. To improve results of height prediction in sloped terrain, we also predict a dense surface normal map and a dense weight map separate single-stack hourglass networks. These initial height and normal maps are fused together, weighted by the predicted weight map, in a global optimization step formulated as a sparse linear least squares problem.

Predicting height maps for sloped terrain is particularly difficult. Slopes occur where the topography varies as in hilly regions, Figure 1. The height is difficult to predict in part due to the lack of available perspective cues in near-orthographic satellite images. During height map regression, the height of each pixel depends not only on the local surrounding of each pixel, but also on the global terrain. In contrast, a network for predicting dense surface normals at each pixel only needs to leverage the input from a small surrounding region of each pixel to predict its normal. Visual cues for this task include the texture of the sloped surface, shading of the terrain, shape along creases and height discontinuities. Predicted surface normal maps can then be integrated to obtain the per-pixel height map of the input image [19, 30].

We show that fusing both predicted normals and heights in a global optimization post-processing step provides improved performance for height prediction in sloped terrain. However, we find that the hyper-parameter settings that provided the best results on the sloped terrain lead to a regression on images with a flat terrain. In order to overcome this, we propose a novel weight prediction model that predicts dense weight maps which are used as input to the optimization step. The weight prediction model chooses whether to use the normal prediction or the height prediction for each pixel. We show that this strategy improves results on the sloped terrain without regressing on the flat terrain.

To train and evaluate our methods, we generate a new benchmark dataset for single-view height prediction. We explore the use of a highly scalable source of data for this problem: satellite imagery captured with a small disparity between the image pairs for which multi-view stereo (MVS) methods can automatically produce height maps. We generate data from three different sites and from different months of the year to increase diversity in the data. We use this data to define distinct test sets with different characteristics (e.g., sloped vs. flat terrain). We will make the data set publicly available to the community.

In summary, we propose to predict the height map given a single monocular satellite image. Our contributions include:

- a new translation-invariant loss inspired by the scale-invariant loss used for terrestrial depth prediction [6]
- a joint surface normal and height map prediction model that fuses normal and height predictions to yield improved results on sloped terrain
- a novel per-pixel weight prediction model that chooses which of the predicted normal and predicted height to use during the fusion.
- and a new dataset for height prediction using satellite imagery generated using a classical stereo method.

2. Related Work

Single-view depth and normal prediction. A number of methods have been proposed to tackle single-view depth prediction using supervised learning [6, 14, 17, 4], unsupervised learning [12] and synthetic datasets [2]. In addition, many approaches have been proposed for single-view dense surface normal prediction, either independently [28] or jointly with depth and/or semantic labels [15, 5]. These methods are generally trained for prediction in indoor and outdoor scenes, using standard datasets such as the NYU Depth [26], Make3D [25] and KITTI [10]. Generating ground truth train-

ing data for single-view depth prediction using SfM-MVS for terrestrial views has also been proposed [16, 21].

Single-view height prediction. Single-view height prediction for satellite imagery [27, 18] has received much less attention than single-view depth. [18] proposes to tackle height prediction using a residual network with and without skip connections. [27] proposes a multi-task semantic segmentation and single-view height map prediction network. The network is directly learned using an L1 loss on nDSMs [1]. Using nDSM files for ground truth is expensive because they need accurate DTMs which are expensive to acquire. The nDSM used in [27] are generated using a ground versus off-ground classifier [11] that introduces error in the ground truth (F1 score for buildings is 0.909). Additionally, in [27], they train their networks on 11 tiles and test on 5 tiles. In contrast, we use a translation-invariant loss which removes the need nDSMs, propose an approach for predicting on sloped terrain and train on a larger data set of 4,077 tiles of height maps generated using stereo methods across three sites.

Improving depth prediction using normal maps. A number of approaches have been proposed for learning dense surface normals and depths jointly, e.g., using a multi-task network in order to regularize the depth regression task and intrinsically learn the normal maps [15, 5]. Improving depth or normal maps as a post-processing step using the other has been proposed via learning based approaches [22] as well as global optimization [19]. We use the method described in the latter, under the section 'Improve positions using normals', for optimizing the predicted height values. We are also inspired by the fusion of depth and normals proposed by Zhang and Funkhouser [30], although they assume that some known (but sparse) depths are provided as input. In our case, we predict the height maps from scratch.

3. Methodology

In this section, first we introduce our height map regression method and it's associated loss. We then describe our surface normal prediction method, followed by the global optimization step and the per-pixel weight prediction model.

3.1. Height prediction

We are interested in predicting the height of objects on the ground, as seen from a single satellite image, with respect to some arbitrary reference height (i.e., a "ground level" or "sea level") and this differs from the task of single-view depth prediction. The latter involves estimating the depth of the object with respect to the camera, whereas, we assume that the satellite is at an infinitely far, fixed point and hence predicting depth relative to the satellite does not apply.

Predicting absolute height maps from satellite imagery is an ill-posed problem. Beyond the mathematical ill-posedness of recovering 3D from 2D, there is also an absolute height ambiguity. The distance between the satellite and the ground is generally orders of magnitude larger than the height of the typical building or hill. Hence, all terrain within a wide range of background elevation will look nearly indistinguishable to the satellite's camera. We are interested in learning to predict the height of the buildings relative to the height of some (arbitrary) reference ground point. Thus, we introduce the translation-invariant mean absolute error (MAE) metric, which extends the scale-invariant error [6].

The translation-invariant MAE subtracts the mean of the difference of the height of the ground truth and the predicted height when calculating the error of the height prediction for a pixel, *i.e.*, it calculates the L1 error up to a global shift in the estimated height. Mathematically, if h_i is the predicted height and h_i^* the ground truth height for pixel i, the translation-invariant MAE, $D(h, h^*)$, is defined as

$$D(h, h^*) = \frac{1}{N} \sum_{i}^{N} |h_i - h_i^* + \alpha(h, h^*)| \tag{1}$$

where

$$\alpha(h, h^*) = \frac{1}{N} \sum_{i}^{N} (h_i^* - h_i) \tag{2}$$

We use the translation-invariant MAE loss for training our height map prediction network. To encourage smoother gradient changes and sharper height discontinuities, we introduce a translation-invariant gradient loss $Lgrad$, defined as an L1 penalty on differences in height gradients between the predicted and ground truth heights [5, 16]:

$$L_{height} = D(h, h^*) + L_{grad} \tag{3}$$

$$L_{grad} = \frac{1}{N} \sum_{i}^{N} |\nabla_x h_i + \nabla_y h_i| \tag{4}$$

Where $\nabla_x h_i$ and $\nabla_y h_i$ are the horizontal and vertical image gradients of the difference in height.

3.2. Surface normal prediction

We generate the ground truth surface normal maps using the ground truth height map. We first filter the height map using a 25×25 box filter to reduce noise in the generated normal map. Then, for each pixel in the ground truth height map, we estimate the 2D surface normal by fitting a local plane to the 8 neighbouring pixels [26].

Angle-based surface normal loss. We define the loss term L_{normal} in terms of the cosine (dis)-similarity between the predicted and ground-truth normal vectors. Mathematically, if the predicted normal map is denoted as $\mathbf{n}_i^p$ and the ground truth surface normal is denoted as $\mathbf{n}_i^h$ for each pixel i, the

normal loss is defined as:

$$L_{normal} = \frac{1}{N}\sum_{i}^{N}\left(1 - \frac{(\mathbf{n}_i^d \cdot \mathbf{n}_i^p)}{||\mathbf{n}_i^d|| \cdot ||\mathbf{n}_i^p||}\right) \quad (5)$$

Predicting normals. Can we train a network to predict surface normals for satellite images? An interesting property of this domain is that the normal direction for many pixels is perfectly vertical, because the ground is often flat. Hence, a learning model can achieve a low loss by simply learning to predict constant, upwards-facing normal maps, and learning tends to stagnate almost immediately after the first epoch. To encourage the model to predict better normals, we experiment with two other strategies. The first is to train the normal prediction jointly with the height prediction using a multi-task network, using a common single stack hourglass network. The second is to predict the normal maps at two scales, one at the original scale of the input image and the other at a scale 20 times smaller than the original scale. The second strategy is used in order to encourage the model to learn normal maps consistent with global slope in the image.

3.3. Fusing height and surface normals

Using the predicted height h^p and predicted surface normal map $\mathbf{n}^p$, we solve a system of equations to obtain an optimized set of height predictions h^o [19, 30]. The least-squares objective function is the weighted sum of squared errors of the difference between the optimized height and the predicted height, E^h, and the dot product of the tangents along the surface of the optimized height and the predicted normal vectors $E^{\mathbf{n}}$:

$$h^o = \underset{h}{\operatorname{argmin}}\, E^h + \lambda_N^2 E^{\mathbf{n}} \quad (6)$$

where

$$E^h = \sum_i ||h_i^p - h_i||^2 \quad (7)$$

and

$$E^{\mathbf{n}} = \sum_i (||\mathbf{t}_x(h_i) \cdot \mathbf{n}_i^p||^2 + ||\mathbf{t}_y(h_i) \cdot \mathbf{n}_i^p||^2) \quad (8)$$

Here, $\mathbf{t}_x(h_i)$ and $\mathbf{t}_y(h_i)$ denote the x- and y- surface tangents along surface of the optimized height values. We compute $\mathbf{t}_x(h_i)$ and $\mathbf{t}_y(h_i)$ as follows:

$$\mathbf{t}_x(h) = \begin{bmatrix} 1 & 0 & \frac{dh}{dx} \end{bmatrix}^T \qquad \mathbf{t}_y(h) = \begin{bmatrix} 0 & 1 & \frac{dh}{dy} \end{bmatrix}^T \quad (9)$$

This normal error calculation method is an approximation for using the normalized form of the tangent vector when calculating the dot product between normal and tangents [19]. We use this approximation because the normalized form makes the system of equations non-linear and harder to solve. However, this approximation makes the system prone to scaling issues when *only* the normal maps are used. That is, smaller heights result in shorter tangents and smaller $E^{\mathbf{n}}$ terms [30], whereas larger heights lead to tangents with more weight. We do not encounter the scaling issue when we incorporate the predicted height in the system, which maintains the original scale of the predicted height during the optimization.

The linear system is sparse, and we use scipy's implementation of [8] to solve the optimization problem. Through this joint optimization, we are able to leverage predicted normal maps, which provide local normal information useful for predicting slopes and the predicted height estimate, which provides the global context. We refer to the final result output of the system as optimized height predictions.

3.4. Learning dense weights for fusion

In Equation 10, λ_N is a dimensionless hyper-parameter that controls the impact of the different error values on the optimization objective. As the sloped terrain in the input images increases, we observe empirically that the optimal choice for λ_N increases, Figure 6. We find that choosing a λ_N in the range $[80, 160]$ leads to an overall improvement in the height estimates. However, using a constant λ_N for the complete image has its limitations. Using a λ_N in the range $[40, 160]$ leads to a regression in the results on the flat data set, Table 3. Additionally, when an oracle is used to select the best λ_N from the range $[1, 160]$ for the optimization of each data instance, the gain in improvement of height predictions doubles. This motivated us to predict the λ_N values per input image. We predict a λ_N value per-pixel, as different regions of the same image may have different terrains.

We use a single stack hourglass network to predict a per-pixel weight map given only the satellite image as input. The weight map, $W^{\mathbf{n}}$, is used to weight the position error E^h during the optimization step. Each value in the weight map w_i is associated with the i^{th} pixel in the input image and is squashed to between 0 and 1 using a sigmoid operator. We weight the normal error $E^{\mathbf{n}}$ with the complement of the predicted weight map, $1 - W^{\mathbf{n}}$. These pairs of weight maps are used as input weights for the optimization step. The optimized height becomes:

$$h^o = \underset{h}{\operatorname{argmin}}\, W^{\mathbf{n}} * E^h + (1 - W^{\mathbf{n}}) * E^{\mathbf{n}} \quad (10)$$

Penalty-based weight loss. In the optimization step, we use two sources of error. The first is the MSE between the predicted and optimized height, Equation 7 and the second is the consistency error between the tangents and the normals, Equation 8. We train the predicted weight model to "choose" which of these errors it should penalize such that the overall

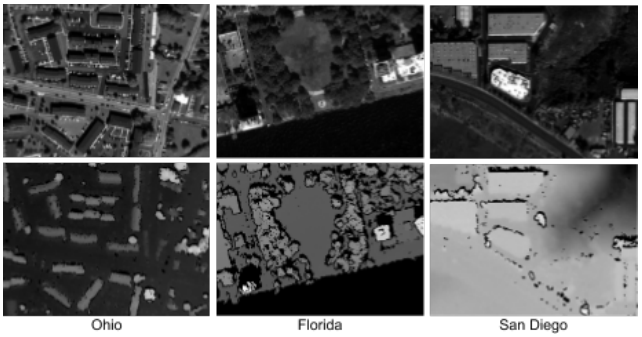

Figure 3. **Data set samples from each site.** The top row is the satellite image and the bottom row is the generated ground truth.

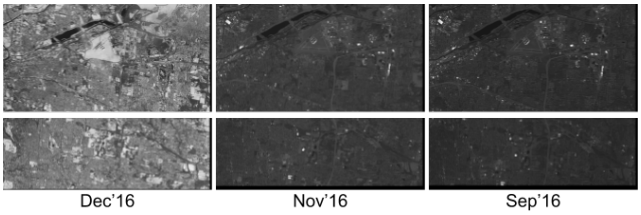

Figure 4. **Split of the Ohio data set collected on three different dates.** The top row is the partition used for training and the bottom row is the partition for testing. The test and the train set do not overlap, even across different dates. An increase in the snow on the surface can be observed from Sep. to Dec. Data collection from different dates adds robustness to weather factors such as snow.

loss decreases. This is equivalent to choosing either the height prediction or the normal prediction for each pixel. The weight training loss, L_{wt} is:

$$L_{wt} = w_i * ||h_i^p - h_i^*||^2 + (1 - w_i) * \sum_{r \in x,y} ||\mathbf{t}_r(h_i^p) \cdot \mathbf{n}_i^p||^2 \tag{11}$$

Note, the model is trained using only the satellite image as input and the predicted height and normals maps are used only during the loss computation. During the normal-tangent consistency error computation for the weight model training, we use the raw normals output by the normal prediction network before they are normalized to unit magnitude. This helps the weight prediction network learn better by re-scaling the consistency error to the same scale as the height error.

4. Dataset

We generate ground truth height maps using the S2P pipeline [7], which uses a stereo method that won the IARPA Multi-View Stereo 3D Mapping Challenge 2016. By using a stereo method to generate ground truth, we can cheaply collect a large amount of diverse training data from images taken from sites in different parts of the world, which can help improve the generalizability of the models.

Input to the S2P pipeline are multiple satellite images of a target site that are slightly spatially displaced with respect to each other. The images are captured at a resolution of 0.31m, using a WorldView-3 sensor and the panchromatic imaging system. The S2P pipeline divides input images into tiles (we use 1000×1000-resolution tiles) and computes a disparity map and height map for pair of neighboring tiles and merges the result into a height map output per tile. Two failure cases for this methods are failure to capture height for bodies of water and failure to capture moving vehicles. For a measure of error margin of the ground truth, we direct the readers to [7].

We generate data from several sites, selecting sets of images taken on the same date for generating ground truth height maps. Each input image has dimensions in the order of 43000×36800 pixels and spans in the order of 14×12 kilometers. We collect the data set from three sites, Figure 3. From a single site, different pairs of images can be generated from different months/years, and thus generate multiple instances of image-target pairs for the same geographic area. This adds robustness to the training as different seasons aids in learning invariance to factors such as weather or lighting. We use two different test sets, one from a site with flat terrain, and another from a site with sloped/hilly terrain and report the performance of our models on each individually.

Florida. The first site is from Florida. The data is obtained for the same site during two different months from two different years, October 2014 and November 2015. The first data collection generated a total of 1,279 tiles and the second generated 1,079 tiles, each of size 1000×1000 pixels. This data set has a high density of water bodies visible in the input image. The stereo methods do not perform well on water bodies and we filter out all images from this data set with more than 35% invalid pixels (as determined by S2P). Further, we use 100% of these images for training (and none for evaluation). After filtering, we obtain a total of 2,076 tiles from this region.

Ohio. The second site is from Ohio. We obtain data for this site on three different dates in the months of September, November, and December 2016. The images from December are covered in snow and add diversity to our dataset. We use 60% of the data (1,546 tiles) for training, 20% for the development set and 20% for the test set. This test set has predominantly flat terrain, and we denote this as the flat test set when reporting performance. The average, maximum and minimum standard deviation in height per image for this test set are 6.25m, 14.79m, 0.23m respectively.

Images from the same site from different dates may not overlap 100% with each other. We use the following strategy to avoid leaking any test data from one date into the training data from another date. The images we obtain are roughly aligned in the $x - y$ axis, where the $y-$axis corresponds to the latitude and the x-axis corresponds to the longitude. We

choose the smallest of the provided input images and partition the image along the latitude such that 60% is assigned to the training set, 20% to the development set and 20% for the test set. The latitude used to split this image is then used to split all the other images from the different dates. As the center of the site is consistent across dates, this split ensures that no train and test overlap occurs Fig. 4. Most images across different dates and sites align with each other. However, a few pairs have a minor rotation between pairs of images, which could cause overlap of about 500 pixels near the edges. Thus, we discard about 2000 pixels between the train and val split along the latitude.

San Diego. The third site is from San Diego, California. We obtain data for this site on a single date. Similar to Ohio, we use 60% of this data (456 tiles) for training, 20% for the development set and 20% for the test set. This test set has predominantly sloped terrain. The average, maximum and minimum standard deviation in height per image for this test set are 9.02m, 21.27m and 0.93m respectively. Hence, we denote this set as the sloped test set when reporting performance.

5. Training details and metrics

For the height, normal and weight prediction networks, we use the hourglass architecture proposed in [20]. Our models contain a single hourglass stack. We use the translation-invariant MAE loss for training the height regression network, the 1-minus-cosine loss for the normal regression network and the penalty-based weight loss for the weight prediction network. The height and normal models are trained with a batch size of 4 on two Nvidia TITAN X GPUs, two batches per GPU, using Adam to optimize the weights and an initial learning rate of 0.01 for 30 epochs. Then, the models are fine-tuned on the San Diego training set for 10 epochs at an initial learning rate of 0.001. The networks outputs height and normal maps at 0.32fps for images of size 1000×1000 on a single TITAN X GPU. For the optimization step, we use $\lambda_N = 140$. This hyper-parameter is selected using the development set and the final results are reported on the test sets. The optimization step takes a variable amount - the time to optimize a 1000×1000 image varies from a few seconds for $\lambda_N = 1$, to a few minutes for $\lambda_N = \infty$. Using $\lambda_N = \infty$ is equivalent to integrating the normal maps.

For multi-task architectures we use the single stack hourglass as the backbone, where each branch has two 1×1 convolution layers. The first is followed by a BatchNorm and a ReLU layer and the second maps the filters to the output dimension. For multi-scale normal prediction, the prediction at the smaller scale is also a separate branch in the network, with an adaptive average pooling between the first and the second convolution layers in the branch.

The weight prediction network is trained only on the development set, as the height and normal predictions on the training set do not represent the performance on an unseen image. The weight network is setup by attaching a branch of three bottleneck layers to an hourglass stack. The hourglass stack is initialized using the weights of the height prediction network. We train the weight model for 15 epochs at an initial learning rate of 1e-4 and 10 epochs at an initial learning rate of 1e-5 using the Adam optimizer.

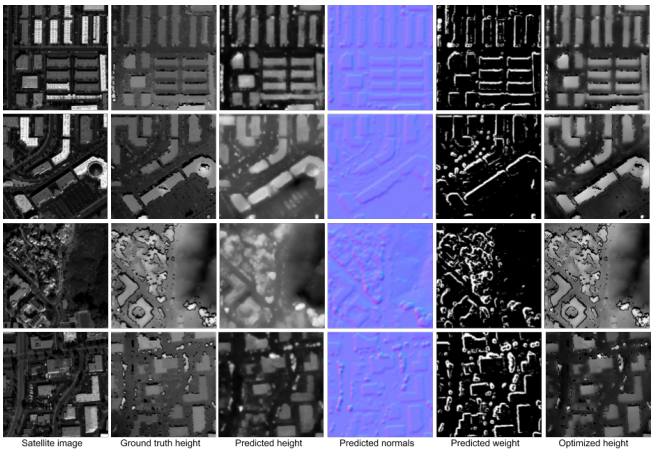

Figure 5. **Qualitative results for height, surface normal and weights.** The optimized height maps are generated by using the dense weight map as input to the optimization step and are overlaid with the nan mask from the ground truth for easy comparison.

Data augmentation. In the domain of aerial imagery, unlike depth prediction in indoor settings, images can be rotated about the Z-axis by up to 360°and still belong to the input domain. During training, we augment the training data with rotation and flips. Per image, the rotation angle is chosen from one of 0°, 90°, 180°or 270°or it can be flipped about the horizontal or vertical axis.

Evaluation metrics. We report the performance of the height model up to a global shift (Section 3.1) using several error measures from prior work:

MAE	$\frac{1}{N}\sum_i^N \lvert h_i^* - h_i^s \rvert$
RMSE	$\sqrt{\frac{1}{N}\sum_i^N (h_i^s - h_i^*)^2}$
Threshold (1,2,3)	% of h_i s.t. $max(\frac{h_i^s}{h_i*}, \frac{h_i*}{h_i^s}) < thr$ $thr \, \epsilon \, \{1.25, 1.25^2, 1.25^3\}$ resp.
Completeness	fraction of h_i s.t. $\lVert h_i^* - h_i^s \rVert < 1.0$
Mean Abs Rel Err	$\frac{1}{N}\sum_i^N \lvert h_i^* - h_i^s \rvert / h_i^*$

where $h_i^s = (h_i + \frac{1}{N}\sum_j (h_j^* - h_j))$, is the shifted predicted height for pixel i. Completeness is the fraction of predicted points whose error is less than 1 meter [3] and is used by the S2P pipeline. We report the cosine distance between the predicted and ground truth normals and the mean angular error for comparing the normal prediction models.

Model Loss/architecture	MAE			RMSE		Thresh-1	Thresh-2	Thresh-3	Completeness		Abs Rel
	Overall	Flat	Sloped	Flat	Sloped	Flat	Flat	Flat	Flat	Sloped	Flat
Direct MAE loss	85.542	66.94	96.144	27.54	31.72	0.8631	0.9660	0.9849	0.061	0.032	0.103
Scale-invariant loss [6]	5.09	3.42	6.76	4.69	8.35	0.9988	0.9997	0.9999	0.263	0.142	0.018
Translation-inv MSE + Multi-task	4.72	2.89	6.56	4.03	8.10	0.9993	0.9999	0.9999	0.312	0.147	0.013
Translation-inv MAE	4.42	2.63	6.22	3.66	7.65	0.9993	0.9999	0.9999	0.372	0.158	0.012
Translation-inv MAE + Multi-task	**4.29**	**2.60**	**5.99**	**3.67**	**7.42**	**0.9993**	**0.9999**	**0.9999**	**0.372**	**0.161**	**0.012**

Table 1. **Performance of height regression models.** The performance is reported on the two test sets together and then individually. All metrics are reported up to a global shift in the predicted height. Translation-invariant MAE loss performs better than other loss functions. Multi-task indicates that a normal prediction branch was also trained.

Model	Cosine distance		Angular error (°)	
	Flat	Sloped	Flat	Sloped
Single scale	0.0049	0.0061	3.71	4.31
Joint training with height	0.0055	0.0068	3.86	4.71
Joint training+multiple scales	0.0055	0.0067	3.90	4.81
Multiple scales	**0.0043**	**0.0056**	**3.35**	**4.09**

Table 2. **Performance of dense surface normal prediction models.** Multi-scale training predicts normals at two scales and improves results on both test sets over the single-scale model.

Dataset	$\lambda_N = 0$	$\lambda_N = 10$	$\lambda_N = 40$	$\lambda_N = 80$	$\lambda_N = 140$	$\lambda_N = \text{inf}$
Flat	**2.48**	2.49	2.53	2.58	2.68	3.74
Sloped	5.86	5.83	5.71	5.55	5.33	**4.81**
Overall	4.17	4.16	4.12	4.07	**4.01**	4.28

Table 3. **Height error on the development set** MAE loss for a range of λ_N values used for global optimization.

Method	MAE			RMSE		Completeness	
	Overall	Flat	Sloped	Flat	Sloped	Flat	Sloped
Height regression	4.29	**2.60**	5.99	3.67	7.42	**0.372**	0.161
Integrating normals	4.31	3.70	**4.92**	4.84	**6.16**	0.225	0.172
Optimization ($\lambda_N = 140$)	**4.07**	2.79	5.35	3.76	6.67	0.327	**0.177**
Optimization - dense weights	4.20	2.60	5.81	**3.63**	7.12	0.371	0.173

Table 4. **Performance of height estimation after global optimization.** The gain on the sloped test set is significant which leads to an overall improvement of the height predictions.

6. Results

We now compare the performance of our methods, first individually for each of the height and normal prediction networks and then the performance of the fused prediction.

Height prediction. For the height prediction task we report the performance of the network trained with our translation-invariant loss and compare it against networks trained with existing loss functions in Table 1. Each experiment uses the same backbone of a single-stack hourglass network. The results are reported on the two test datasets together and separately for finer-grained analysis. The translation-invariant MAE and RMSE improve significantly when the network is trained with our proposed loss. Qualitative results for height prediction using the proposed translation-invariant loss are shown in Figure 5. Multi-task networks that learn to predict the height and normal maps jointly lead to better height estimation models.

Surface normal prediction. The results of our surface normal prediction evaluation are shown in Table 2. The model trained to predict normals at two scales outperforms single-scale training and multi-task training on both the test sets. This indicates that using two scales is beneficial beyond learning the global slope of the image. From results in Table 1, it can be seen that multi-task learning improves height prediction but hurts the performance of surface normal prediction. This suggests that the height estimation loss out-weighs the normal loss during training. We use normals predicted using the multi-scale setting as input for the global optimization.

Fusing height and normals. Table 4 reports the metrics on output heights after global optimization of predicted heights and predicted surface normals (as described in Section 3.3). The errors are reported at the default λ_N value of 140. The accuracy gain for the sloped terrain is significant. This leads to decreased overall error, although the optimization step increases error for the flat test set. Setting $\lambda_N = \infty$ is equivalent to integrating the surface normals (and ignoring input heights), and does not perform well because the output tends to predict extraneous details and slopes for flat terrain. Qualitative examples of predicted height, surface normal maps, and optimized height maps are shown in Figure 6.

Learning dense weights for fusion. The results of the weight prediction model are shown in Table 5. The performance is reported on the height error and the consistency error of the normals and the tangents when using the learned weight maps, section 3.4. The initial error is computed with equal weights for both errors. Table 4 reports the performance of the global optimization using dense weights predicted using the weight network. The model using the dense weight maps improves results on the sloped data set

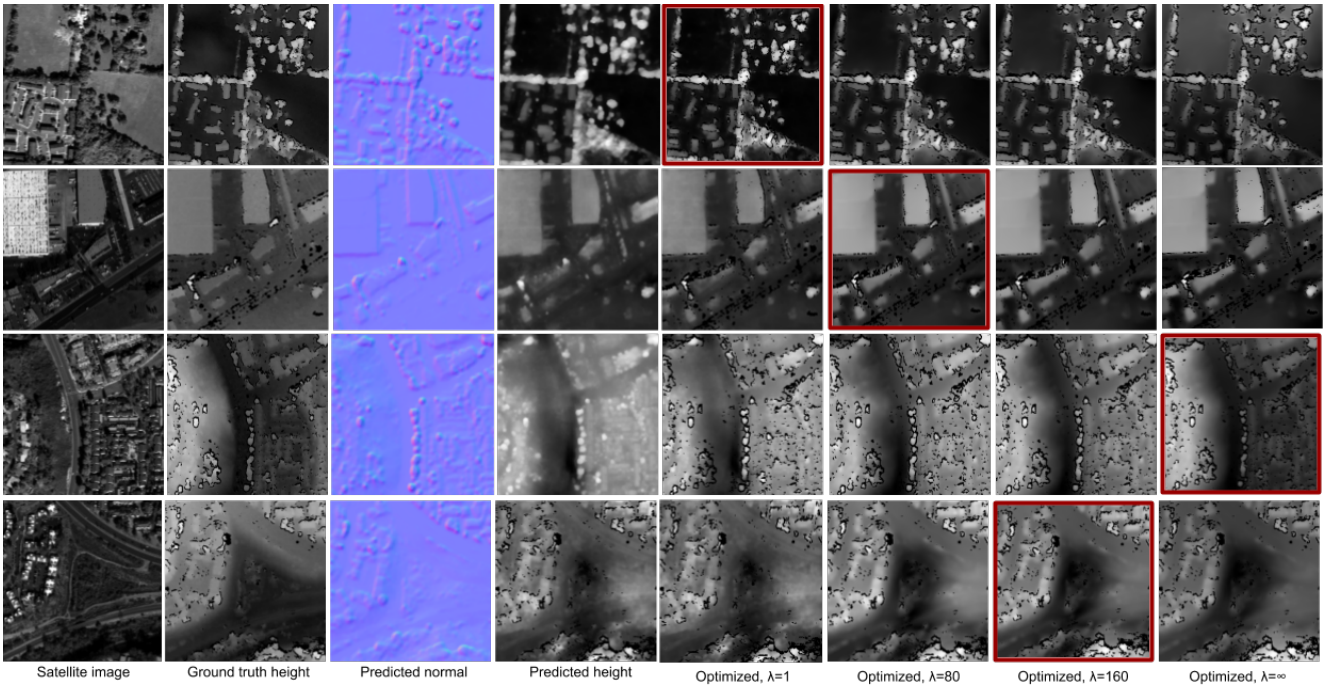

Figure 6. **Global optimization results with different λ_N values.** For each row, the best performing λ_N is outlined in red. We use a fixed $\lambda_N = 140$ for the optimization. The valid pixel mask of the ground-truth is overlaid on results for better comparison with the ground truth.

Data set	Initial Consistency Error	Initial MSE	Weighted Consistency Error	Weighed MSE
Flat	10.87	10.34	2.71	3.17
Sloped	5.78	41.67	3.33	5.04

Table 5. **Results of weight prediction model.** The weighted height error and the consistency error described in 3.4. The initial errors are computed with equal weights for both errors.

without regressing on the flat data set, even improving on the RMSE metric for the flat terrain and thus, validating this approach.

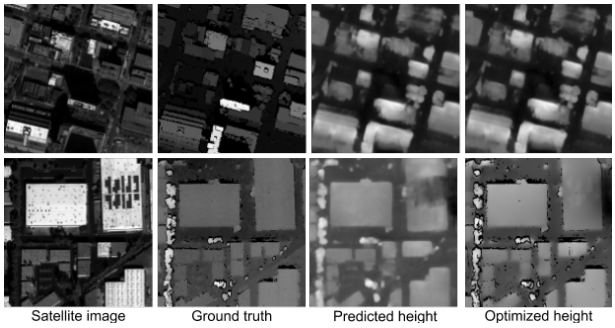

Figure 7. **Failure modes.** The two whitest buildings in the ground-truth in the top row are skyscrapers. The model is unable to predict that their height is much higher than surrounding buildings. The model also fails to estimate correct height for large roofs with a surface resembling the ground on top of the building (top-right of the image in the bottom row).

Failure modes of height prediction. Two failure modes are shown in Figure 7. The first shows the model is unable to predict that the skyscraper is taller than its surrounding buildings. We attribute this to paucity of visual cues for predicting relative heights between buildings at nadir. Cues such as cast shadows and thinner haze over a tall building can be exploited to address this. The second failure case occurs when the surface on the top of a building resembles the ground, *e.g.*, a dark surface or a garden. The height for such pixels is estimated to be at ground level. The optimization result mitigates this failure to a certain extent.

7. Conclusion

In this work, we propose a deep learning framework for estimating the height of objects seen from a gray-scale satellite image. This work has the following contributions. First, we propose to learn the direct height using a new translation-invariant loss and show that this loss performs better than existing loss functions. Second, we propose predicting the surface normals using the input image and optimizing the predicted height estimates using the normals. We show that optimization improves results for sloped terrain. Third, we propose a novel weight prediction model that predicts per-pixel weights as input for the optimization step and show that this solution improves height prediction for sloped terrain without regressing on the flat terrain. Finally, we provide a new benchmark dataset generated using existing stereo methods for single-view height estimation.

References

[1] Isprs2d-vaihingen. http://www2.isprs.org/commissions/comm3/wg4/2d-sem-label-vaihingen.html. Accessed: 2018-11-09. 3

[2] A. Atapour-Abarghouei and T. Breckon. Real-time monocular depth estimation using synthetic data with domain adaptation. *CVPR*, 2018. 2

[3] M. Bosch, Z. Kurtz, S. Hagstrom, and M. Brown. A multiple view stereo benchmark for satellite imagery. *IEEE Applied Imagery Pattern Recognition Workshop*, 2016. 6

[4] W. Chen, D. Xiang, and J. Deng. Surface normals in the wild. *ICCV*, 2017. 2

[5] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. *ICCV*, 2015. 2, 3

[6] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. *NIPS*, 2014. 2, 3, 7

[7] G. Facciolo, C. D. Franchis, and E. Meinhardt-Llopis. Automatic 3d reconstruction from multi-date satellite images. *CVPRW*, 2017. 1, 5

[8] D. C.-L. Fong and M. Saunders. Lsmr: An iterative algorithm for sparse least-squares problems. *SIAM Journal on Scientific Computing*, 33(5):2950–2971, Oct. 2011. 4

[9] Y. Furukawa and C. Hernández. Multi-view stereo: A tutorial. *Foundations and Trends in Computer Graphics and Vision*, 9(1-2):1–148, June 2015. 1

[10] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. *CVPR*, 2012. 2

[11] M. Gerke. Use of the stair vision library within the isprs 2d semantic labeling benchmark (vaihingen), 2015. 3

[12] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. *CVPR*, 2017. 2

[13] G. Kuschk. Large Scale Urban Reconstruction from Remote Sensing Imagery. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2013. 1

[14] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional-residual networks. *3DV*, 2016. 2

[15] B. Li, C. Shen, Y. Dai, A. van den Hengel, and M. He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. *CVPR*, 2015. 2, 3

[16] Z. Li and N. Snavely. Megadepth: Learning single-view depth prediction from internet photos. *CVPR*, 2018. 3

[17] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. *CVPR*, 2015. 2

[18] L. Mou and X. X. Zhu. IM2HEIGHT: height estimation from single monocular imagery via fully residual convolutional-deconvolutional network. *arXiv*, 2018. 3

[19] D. Nehab, S. Rusinkiewicz, J. Davis, and R. Ramamoorthi. Efficiently combining positions and normals for precise 3d geometry. *ACM Transactions on Graphics*, 24(3):536–543, July 2005. 2, 3, 4

[20] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. *ECCV*, 2016. 6

[21] D. Novotný, D. Larlus, and A. Vedaldi. Learning 3d object categories by looking around them. *ICCV*, 2017. 3

[22] X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia. Geonet : Geometric neural network for joint depth and surface normal estimation. *CVPR*, 2018. 3

[23] R. Qin, J. Tian, and P. Reinartz. 3d change detection approaches and applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 122:41 – 56, 2016. 1

[24] B. L. Saux, N. Yokoya, R. Hansch, M. Bosch, G. D. Hager, and H. Kim. IEEE GRSS Data Fusion Contest: Semantic 3D Reconstruction. *IEEE Geoscience and Remote Sensing Magazine*, March 2019. 1

[25] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *ICCV*, 2007. 2

[26] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. *ECCV*, 2012. 2, 3

[27] S. Srivastava, M. Volpi, and D. Tuia. Joint height estimation and semantic labeling of monocular aerial images with cnns. *IGARSS*, 2017. 1, 3

[28] X. Wang, D. F. Fouhey, and A. Gupta. Designing deep networks for surface normal estimation. *CVPR*, 2015. 2

[29] C. Yang, J. H. Everitt, Q. Du, B. Luo, and J. Chanussot. Using high-resolution airborne and satellite imagery to assess crop growth and yield variability for precision agriculture. *Proceedings of the IEEE*, 101(3):582–592, March 2013. 1

[30] Y. Zhang and T. Funkhouser. Deep depth completion of a single RGB-D image. *CVPR*, 2018. 2, 3, 4