UNIVERSITY OF ZAGREB

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

MASTER THESIS No. 601

SELF-SUPERVISED LEARNING OF STEREOSCOPIC RECONSTRUCTION THROUGH PSEUDO-LABELING

David Kerman

Zagreb, June 2024

UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

MASTER THESIS No. 601

SELF-SUPERVISED LEARNING OF STEREOSCOPIC RECONSTRUCTION THROUGH PSEUDO-LABELING

David Kerman

Zagreb, June 2024

UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Zagreb, 04 March 2024

MASTER THESIS ASSIGNMENT No. 601

Student:	**David Kerman (0036522014)**
Study:	Computing
Profile:	Computer Science
Mentor:	prof. Siniša Šegvić

Title: **Self-supervised learning of stereoscopic reconstruction through pseudo-labeling**

Description:

Stereoscopic reconstruction is an important computer vision task with many interesting applications. However, standard supervised learning requires huge amounts of labeled data that are neither easy nor cheap to procure. This is why we are interested in algorithms that can learn on unlabeled image pairs. The candidate will choose a frame for automatic differentiation and get to know the libraries for handling matrices and images. Study and briefly describe existing architectures for stereoscopic reconstruction. Experiment with different methods for learning the reconstruction model on unlabeled image pairs. Carry out model learning and hyperparameter validation. Evaluate the learned models and present the achieved performance. The work is to be accompanied the original and executable code of the developed procedures, test sequences and results, along with the necessary explanations and documentation. Cite the literature used and indicate the help received.

Submission date: 28 June 2024

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

DIPLOMSKI ZADATAK br. 601

Pristupnik:	**David Kerman (0036522014)**
Studij:	Računarstvo
Profil:	Računarska znanost
Mentor:	prof. dr. sc. Siniša Šegvić

Zadatak: **Samonadzirano učenje stereoskopske rekonstrukcije pseudooznačavanjem**

Opis zadatka:

Stereoskopska rekonstrukcija važan je zadatak računalnog vida s mnogim zanimljivim primjenama. Međutim, standardno nadzirano učenje zahtijeva ogromne količine označenih podataka koje nije ni jednostavno ni jeftino pripremiti. Zbog toga smo zainteresirani za oblikovanje postupaka koji mogu učiti na neoznačenim parovima slika. U okviru rada, potrebno je odabrati okvir za automatsku diferencijaciju te upoznati biblioteke za rukovanje matricama i slikama. Proučiti i ukratko opisati postojeće arhitekture za stereoskopsku rekonstrukciju. Isprobati različite postupke za učenje rekonstrukcijskog modela na neoznačenim parovima slika. Uhodati postupke učenja modela te validiranje hiperparametara. Vrednovati naučene modele te prikazati i ocijeniti postignutu točnost. Radu priložiti izvorni i izvršni kod razvijenih postupaka, ispitne slijedove i rezultate, uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 28. lipnja 2024.

UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

MASTER THESIS No. 601

SELF-SUPERVISED LEARNING OF STEREOSCOPIC RECONSTRUCTION THROUGH PSEUDO-LABELING

David Kerman

Zagreb, June, 2024

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 601

SAMONADZIRANO UČENJE STEREOSKOPSKE REKONSTRUKCIJE PSEUDOOZNAČAVANJEM

David Kerman

Zagreb, lipanj, 2024.

UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Zagreb, 04 March 2024

MASTER THESIS ASSIGNMENT No. 601

Student:	**David Kerman (0036522014)**
Study:	Computing
Profile:	Computer Science
Mentor:	prof. Siniša Šegvić

Title: **Self-supervised learning of stereoscopic reconstruction through pseudo-labeling**

Description:

Stereoscopic reconstruction is an important computer vision task with many interesting applications. However, standard supervised learning requires huge amounts of labeled data that are neither easy nor cheap to procure. This is why we are interested in algorithms that can learn on unlabeled image pairs. The candidate will choose a frame for automatic differentiation and get to know the libraries for handling matrices and images. Study and briefly describe existing architectures for stereoscopic reconstruction. Experiment with different methods for learning the reconstruction model on unlabeled image pairs. Carry out model learning and hyperparameter validation. Evaluate the learned models and present the achieved performance. The work is to be accompanied the original and executable code of the developed procedures, test sequences and results, along with the necessary explanations and documentation. Cite the literature used and indicate the help received.

Submission date: 28 June 2024

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

DIPLOMSKI ZADATAK br. 601

Pristupnik:	**David Kerman (0036522014)**
Studij:	Računarstvo
Profil:	Računarska znanost
Mentor:	prof. dr. sc. Siniša Šegvić

Zadatak: **Samonadzirano učenje stereoskopske rekonstrukcije pseudooznačavanjem**

Opis zadatka:

Stereoskopska rekonstrukcija važan je zadatak računalnog vida s mnogim zanimljivim primjenama. Međutim, standardno nadzirano učenje zahtijeva ogromne količine označenih podataka koje nije ni jednostavno ni jeftino pripremiti. Zbog toga smo zainteresirani za oblikovanje postupaka koji mogu učiti na neoznačenim parovima slika. U okviru rada, potrebno je odabrati okvir za automatsku diferencijaciju te upoznati biblioteke za rukovanje matricama i slikama. Proučiti i ukratko opisati postojeće arhitekture za stereoskopsku rekonstrukciju. Isprobati različite postupke za učenje rekonstrukcijskog modela na neoznačenim parovima slika. Uhodati postupke učenja modela te validiranje hiperparametara. Vrednovati naučene modele te prikazati i ocijeniti postignutu točnost. Radu priložiti izvorni i izvršni kod razvijenih postupaka, ispitne slijedove i rezultate, uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 28. lipnja 2024.

I want to thank my mentor, Prof. Siniša Šegvić, for all the knowledge and help he provided during my studies.

A big shoutout to Nenad Markuš for the brainstorming sessions and great advices.

Lastly, I am incredibly grateful to my family, friends, and girlfriend for their constant support and encouragement.

Contents

1 Introduction

Stereoscopic reconstruction focuses on estimating the three-dimensional positions of points from two or more images. This important task has found use in areas such as autonomous navigation, computer modeling, and virtual reality. Early approaches established correspondence by relying on pixel space distances and manually designed features. With the rise of more powerful computers, deep learning has enabled learning these correspondence metrics directly from real data.

To achieve stereoscopic reconstruction, rectified stereo images are used, originating from a calibrated pair of cameras aligned in the same direction, ensuring that the rows of both cameras lie on the same plane. The goal is to find matching pixels in the left and right images. The difference in position between matched pixels, known as disparity, is inversely related to distance: pixels with greater disparity are closer, while those with less disparity are farther away.

The two main approaches for stereoscopic reconstruction: build upon handcrafted similarity and end-to-end learning and deep learning-based methods. However, providing the labeled data for learning the correspondence is hard because it requires expensive, complex multi-sensor setups. Recently, self-supervised methods have become popular, as they can learn to produce reliable disparities and predictions by learning on unlabeled stereo image pairs, even though they are more complex than supervised methods.

In this work, we present a self-supervised approach to stereoscopic reconstruction through pseudo-labeling. [1] and [2] We begin with an overview of the key concepts in stereoscopic reconstruction and the relevant deep learning models. Next, we describe our self-supervised learning method, our filtering techniques to improve the accuracy of pseudo-labels. We then discuss the performance of the supervised stereo model using these enhanced pseudo-labels. Finally, we present our experimental results and conclude with some directions for future work.

2 Stereoscopic reconstruction

Stereoscopic reconstruction is a challenging and fascinating problem in computer vision. The problem aims to determine the 3D location of each pixel of a stereo pair. This problem has significant applications in robotics, autonomous vehicles, and various industries [3, 4]. Methods for stereoscopic reconstruction are divided into methods with hand-crafted correspondence i methods with end-to-end learning and deep learning-based methods.

Stereoscopic reconstruction algorithms with hand-crafted correspondence generally follow four main steps [5]. First, the matching cost is calculated to measure pixel similarity between images. Second, these costs are aggregated for robustness. Third, optimization finds the best disparity for each pixel by minimizing the aggregated cost, using either local methods, which consider pixel neighborhoods, or global methods, which optimize over the entire image. Finally, disparity refinement improves the disparity map by reducing noise and errors. These steps enable stereoscopic algorithms to produce accurate 3D scene representations, essential for applications requiring precise depth information, such as autonomous driving.

Understanding two-view geometry is crucial in stereoscopic reconstruction, as it helps in grasping spatial relationships between two images of the same object. This involves concepts such as epipolar geometry and rectified image pairs. The main task in stereoscopic correspondence is identifying matching pixels in two images that correspond to the same point in 3D space. Sparse correspondences involve disparities for pixels,which are typically used in camera motion estimation, while dense correspondences are used for scene structure reconstruction. Challenges in dense correspondence include textureless regions, reflective surfaces, and occlusions.

2.1 Two-view geometry

Two-view geometry describes the relationship between points in images obtained from two cameras. A significant outcome of two-view geometry is the epipolar constraint, which greatly reduces the number of points that need to be checked when searching for corresponding points. Figure 2.1 shows a point X in 3D space being captured by two cameras. Points C and C' represent the centers of the left and right cameras, respectively. The projection of point X onto the plane of the left camera is x, and onto the plane of the right camera is x'. It is important to note that points X, x, and x' lie in the same plane and together with the camera centers C and C' form the so-called epipolar plane π.

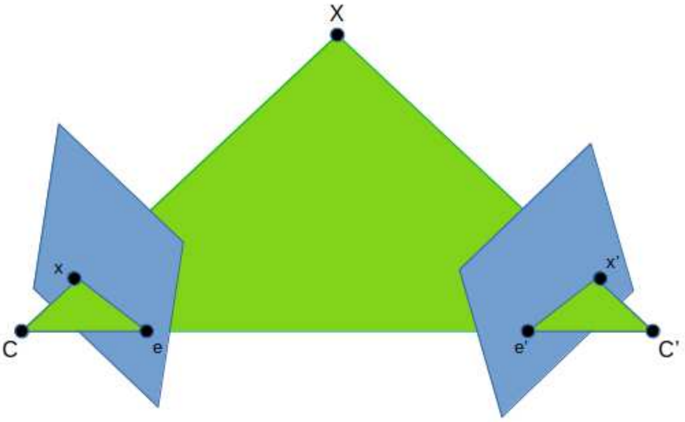

Figure 2.1: The epipolar plane is defined by the observed point X and the centers of the two cameras, C and C'.

The epipolar plane is defined by the observed point X and the centers of the two cameras, C and C'. Additionally, the image shows epipoles, denoted as e and e', which represent the points where the line connecting the camera centers C and C' intersects the image planes. Epipolar lines connect x and e, and x' and e'. These lines are formed by the intersection of the epipolar plane with the image plane. The epipole is the point where all epipolar lines intersect. Knowing that points X, C, and C' lie in the same plane, and if the positions of points x, e, and e' are known, we can estimate the position of point x'. The potential positions of point x' are located on the epipolar line. This epipolar constraint significantly simplifies the search for corresponding points.

The parameters of a camera system's geometry can be divided into two types: intrinsic and extrinsic. Extrinsic parameters describe the relationship between a pair of stereo cameras. Intrinsic parameters describe the properties of the cameras that pertain to each camera individually. For example, intrinsic parameters describe lens imperfections, sensor displacement from the lens center, and other physical characteristics of the cameras.

Extrinsic parameters, on the other hand, lead to necessary transformations that align the images to the same projection plane, ensuring that the pixels along a horizontal line in one image have correspondences on the same line in the other image. This line, already mentioned, is the epipolar line. Such transformations greatly simplify the search for corresponding points, which in the rectified case need only be searched along a single line along the epipolar line. The problem is thus reduced to one dimension. Intrinsic and extrinsic parameters are obtained through appropriate calibration procedures.

For the rectification process, extrinsic camera parameters are crucial, while their calculation is influenced by intrinsic parameters. Through image rectification transformation, we obtain images whose pixels correspond to points at the same height in image and lie along the same epipolar line.

Figure 2.2 shows a pair of images from stereo cameras that have been rectified. The images are from the KITTI 2015 dataset and display epipolar lines. It is easy to notice that the corresponding pixels have the same y coordinat. With this transformation, corresponding pixels need only be searched along the same y-coordinate.

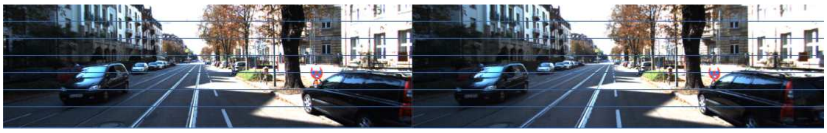

Figure 2.2: Epipolar lines in rectified images from the KITTI 2015 dataset.

Calibration and rectification enable simple depth reconstruction in terms of disparity. Disparity determines the pixel in the image taken by the right camera that corresponds to each pixel taken by the left camera, but shifted by a distance d pixels along the horizontal axis. Disparity is defined as the horizontal shift d between corresponding pixels in images taken by two concurrent stereo cameras. As a result of disparity calculation, a disparity map is created that contains the disparity for each pixel of the reference camera. The relationship between pixels in the left camera image, which is the reference in this case, and the right camera image is defined as:

$$I_L(x,y) = I_R(x-d,y) \tag{2.1}$$

where I_L and I_R are the images taken by the left and right cameras, respectively. The ordered pair (x, y) describes the pixels in the left camera image, and their corresponding pixels in the right camera image are shifted by the disparity d along the horizontal axis $(x-d, y)$. Using the calculated disparity, the relationship between the camera pixels and the scene depth can be expressed as:

$$Z = \frac{B \cdot f}{d} \tag{2.2}$$

where Z is the scene depth, d is the disparity, B is the distance between the camera centers, and f is the focal length of the camera. Equation 2.2 shows the inverse proportionality between the object distance from the cameras and the disparity. When the object is close to the cameras, the disparity will be large. As the object moves away from the cameras, the disparity decreases, reaching zero at infinity.

Figure 2.3 shows that the epipolar lines are parallel and that the y-coordinates of corresponding pixels are equal. The disparity, denoted by d, represents the horizontal shift.

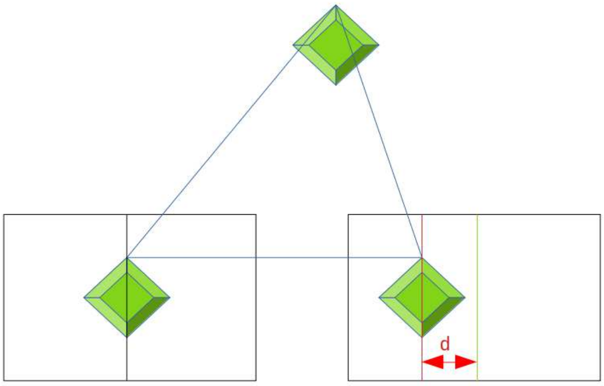

Figure 2.3: Object representation in the images from the left and right cameras. The epipolar plane is marked in blue. The vertical axes indicate the projected points in the left and right images. The correct projection using the epipolar plane is marked in red, while the transformation of the left projection is marked in green. Disparity is denoted by the letter d.

2.2 Optical flow

Optical flow is a crucial concept in computer vision that involves the analysis and prediction of object movements within a sequence of images. While stereoscopic reconstruction focuses on reconstructing three-dimensional structures using two or more images captured from different viewpoints, optical flow concentrates on the temporal analysis of motion between consecutive images.

The relationship between stereoscopic reconstruction and optical flow arises from the fact that both methods use information about the disparity or shift between images to obtain four-dimensional information. In stereoscopic reconstruction, disparity is used to calculate the depth and three-dimensional structure of an object. In contrast, optical flow analyzes spatial changes between successive images to predict dense motion vectors in the scene.

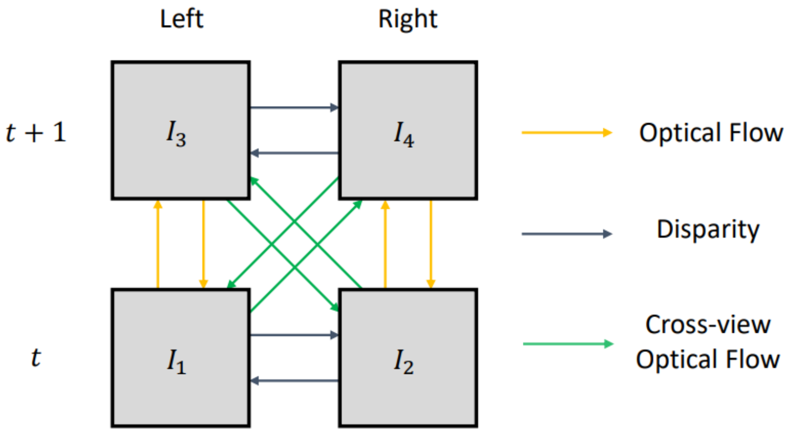

Figure 2.4: The relationship between optical flow and disparity as illustrated in [6].

A key concept connecting these two methods is the assumption of consistency, which implies that corresponding points in two images captured from different views are located at the same three-dimensional position. When this assumption holds, disparity and optical flow can be used together to create a more complete and accurate reconstruction.

Figure 2.4 illustrates the geometric relationship between disparity and optical flow. Both concepts involve estimating the change between two images; however, optical flow seeks changes over the temporal component, i.e., the change from time t to time $t+1$ for two left or right images. It is important to note that in optical flow, changes must be determined for both the x and y directions.

Furthermore, stereoscopic reconstruction implies movement along the epipolar line, meaning that the rows in the left and right images lie in the same plane. This implies that there is no need to determine the shift in the y component, as it is zero for all pixels. Thus, stereoscopic reconstruction can be considered a special case of optical flow where the y component of the flow is set to zero, and two images from the present moment, separated by a certain baseline, are used instead of images from the current and future time.

2.3 Deep models for stereoscopic reconstruction

Deep learning models have significantly advanced the field of stereoscopic reconstruction, providing more accurate and efficient methods for generating 3D structures from stereo image pairs. These models leverage the power of convolutional neural networks (CNNs) and other deep learning architectures to automatically learn features from large datasets, which helps in improving disparity estimation and depth perception.

A typical deep learning pipeline for stereoscopic reconstruction involves several key stages:

- **Feature extraction:** CNNs are employed to extract meaningful features from the input images. These features embed local neighbourhoods into a suitable metric space.

- **Matching cost computation:** The extracted features are used to compute a matching cost, which quantifies the similarity between corresponding points in the stereo images.

- **Cost aggregation:** To improve robustness, the matching costs are aggregated over a local neighborhood. This helps in reducing noise and improving the accuracy of disparity maps.

- **Disparity estimation:** The aggregated costs are used to estimate the disparity for each pixel. This step may involve optimization techniques or further deep learning models to refine the disparity map.

- **Post-processing:** Finally, the disparity map may be refined using post-processing techniques to correct any errors and enhance the overall quality of the reconstructed 3D structure.

Previous advancements have introduced end-to-end deep learning models that streamline the entire process, from feature extraction to disparity estimation. These deep models not only enhance the accuracy of stereoscopic reconstruction but also offer greater efficiency, making them suitable for real-time applications in robotics, autonomous driving, and augmented reality.

2.3.1 McCNN

The Multi-Column Convolutional Neural Network (McCNN) was a state-of-the-art method for stereo matching, developed by Žbontar and LeCun [7] [8]. It leverages deep learning techniques to compute disparity maps from stereo image pairs. Deep learning significantly improves both accuracy and computational efficiency.

Architecture overview: McCNN comprises two distinct architectures tailored for different performance needs:

- *Accurate architecture*: This version focuses on minimizing error rates, using a deeper and more complex network structure. It replaces the traditional cosine similarity measure with multiple fully-connected layers. Those modifications enhances precision at the cost of increased computational time. Visualized in Figure 2.6.

- *Fast architecture*: Designed for real-time applications, this architecture is optimized for speed. It employs fewer layers and a simpler structure, achieving rapid computations with a slight compromise in accuracy. Visualized in Figure 2.5

Training examples are created by extracting positive and negative pairs of image patches based on known disparities. Positive pairs consist of patches centered around the same 3D point, while negative pairs do not match. Image patches from the left and right stereo images are processed through several convolutional layers with ReLU activation function.

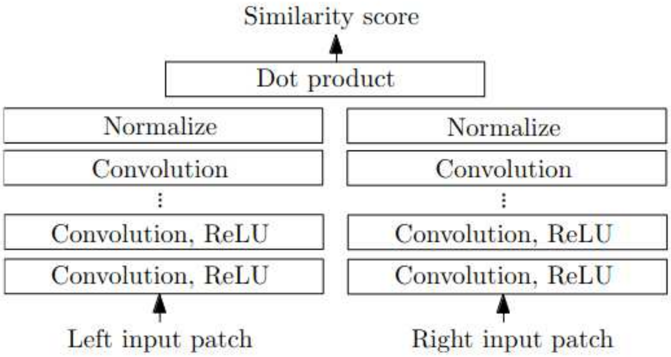

Figure 2.5: The fast architecture is a siamese network. The two sub-networks consist of a number of convolutional layers followed by rectified linear units (ReLU). The similarity score is obtained by extracting a vector from each of the two input patches and computing the cosine similarity between them.[8]

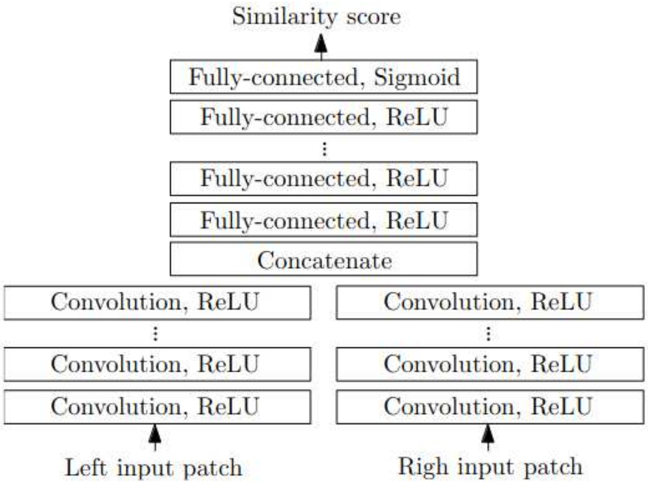

Figure 2.6: The accurate architecture begins with two convolutional feature extractors. The extracted feature vectors are concatenated and compared by a number of fully-connected layers. The inputs are two image patches and the output is a single real number between 0 and 1, which we interpret as a measure of similarity between the input images.[8]

The accurate architecture further refines these patches using fully-connected layers, whereas the fast architecture utilizes a simpler normalization and dot product for similarity computation. After generating the initial disparity map, several post-processing techniques can be applied such as cross-based cost aggregation, semiglobal matching, left-right consistency check and subpixel enhancement. [8]

2.3.2 RAFT-Stereo

RAFT-Stereo [9], depicted in Figure 2.2, is an extension of the RAFT [10] model designed specifically for processing stereo images. The original RAFT model was developed for optical flow estimation and consists of three key components: a feature encoder, a cost volume construction module, and an iterative feature update operator, as illustrated in Figure 2.1.

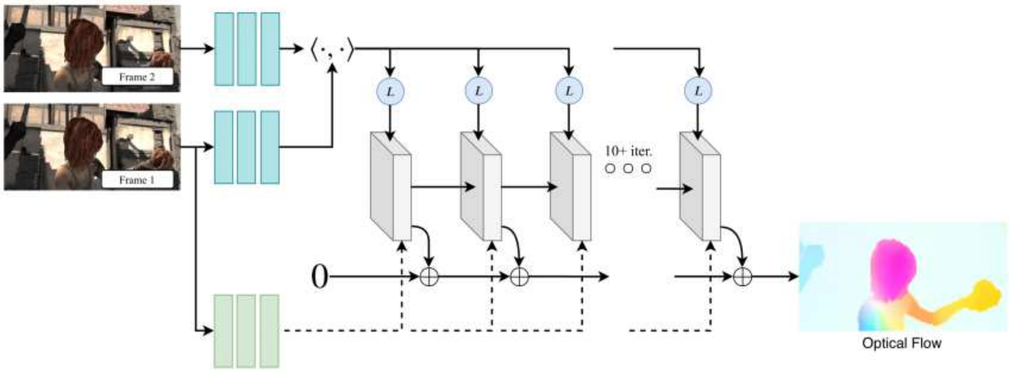

Figure 2.7: The RAFT model [10] consists of three main components: a feature encoder, a cost volume construction module, and an iterative flow update operator.

The feature encoder is used to extract features from the input images, which are then used to construct the cost volume. It is designed as a series of residual blocks, independently applied to both images. Additionally, there is a context feature encoder that takes only the first image as input. The context features are utilized in the update operator.

When addressing the problem of disparity estimation or stereoscopic reconstruction, it is assumed that the motion occurs only along epipolar lines. The construction of the correlation volume calculates visual similarity only between pixels at the same height, i.e., along the same epipolar line. The resulting volume is three-dimensional, whereas in the original RAFT model, it is four-dimensional due to the calculation of the correlation volume over all pixels in the image.

The third component of the model is the iterative update operator, which consists of convolutional Gated Recurrent Unit (GRU) cells. These cells search through the cost volume to refine the disparity estimation.

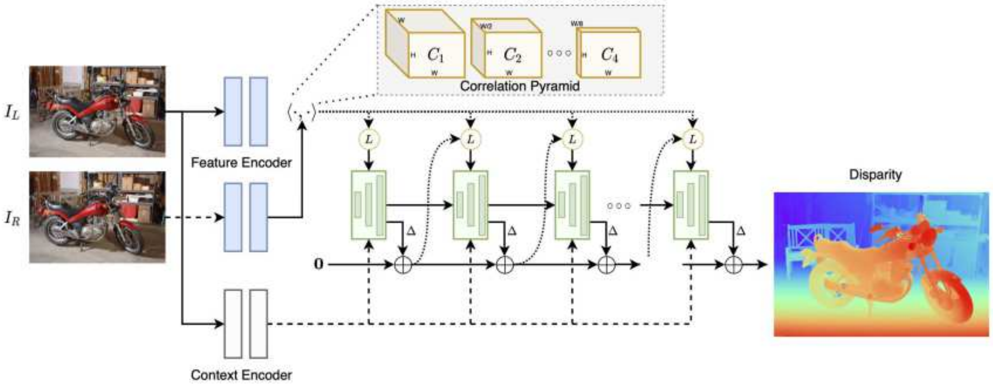

Figure 2.8: The RAFT-Stereo model [9]. The cost volume is now three-dimensional since it assumes movement along the epipolar line.

In the RAFT-Stereo model, multiple convolutional GRU cells are used to search at various resolution levels, thereby increasing the model's receptive field and its ability to reconstruct fine details in the images. During the application of the update operator, the model uses features at /8, /16, and /32 resolutions.

RAFT-Stereo's architecture allows to efficiently handle stereo image pairs, leveraging the epipolar constraint to reduce computational complexity and improve accuracy in disparity estimation.

3 Self-supervised stereoscopic reconstruction

Laser scanners (LiDARs) are often expensive and challenging to operate, which has led to the need for learning algorithms that do not require the ground truth labels. Therefore, there is a growing interest in developing accurate and efficient methods for disparity estimation using only images, without relying on any labels.

Self-supervised learning is a form of deep learning where the model does not have access to ground truth labels during training. Instead, it uses the data itself to generate supervisory signals. Several highly effective solutions have been proposed in the literature for self-supervised stereoscopic reconstruction. Most of these methods rely on training models based on the cycle consistency.[11] This method involves predicting disparities in both directions (left-to-right and right-to-left) and ensuring that these predictions are consistent when cycled through both images. This approach helps in regularizing the disparity map and reducing errors due to occlusions or textureless regions.

Moreover, self-supervised learning models often incorporate photometric loss [12], which penalizes differences in pixel intensity between the original and reconstructed images. This loss function encourages the model to produce disparities that result in realistic image reconstructions. In addition to these techniques, augmentation strategies like flipping, cropping, and color jittering are employed to make the model robust to various scenarios and improve generalization.

Overall, self-supervised learning for stereoscopic reconstruction holds significant promise due to its ability to train accurate and robust models without the need for expensive labeled data. This approach not only reduces the reliance on costly hardware but also opens up new possibilities for large-scale deployment in real-world applications.

3.1 Reversing PSM-Net

The method, as proposed by Aleotti et al., focuses on leveraging monocular distillation techniques to improve self-supervised deep stereo learning. Figure 3.1 illustrates the key stages of their approach:

1. **Initial disparity estimation**: Calculation of an initial disparity map (D) using traditional stereo methods.

2. **Disparity refinement**: Application of a filtering mechanism (F) to refine the initial disparity map D, reducing noise and outliers.

3. **Monocular distillation**: Training of a monocular completion network (MCN) using the refined disparity map (D') to predict dense depth maps (D_O).

4. **Pseudo-label generation**: Generation of pseudo-labels (D_P) through a consensus mechanism from MCN predictions to supervise deep stereo network training.

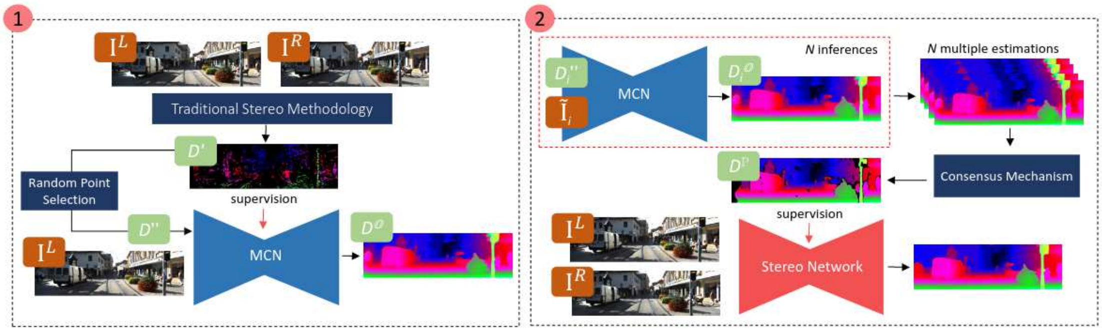

Figure 3.1: Overview of the method proposed by Aleotti et al. [1]

To begin, Aleotti et al. utilize a traditional stereo matcher (S), a method introduced by the authors in their earlier work [13], to compute an initial disparity map (D) from a stereo pair (I_L, I_R):

$$D = S(I_L, I_R)$$

The method (S) involves the classic Census transformation and measures similarity with Hamming distance. This method allows them to generate a dense disparity map without relying on ground truth labels. However, it's noted that due to the inherent uncertainties in the Census method, some pixels in the disparity map may still exhibit high levels of uncertainty. To enhance the accuracy of the initial disparity map (D), Aleotti et al. apply a filtering strategy (F) [1]:

$$D' = F(D)$$

Tosi et al.[14] introduced a series of filtering and prediction confidence estimation techniques. This step aims to improve the quality of disparity estimation by removing noisy or erroneous disparity points, resulting in a more reliable disparity map (D'). Using the refined disparity map (D'), Aleotti et al. train a monocular completion network (MCN) to predict dense disparity maps (D_O):

$$D_O = \mathrm{MCN}(I_L, D')$$

. The MCN is designed to handle occlusions and other challenges inherent in stereo disparity estimation tasks. In their work, the authors employ the deep convolutional model MonoResMatch [13], which takes the left image as input and produces a percentage of randomly selected pseudo-labels to serve as initial points for depth completion, while the rest of the pseudo-labels are used to supervise the learning process itself. With the trained MCN, Aleotti et al. generate pseudo-labels (D_P) to supervise deep stereo network training. These pseudo-labels are obtained through a consensus mechanism by multiple passes through the MCN model. The consensus mechanism considers only those pixels for which multiple predictions overlap, meaning there are no significant deviations in predictions. For these overlapping predictions, the consensus mechanism calculates the average disparity representation and then computes the standard deviation; pixels with deviations exceeding a specified threshold are disregarded.

The final phase of the algorithm involves training the deep model PSMNet [15], using the pseudo-labels obtained from the consensus mechanism.

3.2 Self-supervised learning of stereoscopic reconstruction through pseudo-labeling

In previous sections, we introduced several approaches for stereoscopic reconstruction. Our goal was to develop an efficient method based on pseudo-labeling, initially introduced in [2] and further inspired by the works in [1] and [16]. We simplified the method proposed by [1] by employing the RAFT-Stereo model on filtered pseudo-labels, omitting the need for a monocular completion network (MCN). Additionally, we integrated pseudo-labels derived from the McCNN model using metric embeddings, enhanced by several intelligent filtering techniques. Our method consists of two main phases: generating pseudo-labels and self-supervised learning of deep stereo model on those pseudo-labels.

3.3 Generating initial correspondences

Initial pseudo-labels are crucial in self-supervised learning for stereo reconstruction. To perform well, pseudo-labels should be as dense as possible but highly accurate. In this work, we used two approaches in generating pseudo-labels.

The first approach involves a traditional stereo method, which consists of Census transformation with corresponding aggressive filtering [14] (further reffered to as Aleotti pseudo-labels). This method is derived from [1]. The second approach is based on the deep convolutional model McCNN [7], which is trained in a weakly-supervised manner using triplet loss [16] (further reffered to as McCNN pseudo-labels). Our ultimate objective is to bridge the gap between these two types of label generation methods.

3.3.1 Weakly-supervised McCNN pseudo-labels

Our second approach for generating pseudo-labels draws inspiration from the work of [16]. We intend to replace a complex handcrafted approach that may not be easily ported to other tasks with a simple transparent approach that trains exclusively on data. Initially, the left and right images are processed through a neural network to produce a feature map in a high-dimensional space. We employed a fully convolutional model, McCNN, as outlined in [7], with several modifications. This model and its associated methodology have been detailed in earlier chapters.

Our training method for McCNN leverages a triplet loss function, as originally used in [7]. However, unlike the original supervised training approach, we adapted it for weakly-supervised learning.[17] Although we do not have exact pixel correspondences in weakly-supervised learning, we know that pixels from a row in the left image correspond to pixels from the same row in the right image, while pixels from different rows do not match, based on the epipolar line assumption.

We train the model by selecting an anchor pixel from the left image and identifying the positive bag (target) and negative bag (non-target) examples from the right image. The positive bag contains pixels in the right image on the same row as the anchor in the left image, while the negative bags are pixels in both images that are 3 rows above and below the positives.

For the chosen anchor, positives, and negatives, we calculate a similarity matrix using a row-wise matrix product. We subtract a threshold value from this similarity matrix to control the strictness of the similarity criterion, then apply a softmax function to scale the similarities to values between 0 and 1 and further identifying exact matches through the maximum value.

The resulting pseudo-labels are relatively dense with low error. To enhance the results further, we introduced additional consistency checks and filtering techniques. In the next chapter, we will introduce techniques aimed at producing more accurate, but sparser disparity maps.

3.3.2 Aleotti pseudo-labels

As previously mentioned, Aleotti pseudo-labels require Census transformation and several hand-crafted filtering techniques [14]. In Figure 3.2, the Census transformation is applied to a random image from the KITTI 2015 dataset. As can be observed, the Census transformation enhances borders and internal object structures, which is crucial for disparity estimation in texture-less areas.

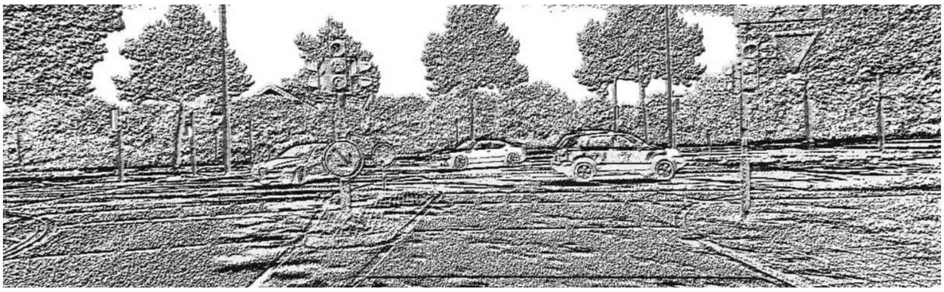

Figure 3.2: Visualization of the Census transformation on a random image from the KITTI 2015 dataset.

3.4 Self-supervised learning

The second phase of the method involves supervised learning of the stereoscopic model with respect to the previously generated pseudo-labels. For this purpose, we utilize the RAFT-Stereo model, which was introduced in one of the earlier chapters.

The training involves L1 loss with scaling based on the iteration. The loss function is as presented in the original paper. Mathematically, the loss function can be expressed as:

$$\mathcal{L} = \sum_{i=1}^{N} \gamma^{N-i} \parallel d_{gt} - d_i \parallel_1 \tag{3.1}$$

The i denotes the iteration from the update operator out of a total of N iterations, γ is set to 0.9, d_i represents the model's disparity prediction at the i-th iteration of the update operator, and d_{gt} are the pseudo-labels generated in the first phase of the algorithm.

As mentioned, the training is conducted in a supervised manner with respect to pseudo-labels obtained from a weak supervision on unlabeled data, which makes learning effectively self-supervised.

4 Filtering McCNN pseudo-labels

As previously outlined, our primary goal is to bridge the performance gap between the opaque hand-crafted approach [1] and our simpler alternative based on weakly supervised learning.. In the following sections, we will introduce a range of filtering techniques that showed initial promise during our research.

First, we explore the use of Mask2Former [18] for panoptic segmentation. Additionally, we apply a distance transform operator to the panoptic maps to emphasize the distance from border of an object. Given our objective to rigorously filter McCNN pseudo-labels, we proposed the use of Lowe ratio filtering based on McCNN embedding similarity scores. Finally, we experimented with a modified version of the Lowe ratio filtering technique.

4.1 Employing Mask2Former for panoptic segmentation

Mask2Former is a state-of-the-art architecture designed to handle various image segmentation tasks, including panoptic, instance, and semantic segmentation, with a single unified model. Mask2Former leverages a masked-attention mechanism within a Transformer decoder, which significantly improves the performance across multiple segmentation benchmarks. [18]

Panoptic segmentation aims to classify each pixel in an image into either a semantic category (e.g., sky, road) or an instance of an object (e.g., a specific car or person). The challenge lies in effectively combining the capabilities of semantic segmentation and instance segmentation within a single framework. Mask2Former addresses this by predicting binary masks for each segment in the image.

The architecture of Mask2Former comprises three main components:

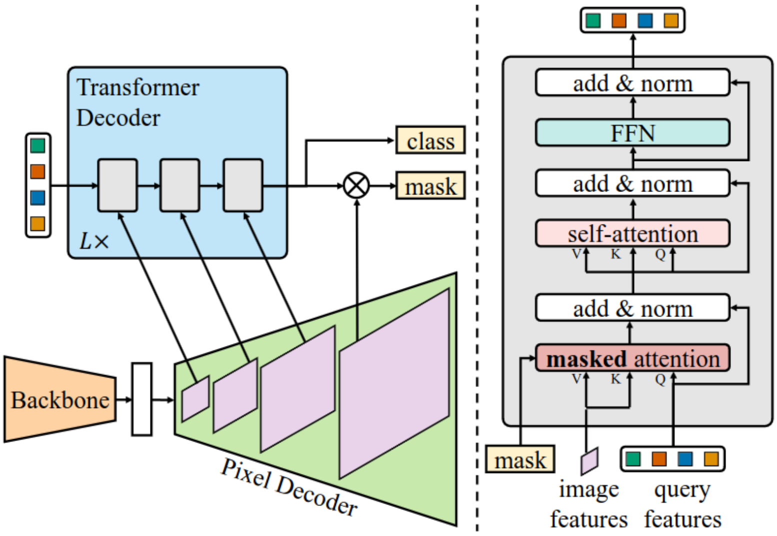

Figure 4.1: Overview of Mask2Former architecture Mask2Former adopts the same meta architecture as MaskFormer [19] with a backbone, a pixel decoder and a Transformer decoder. Cheng et al. proposed a new Transformer decoder with masked attention instead of the standard cross-attention [18]

- **Backbone feature extractor**: Extracts low-resolution feature maps from the input image.
- **Pixel decoder**: Gradually upsamples the low-resolution features to generate high-resolution per-pixel embeddings.
- **Transformer decoder with masked attention**: Processes object queries using masked attention, which focuses on localized features within the predicted mask regions.

Unlike standard cross-attention mechanisms, masked attention restricts the focus to regions within the predicted masks. This not only speeds up convergence but also improves the accuracy of segmentation by emphasizing relevant features and ignoring background noise. To handle objects of various sizes effectively, Mask2Former employs a multi-scale strategy. High-resolution features from the pixel decoder are fed into the Transformer decoder, ensuring that both large and small objects are accurately segmented.

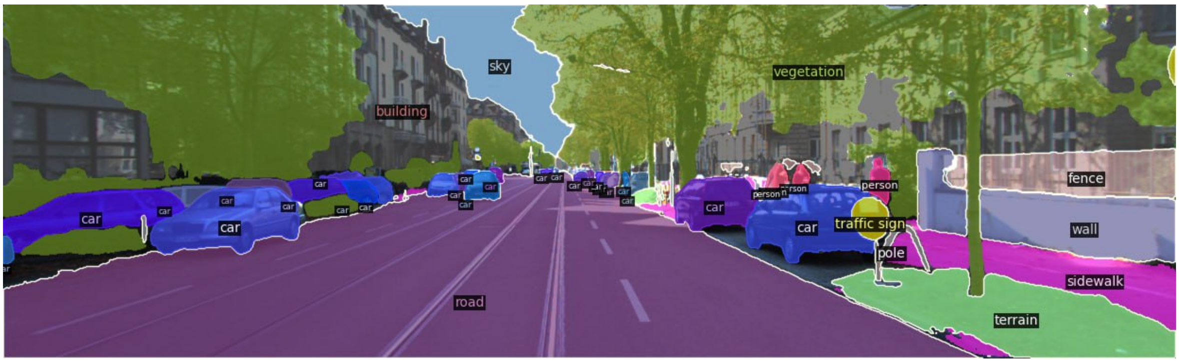

Figure 4.2: Example of panoptic Mask2Former applied to a random image from the KITTI 2015 dataset.

Figure 4.2 presents an example of panoptic Mask2Former applied to a random image from the KITTI 2015 dataset. We adopted panoptic Mask2Former in our work with the goal of generating panoptic segmentation maps and subsequently detecting the borders of objects.

4.1.1 Distance transform operator

The distance transform operator is a technique used to compute the distance from each pixel to the nearest boundary pixel in a binary image. This operator is particularly useful in emphasizing regions of interest, such as object borders, by providing a gradient of distances that highlights proximity to these boundaries.
To apply the distance transform operator in our work, we followed these steps:

- **Panoptic boundaries**: After generating the panoptic segmentation map with Mask2Former, we created a binary image where the object borders were marked with a value of 1, and all other pixels were set to 0. This binary representation isolated the borders of objects within the scene.

- **Distance transform application**: We then applied the distance transform operator to this binary image. This process computed the Euclidean distance from each pixel to the nearest border pixel, resulting in a distance map where the intensity of each pixel indicates its proximity to the nearest object boundary.

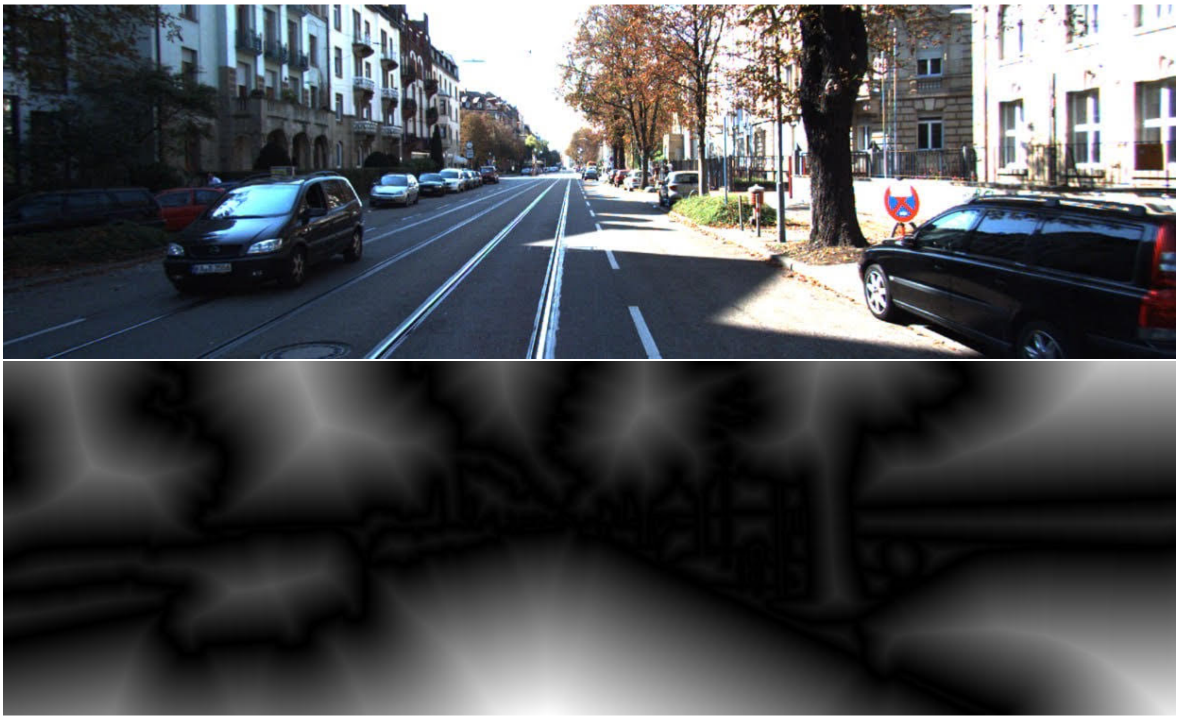

Figure 4.3: Distance transform operator applied to a binary image created using top image.

The resulting distance map, as shown in Figure 4.3, provides a clear visualization of how close each pixel is to the object borders. This information is crucial for further processing steps.

4.2 Lowe Ratio

The Lowe ratio test is a widely used technique in computer vision for feature matching, primarily employed to filter out incorrect matches between feature points. Named after David Lowe, who introduced it as part of the SIFT algorithm [20], this method enhances the robustness of feature matching by considering the best and second-best matches.

The principle behind the Lowe ratio test is based on the assumption that correct feature matches are significantly closer to each other in the feature space than incorrect matches. To implement this, features are detected and described in both the reference image and the target image using a feature extraction algorithm. The distance of the best match (d_1) is compared to the distance of the second-best match (d_2). A match is considered valid if the ratio $\frac{d_1}{d_2}$ is below a certain threshold:

$$\frac{d_1}{d_2} < \text{threshold} \tag{4.1}$$

In our work, we employed the Lowe ratio test on McCNN embedding scores within the cost volume. The objective was to identify and retain only the most reliable matches by analyzing the similarity scores of embeddings.

To apply this method, we first extracted embeddings by the McCNN model. A cost volume was then constructed by calculating the similarity scores between the embeddings of the reference and target images. For each embedding in the reference image, the first and second best matches in the target image were identified based on the similarity scores within the cost volume. The Lowe ratio test was then applied by comparing the similarity scores of the first best match (s_1) and the second best match (s_2). A match is considered valid if the ratio $\frac{s_1}{s_2}$ is below a certain threshold.

$$\frac{s_1}{s_2} < \text{threshold} \tag{4.2}$$

Pseudo-labels that pass the ratio test are considered reliable for further processing.

4.2.1 Modified Lowe ratio

While the original Lowe ratio test provided significant improvements, we proposed a modified version to further refine the selection of reliable pseudo-labels. In this modified approach, our intention was to consider the second-best match from the entire search space, excluding a specific interval around the best similarity score disparity. Specifically, we excluded the interval between the best similarity score disparity ± 10 disparities.

By doing so, we aimed to avoid selecting second-best matches that were too close to the first-best match in terms of disparity, which could lead to false positives in regions with similar but incorrect matches. This modified Lowe ratio test involves first extracting embeddings by the McCNN model. A cost volume was then constructed by calculating the similarity scores between the embeddings of the reference and target images. For each embedding in the reference image, the first-best match in the target image was identified based on the similarity scores within the cost volume. The interval of ± 10 disparities around this first-best match was excluded and then the second-best match was identified from the remaining search space.

The modified Lowe ratio test was applied by comparing the similarity scores of the first-best match (s_1) and the second-best match (s_2), where the second-best match is chosen from the non-excluded interval. A match is considered valid if the ratio $\frac{s_1}{s_2}$ is below the threshold.

5 Implementation

The programming language Python was used for the creation of this work. The PyTorch framework was utilized for building and training models. This framework offers a wide range of tools and support, one of which is automatic differentiation, significantly speeding up the process of model building and training. Torch, the core component of PyTorch, enables tensor computations on a wide array of Nvidia graphics cards.

The Numpy library was used for data preparation and tensor computations. This library is written in the low-level language C and employs libraries such as OpenBLAS, whose key components are directly written in machine code for various modern computer architectures. Such implementation makes operations very fast.

The KITTI 2015 dataset was downloaded from a publicly available website owned by the collaboration between the Karlsruhe Institute of Technology and the Toyota Technological Institute.

Since we used the RAFT Stereo and McCNN architectures in our work, most of our code is based on the official GitHub repositories of these models. The implementation of the RAFT-Stereo model is available on the GitHub profile of Princeton's Computer Vision Laboratory[1], while the implementation of the customized McCNN model is available on the GitHub repository wlrn[2]. The rest of the code is present in repository tinyrend[3].

[1] `https://github.com/princeton-vl/RAFT-Stereo`

[2] `https://github.com/nenadmarkus/wlrn`

[3] `https://github.com/nenadmarkus/tinyrend`

6 Experiments

In this work we performed extensive experimenting of the listed methods in the previous chapters. We have performed experiments based on Reversing PSM-Net [1] and RS-IPA [2]. Also, we performed cross-validation to optimize hyperparameters.

All experiments involve self-supervised learning of models, which means we did not use any disparity labels during training. However, during evaluation, we used subsets of datasets with available disparity labels to obtain an accuracy measure. We evaluated all experiments on the KITTI 2015 Stereo dataset [21]. In the experiments on the KITTI dataset, we used an extended version of the dataset KITTI multiview. The subset for KITTI training, which contains actual disparity values, was used as a validation set, and we recorded accuracy on it. The original testing subset of the KITTI dataset does not have publicly available disparity labels, so we could not use it for validating our method.

6.1 Dataset description

Stereoscopic reconstruction requires a more complex data collection system than many other computer vision problems. First of all, it is necessary to have a calibrated stereoscopic camera with known intrinsic and extrinsic parameters, especially if the data is intended to be used for path planning or 3D object reconstruction. Furthermore, to achieve supervised learning of the model or to conduct model evaluation with metric recording, it is necessary to collect disparity or depth labels for each pixel, which is most commonly done using LIDAR (Light Detection and Ranging). Of course, the cameras and LIDAR must be carefully calibrated to ensure the collected labels are accurate.

The KITTI dataset (Karlsruhe Institute of Technology and Toyota Technological Institute) [21] is one of the most well-known and popular datasets for stereoscopic reconstruction of scenes from a driver's perspective, collected during sunny days while driving through the German city of Karlsruhe. The dataset consists of 200 training images and 200 testing images with a resolution of 384 × 1242, where the training set images have available labels collected with a calibrated camera system.

The labels are sparse - only 30% of the pixels in the scene have valid labels, and in places where there are cars, the labels are further processed so that they are supplemented by fitting the 3D model of the actual car. The KITTI dataset also offers an extended and raw version of the data that follows the dataset's recording sequences. The extended dataset, often referred to as multiview, expands each of the 200 training images to 20 temporally spaced and subsampled images.

In our experiments, we used the extended version of the dataset since we conducted self-supervised methods of stereoscopic reconstruction, which require the use of a larger dataset. However, for evaluation, we used the standard training set consisting of 200 images with available disparity labels. To avoid data leakage from the training set into the test set, we excluded images from the multiview set that were temporally close to the frames used in the test set, specifically the images labeled with _09, _10, _11, and _12.

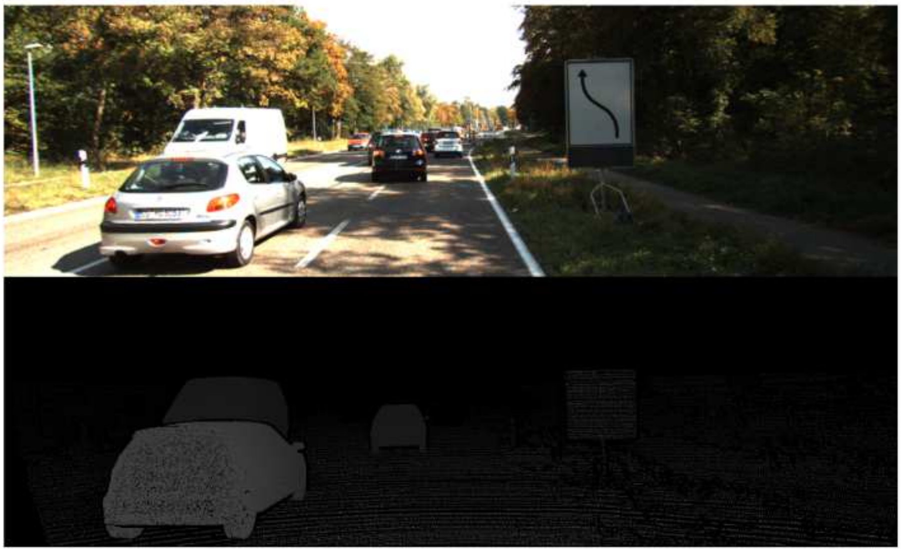

Figure 6.1: An example from the KITTI 2015 dataset. The top image shows the reference left image, and the bottom image shows the corresponding disparity map obtained by a laser sensor.

6.2 Metrics

In the literature, the most commonly used metrics for the problem of stereoscopic reconstruction are AEPE and D1. The AEPE metric (Average End-to-End Pixel Error) or EPE (End-to-End Point Error) is calculated as the Euclidean distance between predictions and actual disparities:

$$EPE = \| d - d_{gt} \|_2 \tag{6.1}$$

where d_{gt} denotes the actual disparity value, and d is the predicted disparity. AEPE is the averaged value over all valid pixels. Valid pixels are those with valid disparity values.

The D1 metric denotes the percentage of incorrectly estimated disparities. This metric is also calculated only on valid pixels. A pixel is considered incorrectly estimated if the deviation of its prediction from the actual value exceeds a certain threshold. Specifically, a pixel is incorrectly estimated if its EPE is greater than the given threshold. For the KITTI dataset, this threshold is 3. The D1 metric is calculated as:

$$D1 = \frac{1}{N} \sum_{i=1}^{N} \| d_i - d_{gt,i} \|_2 > 3 \tag{6.2}$$

where N is the number of valid pixels, d_i is the predicted disparity for the i-th pixel, $d_{gt,i}$ is the actual disparity for the i-th pixel.

6.3 Overview of the results from the literature

To interpret the results of our experimental metrics, Table 6.1 provides an overview of the literature. As shown, supervised methods perform better, in contrast to self-supervised methods, which are trained without labels and perform worse. Our ultimate goal is to approach the performance of the best self-supervised methods.

Method	EPE - all	EPE - noc	D1 - all [%]	D1 - noc [%]
RAFT Stereo* [9]			1.82	1.69
RAFT Stereo [9]	1.13	1.10	5.67	5.44
Reversing PSM-Net [1]	1.00		3.85	
Flow2Stereo [22]	1.34	1.31	6.13	5.93
DDFlow [23]	1.23	1.21	6.37	6.17

Table 6.1: Results from the literature on the subset of the KITTI 2015 dataset. In the table, methods are separated by horizontal lines depending on the type of training used to obtain the results. The first section refers to supervised method, the middle one to method pretrained on other datasets and evaluated only on KITTI, and the last section shows the three self-supervised methods.

6.4 Self-supervised learning through pseudo-labeling

In these experiments, we decided to use the RAFT Stereo model since the upcoming experiments will not utilize the geometric relationship between optical flow and disparity, and the RAFT Stereo model is specifically tailored for disparity estimation tasks. Most of the hyperparameters for training the RAFT Stereo model were adopted from the original paper: image patch size of 320 × 720, learning rate of 0.0001, AdamW optimizer, data augmentation including color jittering and random image block erasing, with a batch size of 8. We conducted training in mixed precision to save on memory. Additionally, the models were trained from random initialization over 12500 iterations with a learning rate of 0.0001.

We conducted experiments with McCNN pseudo-labels, applying various types of filtering methods as presented in previous chapters. Additionally, we compared experiments with Aleotti's pseudo-labels, where no additional filtering was applied apart from the defaults described in the method [1].

6.4.1 Aleotti pseudo-labels

The first method for generating pseudo-labels was adopted from the Reversing PSM-Net method [1]. As said before, we will refer to them as Aleotti pseudo-labels. The method for obtaining Aleotti pseudo-labels is a traditional stereoscopic method based on census transformation and Hamming code distance calculation, meaning that classic model training was not performed; instead, predictions were generated for each image individually.

We did not further experiment with filtering and consistency checks for these pseudo-labels as they are already integrated into the method. These pseudo-labels are quite sparse precisely because of the built-in filtering and consistency checks.

Method	**EPE - all**	**EPE - noc**	**D1 - all [%]**	**D1 - noc [%]**
Reversing PSM-Net [1]	1.00		3.85	
Flow2Stereo [22]	1.34	1.31	6.13	5.93
DDFlow [23]	1.23	1.21	6.37	6.17
Aleotti pseudo-labels	1.00	0.98	3.98	3.81

Table 6.2: Evaluation metrics on the KITTI 2015 dataset demonstrate that we achieved performance close to the Reversing PSM-Net [1] method using Aleotti pseudo-labels but with a significantly simpler training procedure. First three rows are other self-supervised methods. We evaluated on ground-truth disparity maps using *all* valid pixels and only *noc* (non-occluded) valid pixels.

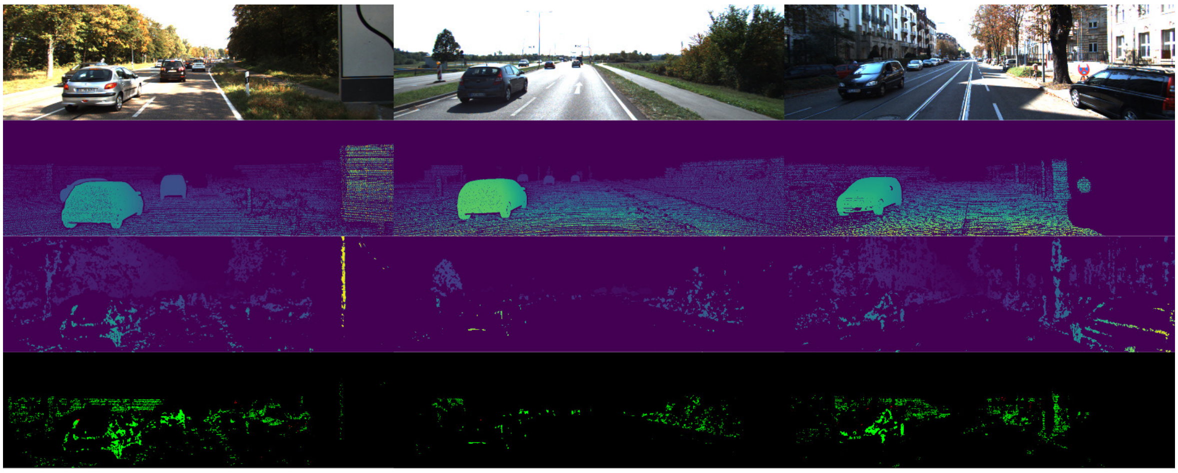

Figure 6.2: The visualization shows the accuracy of Aleotti pseudo-labels. The first row displays the left reference image, the second row the ground-truth disparity maps, the third row the disparity map with Aleotti pseudo-labels, and the fourth row the error map where green pixels are marked as accurate and red as inaccurate. The accurate pixels are those where the disparity is within the interval ±3 pixels, known as the 3px error.

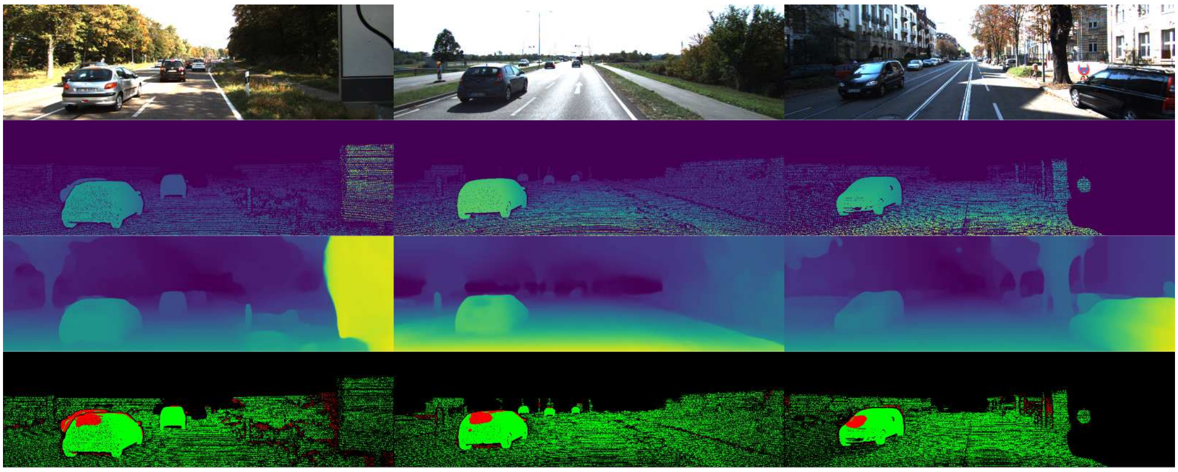

Figure 6.3: The visualization shows the performance of the RAFT-Stereo model trained on Aleotti pseudo-labels. The first row displays the left reference image, the second row the ground-truth disparity maps, the third row the predicted disparity map from the RAFT-Stereo model, and the fourth row the error map where green pixels are marked as accurate and red as inaccurate. The accurate pixels are those where the disparity is within the interval ±3 pixels, known as the 3px error.

The table 6.2 demonstrates that we achieved performance close to the Reversing PSMNet [1] method using Aleotti pseudo-labels but with a significantly simpler training procedure. In figure 6.2, we can observe that Aleotti pseudo-labels are quite sparse but very accurate, with almost no inaccurate labels. Figure 6.3 shows the performance of the RAFT-Stereo model trained on Aleotti pseudo-labels. We can observe that the model struggles in the areas of car instances, especially on reflective or low-texture areas.

6.4.2 McCNN pseudo-labels

In the following section, we present the results of conducted experiments with McCNN pseudo-labels. As an additional filtering step, we applied left-right consistency filtering. Furthermore, to assess the similarity with Aleotti pseudo-labels, we conducted two additional experiments.

The first experiment is with only raw McCNN pseudo-labels. The second one is McCNN ∪ Aleotti pseudo-labels, in which we added Aleotti pseudo-labels to McCNN disparity maps. In other words, we used Aleotti disparity values if they were present in the corresponding map; otherwise, we used McCNN disparity values. In the third experiment, we identified locations where Aleotti disparity values were present and chose those specific locations from the McCNN disparity map.

Method	EPE - all	EPE - noc	D1 - all [%]	D1 - noc [%]
Reversing PSM-Net [1]	1.00		3.85	
Flow2Stereo [22]	1.34	1.31	6.13	5.93
DDFlow [23]	1.23	1.21	6.37	6.17
McCNN pseudo-labels	1.26	1.24	6.53	6.39
McCNN ∪ Aleotti	1.00	0.99	3.99	3.82
McCNN ∩ Aleotti	1.26	1.25	6.55	6.42

Table 6.3: Evaluation metrics on the KITTI 2015 dataset demonstrate performance of RAFT-Stereo models trained on different pseudo-labels. First three rows are other self-supervised methods. We evaluated on ground-truth disparity maps using *all* valid pixels and only *noc* (non-occluded) valid pixels.

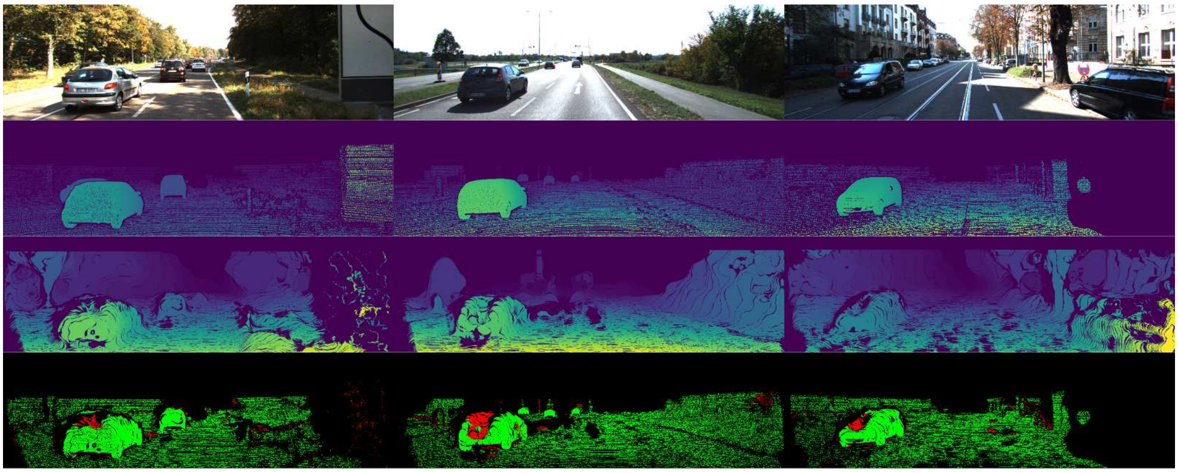

Figure 6.4: The visualization shows McCNN pseudo-labels. The first row displays the left reference image, the second row the ground-truth disparity maps, the third row the disparity map with McCNN pseudo-labels, and the fourth row the error map where green pixels are marked as accurate and red as inaccurate. The accurate pixels are those where the disparity is within the interval ±3 pixels, known as the 3px error.

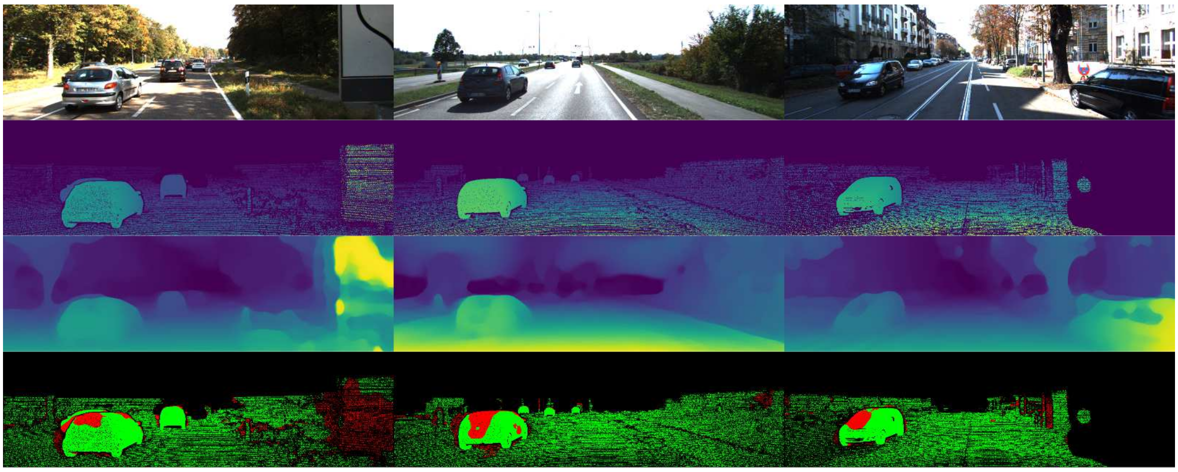

Figure 6.5: The visualization shows the performance of the RAFT-Stereo model trained on McCNN pseudo-labels. The first row displays the left reference image, the second row the ground-truth disparity maps, the third row the predicted disparity map from the RAFT-Stereo model, and the fourth row the error map where green pixels are marked as accurate and red as inaccurate. The accurate pixels are those where the disparity is within the interval ±3 pixels, known as the 3px error.

In table 6.3, it can be observed that raw McCNN pseudo-labels perform worse than Aleotti pseudo-labels by a significant margin. The second experiment showed that adding McCNN to Aleotti pseudo-labels did not improve the RAFT-Stereo performance compared to using only Aleotti pseudo-labels. The third experiment further confirms that McCNN pseudo-labels consistently yield worse results than Aleotti pseudo-labels.

Figure 6.4 shows that raw McCNN pseudo-labels are much denser than Aleotti pseudo-labels but contain a vast number of inaccurate labels. The final RAFT-Stereo model struggles in the same areas of the scene but with significantly more inaccurate pixels, as depicted in figure 6.5.

6.4.3 Utilizing distance transform operator

In previous chapters, we discussed the motivation behind using the distance transform operator. This operator provides a gradient of distances that highlight proximity to object borders, making it a valuable tool for identifying areas close to the edges of objects. We also demonstrated an image after applying the distance transform operator on a binary image representing object borders.

Our primary aim is to discard disparities at the object borders, as there are many inaccurate predictions from the McCNN model in these areas. To leverage the information provided by the distance transform, we propose an operation described by the following equation:

$$M_f = 1 - \sigma(M_d) \tag{6.3}$$

where M_f represents the final map with higher values at the object borders, σ is the sigmoid function applied to all values, and M_d is the map generated by the distance transform operator on binary image.

By applying the sigmoid function to the distance transform map, we obtain a smooth transition of values that enhances the distinction between object borders and other regions. This method ensures that the final map, M_f, emphasizes object borders, helping us to filter out less accurate disparity values and improve the overall performance of our model.

Method	**EPE - all**	**EPE - noc**	**D1 - all [%]**	**D1 - noc [%]**
Reversing PSM-Net [1]	1.00		3.85	
Flow2Stereo [22]	1.34	1.31	6.13	5.93
DDFlow [23]	1.23	1.21	6.37	6.17
McCNN dt=0.33	1.28	1.26	6.47	6.32
McCNN dt=0.38	1.25	1.23	6.40	6.28
McCNN dt=0.43	1.24	1.23	6.39	6.26
McCNN dt=0.48	1.26	1.24	6.53	6.40

Table 6.4: Evaluation metrics on the KITTI 2015 dataset demonstrate the performance of RAFT-Stereo models trained on different pseudo-labels, where dt represents the distance transform threshold. The first three rows are other self-supervised methods. We evaluated on ground-truth disparity maps using *all* valid pixels and only *noc* (non-occluded) valid pixels.

In table 6.4, it can be observed that employing the distance transform operator removes inaccurate pixels on the borders of instances in McCNN pseudo-labels, but does not yield significantly better performance. Figure 6.6 illustrates that there are still a vast number of inaccurate disparities in the car instances. In the following chapters, we will focus on filtering out these disparities.

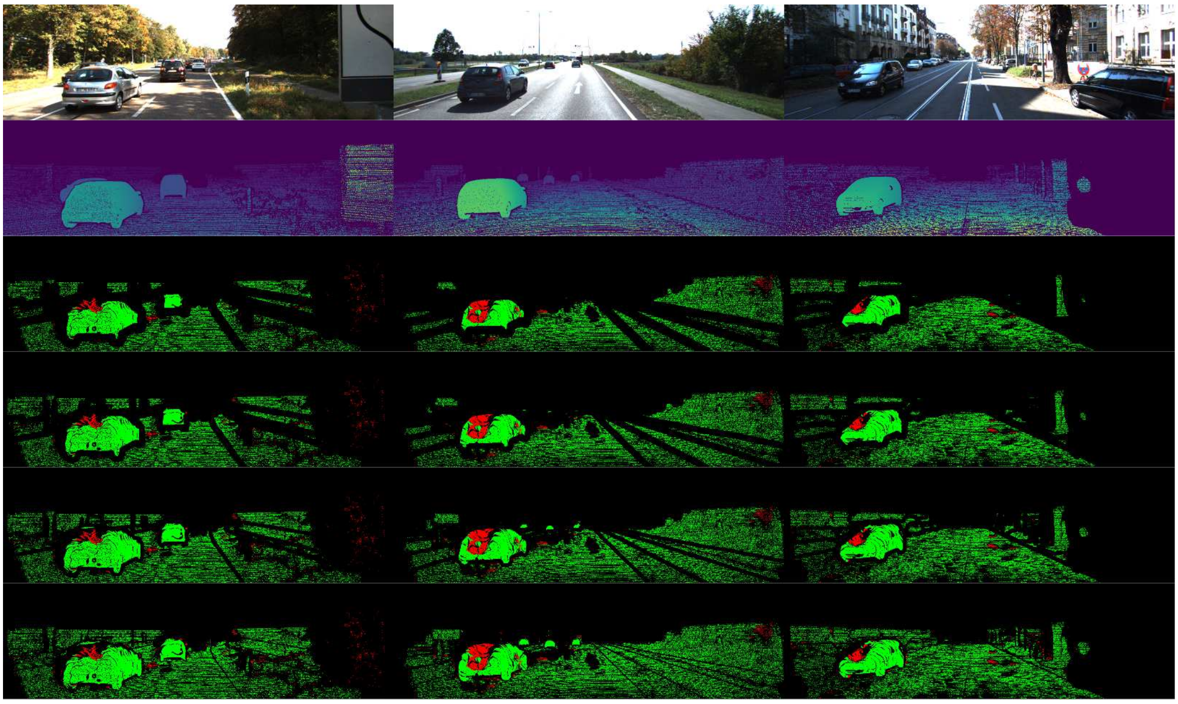

Figure 6.6: The visualization shows McCNN pseudo-labels with different distance transform thresholds applied. The first row displays the left reference image, the second row the ground-truth disparity maps, and from the third to the last rows, the error maps with different thresholds applied (0.33, 0.38, 0.43, 0.48 from top to bottom), where green pixels are marked as accurate and red as inaccurate. The accurate pixels are those where the disparity is within the interval ±3 pixels, known as the 3px error.

6.4.4 Excluding semantic classes

Since we have panoptic segmentation maps, we aim to understand the influence of particular semantic classes. For this purpose, we designed three experiments. We conducted experiments with McCNN pseudo-labels excluding car instances, road areas, and a combined experiment excluding car instances and sky/vegetation areas. Additionally, we applied a left-right consistency check as an additional filtering step.

Method	EPE - all	EPE - noc	D1 - all [%]	D1 - noc [%]
Reversing PSM-Net [1]	1.00		3.85	
Flow2Stereo [22]	1.34	1.31	6.13	5.93
DDFlow [23]	1.23	1.21	6.37	6.17
McCNN w/o cars	1.27	1.25	6.58	6.41
McCNN w/o road	1.27	1.26	6.52	6.38
McCNN w/o car-sky-vegetation	1.28	1.26	6.68	6.50

Table 6.5: Evaluation metrics on the KITTI 2015 dataset demonstrate the performance of RAFT-Stereo models trained on different McCNN pseudo-labels. The first three rows are other self-supervised methods. We evaluated on ground-truth disparity maps using *all* valid pixels and only *noc* (non-occluded) valid pixels.

In table 6.5, it can be observed that excluding car instances and road areas does not significantly affect the model performance. This suggests that there are sufficient disparity locations remaining for training a RAFT-Stereo model.

6.4.5 Applying kernel filtering to panoptic instances

As we are aware of the problem of discontinuity in disparity within car instances in pseudo-labels and its impact on final performance, we proposed applying kernel filtering inside car instances. Kernel filtering helps smooth out disparity values by removing noise and inconsistencies.

Using the generated panoptic segmentation maps, we identify the regions corresponding to car instances. We then apply a kernel filter specifically within these regions. This involves defining a kernel size to create a local neighborhood around each pixel within a car instance. The median disparity value within this neighborhood is computed, and the pixel values are adjusted to be more consistent with this median. Pixels that deviate significantly from the median, based on a predefined threshold, are considered noise and are filtered out. This process ensures that the disparity values within car instances are more uniform and continuous, thereby improving the quality of the pseudo-labels and the final performance of the model.

With kernel filtering applied to car instances, along with additional filtering using distance transform threshold and left-right consistency check, we observed a small improvement in the RAFT-Stereo model. This suggests that we successfully removed a significant number of inaccurate pixels. This improvement can be seen in table 6.6 and figure 6.7.

Method	**EPE - all**	**EPE - noc**	**D1 - all [%]**	**D1 - noc [%]**
Reversing PSM-Net [1]	1.00		3.85	
Flow2Stereo [22]	1.34	1.31	6.13	5.93
DDFlow [23]	1.23	1.21	6.37	6.17
McCNN t=0.40 ks=30 ct=2	1.22	1.21	6.28	6.14
McCNN t=0.44 ks=30 ct=2	1.23	1.22	6.32	6.19
McCNN t=0.48 ks=30 ct=2	1.25	1.24	6.34	6.21
McCNN t=0.40 ks=40 ct=2	1.23	1.22	6.30	6.17
McCNN t=0.44 ks=40 ct=2	1.24	1.23	6.35	6.23
McCNN t=0.48 ks=40 ct=2	1.27	1.25	6.38	6.26

Table 6.6: Evaluation metrics on the KITTI 2015 dataset demonstrate the performance of RAFT-Stereo models trained on different pseudo-labels, where t represents the distance transform threshold, ks kernel size for kernel filtering and ct consistency threshold. The first three rows are other self-supervised methods. We evaluated on ground-truth disparity maps using *all* valid pixels and only *noc* (non-occluded) valid pixels.

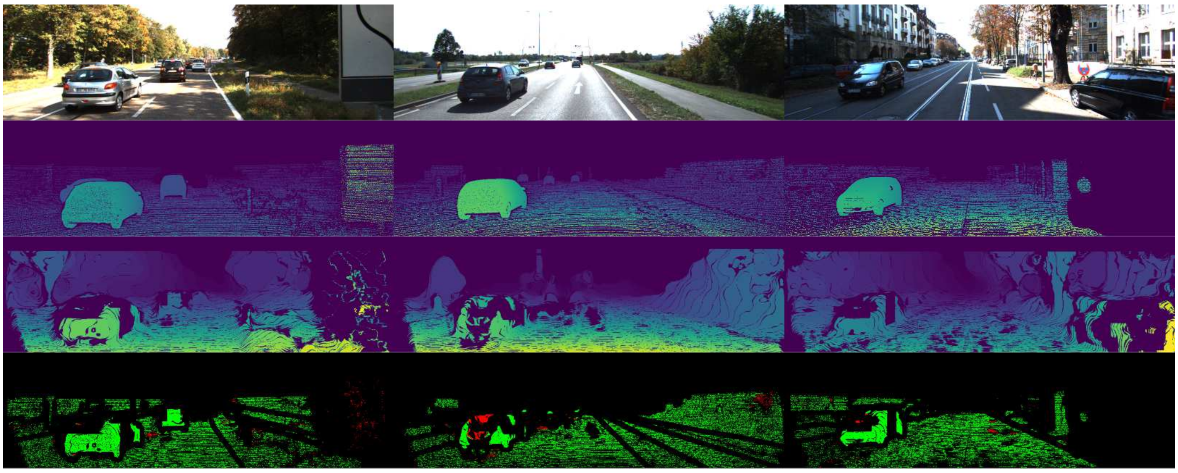

Figure 6.7: The visualization shows McCNN pseudo-labels with kernel filtering applied to car instances. The parameters are $t = 0.40$, $ks = 30$, and $ct = 2$. The first row displays the left reference image, the second row the ground-truth disparity maps, the third row the disparity map with pseudo-labels, and the fourth row the error map where green pixels are marked as accurate and red as inaccurate. The accurate pixels are those where the disparity is within the interval ±3 pixels, known as the 3px error.

6.4.6 Lowe ratio filtering

In our experiments, we applied the Lowe ratio filtering technique to enhance the accuracy of disparity estimation using McCNN embeddings. The objective was to filter out unreliable matches and improve the robustness of the disparity maps.

We utilized McCNN to extract embeddings from both the reference and target images. A cost volume was then computed by comparing these embeddings, representing the similarity scores across pixels.

The Lowe ratio test was applied to each pixel in the reference image:

$$\frac{s_1}{s_2} < q \tag{6.4}$$

where s_1 and s_2 denote the similarity scores of the best and second-best disparity in the target image, respectively, and q is a threshold parameter. We conducted experiments varying the threshold q to analyze its effect on the metrics.

The Lowe ratio filtering technique effectively enhanced the reliability of disparity estimation using McCNN embeddings. It demonstrated sensitivity to parameter settings, particularly the threshold q, which influenced the balance between match filtering and accuracy improvement in stereo vision tasks.

Method	**EPE - all**	**EPE - noc**	**D1 - all [%]**	**D1 - noc [%]**
Reversing PSM-Net [1]	1.00		3.85	
Flow2Stereo [22]	1.34	1.31	6.13	5.93
DDFlow [23]	1.23	1.21	6.37	6.17
McCNN Lowe q=1.03	1.15	1.13	5.25	5.12
McCNN Lowe q=1.05	1.15	1.13	5.23	5.07
McCNN Lowe q=1.10	1.15	1.14	5.25	5.08
McCNN Lowe q=1.20	1.20	1.17	5.64	5.43

Table 6.7: Evaluation metrics on the KITTI 2015 dataset demonstrate the performance of RAFT-Stereo models trained on different pseudo-labels filtered with Lowe ratio test, with the threshold q set to different values. The first three rows are other self-supervised methods. We evaluated on ground-truth disparity maps using *all* valid pixels and only *noc* (non-occluded) valid pixels.

The Lowe ratio test significantly improved the RAFT-Stereo performance, as shown in table 6.7 and figure 6.9. However, similar to Aleotti pseudo-labels, it still exhibits poor performance in low-texture areas.

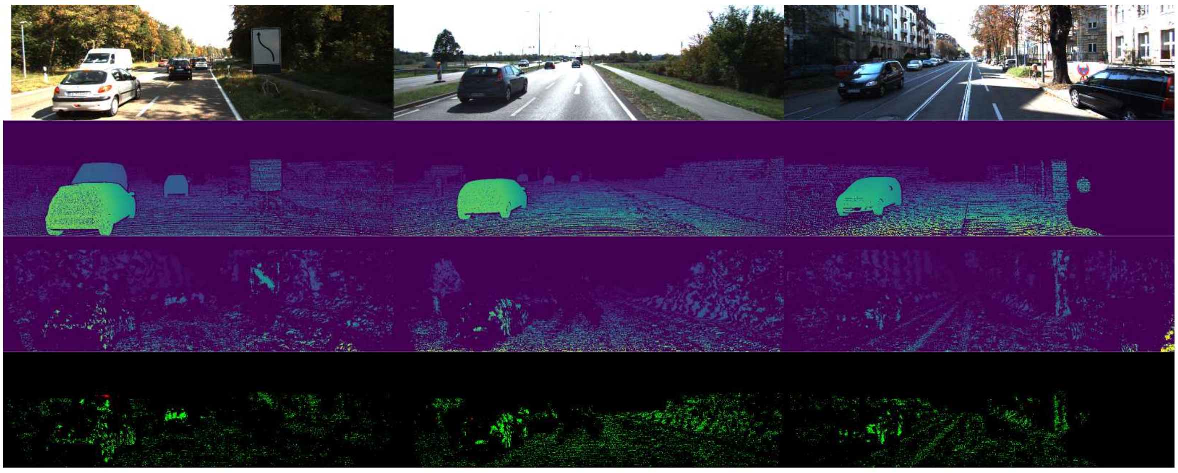

Figure 6.8: The visualization shows McCNN pseudo-labels filtered with the Lowe ratio test with the parameter set to $q = 1.05$. The first row displays the left reference image, the second row the ground-truth disparity maps, the third row the disparity map with pseudo-labels, and the fourth row the error map where green pixels are marked as accurate and red as inaccurate. The accurate pixels are those where the disparity is within the interval ±3 pixels, known as the 3px error.

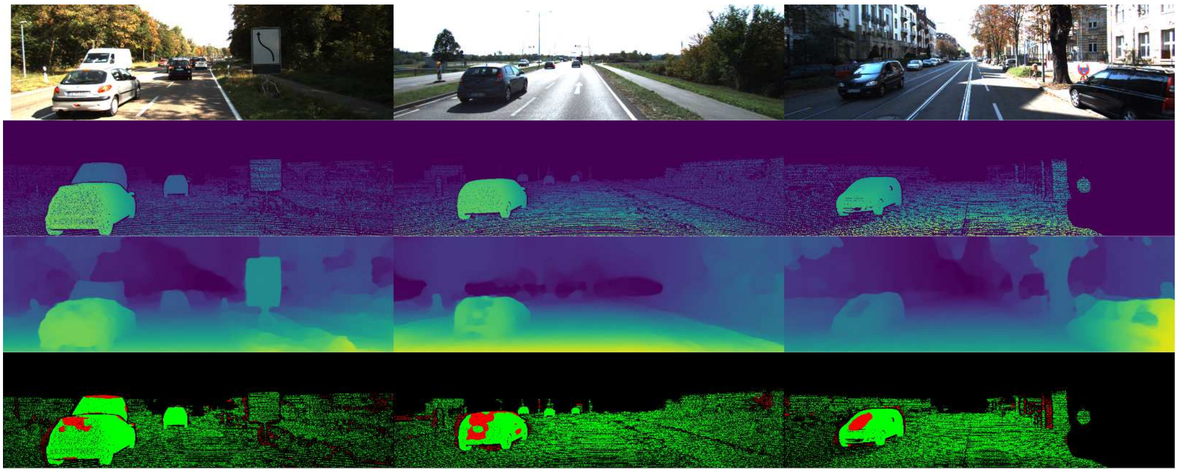

Figure 6.9: The visualization shows the performance of the RAFT-Stereo model trained on McCNN pseudo-labels filtered with the Lowe ratio test with the parameter set to $q = 1.05$. The first row displays the left reference image, the second row the ground-truth disparity maps, the third row the predicted disparity map from the RAFT-Stereo model, and the fourth row the error map where green pixels are marked as accurate and red as inaccurate. The accurate pixels are those where the disparity is within the interval ±3 pixels, known as the 3px error.

Examining the pseudo-labels in figure 6.8, we observe that almost all inaccurate pixels have been successfully removed, aligning closely with the Aleotti pseudo-labels. There remains only a small area for improvement, particularly in object boundary regions.

Additionally, we conducted an experiment to further optimize hyperparameters. Specifically, we explored different kernel sizes for median filtering and evaluated the impact of incorporating left-right consistency checks.

Method	**EPE - all**	**EPE - noc**	**D1 - all [%]**	**D1 - noc [%]**
Reversing PSM-Net [1]	1.00		3.85	
Flow2Stereo [22]	1.34	1.31	6.13	5.93
DDFlow [23]	1.23	1.21	6.37	6.17
McCNN Lowe q=1.05 median=15	1.18	1.16	5.46	5.33
McCNN Lowe q=1.05 median=25	1.14	1.13	5.22	5.09
McCNN Lowe q=1.05 median=35	1.15	1.13	5.23	5.07

Table 6.8: Evaluation metrics on the KITTI 2015 dataset demonstrate the performance of RAFT-Stereo models trained on different pseudo-labels filtered with the Lowe ratio test, with the threshold $q = 1.05$ and different parameters for median filtering. The first three rows are other self-supervised methods. We evaluated on ground-truth disparity maps using *all* valid pixels and only *noc* (non-occluded) valid pixels.

Method	**EPE - all**	**EPE - noc**	**D1 - all [%]**	**D1 - noc [%]**
Reversing PSM-Net [1]	1.00		3.85	
Flow2Stereo [22]	1.34	1.31	6.13	5.93
DDFlow [23]	1.23	1.21	6.37	6.17
McCNN Lowe q=1.05 lrcons=True	1.15	1.13	5.23	5.07
McCNN Lowe q=1.05 lrcons=False	1.45	1.42	6.92	6.76
McCNN Lowe q=1.10 lrcons=True	1.15	1.14	5.25	5.08
McCNN Lowe q=1.10 lrcons=False	1.43	1.40	6.89	6.69

Table 6.9: Evaluation metrics on the KITTI 2015 dataset demonstrate the performance of RAFT-Stereo models trained on different pseudo-labels filtered with the Lowe ratio test, with the threshold $q = 1.05$ and varying existence of left-right consistency check. The first three rows are other self-supervised methods. We evaluated on ground-truth disparity maps using *all* valid pixels and only *noc* (non-occluded) valid pixels.

In tables 6.8 and 6.9, it can be observed that choosing parameters for median filtering results in different, but not significantly better overall performance. However, incorporating the left-right consistency check is crucial, as it effectively removes a considerable number of inaccurate disparities.

6.4.7 Modified Lowe ratio filtering

While the original Lowe ratio test provided significant improvements, we proposed a modified version to further refine the selection of reliable pseudo-labels. In this modified approach, our intention was to consider the second-best match from the entire search space, excluding a specific interval around the best similarity score disparity. Specifically, we excluded the interval between the best similarity score disparity ± 10 disparities.

By doing so, we aimed to avoid selecting second-best matches that were too close to the first-best match in terms of disparity, which could lead to false positives in regions with similar but incorrect matches. For each embedding in the reference image, the first-best match in the target image is identified based on the similarity scores within the cost volume. The interval of ± 10 disparities around this first-best match is excluded, and then the second-best match is identified from the remaining search space.

The modified Lowe ratio test is applied by comparing the similarity scores of the first-best match (s_1) and the second-best match (s_2), where s_2 is chosen from the non-excluded interval. A match is considered valid if the ratio $\frac{s_1}{s_2}$ is below the threshold.

Method	**EPE - all**	**EPE - noc**	**D1 - all [%]**	**D1 - noc [%]**
Reversing PSM-Net [1]	1.00	–	3.85	–
Flow2Stereo [22]	1.34	1.31	6.13	5.93
DDFlow [23]	1.23	1.21	6.37	6.17
McCNN Lowe+ q=1.50	1.16	1.14	5.43	5.29
McCNN Lowe+ q=1.65	1.15	1.13	5.31	5.17
McCNN Lowe+ q=1.80	1.15	1.13	5.33	5.20

Table 6.10: Evaluation metrics on the KITTI 2015 dataset demonstrate the performance of RAFT-Stereo models trained on different pseudo-labels filtered with modified Lowe ratio test, with the threshold q set to different values. The first three rows are other self-supervised methods. We evaluated on ground-truth disparity maps using *all* valid pixels and only *noc* (non-occluded) valid pixels.

Table 6.10 summarizes the experimental results with different thresholds (q) applied to the modified Lowe ratio filtering approach. It suggests that this modification does not improve the overall performance of the RAFT-Stereo model. Figure 6.10 shows that the pseudo-labels are denser compared to the experiment with the classic Lowe ratio test, but they also exhibit slightly more inaccurate disparities. From figure 6.11, it can be concluded that denser disparity maps do not lead to better overall performance. These inaccurate disparities appear to significantly impact the model's performance.

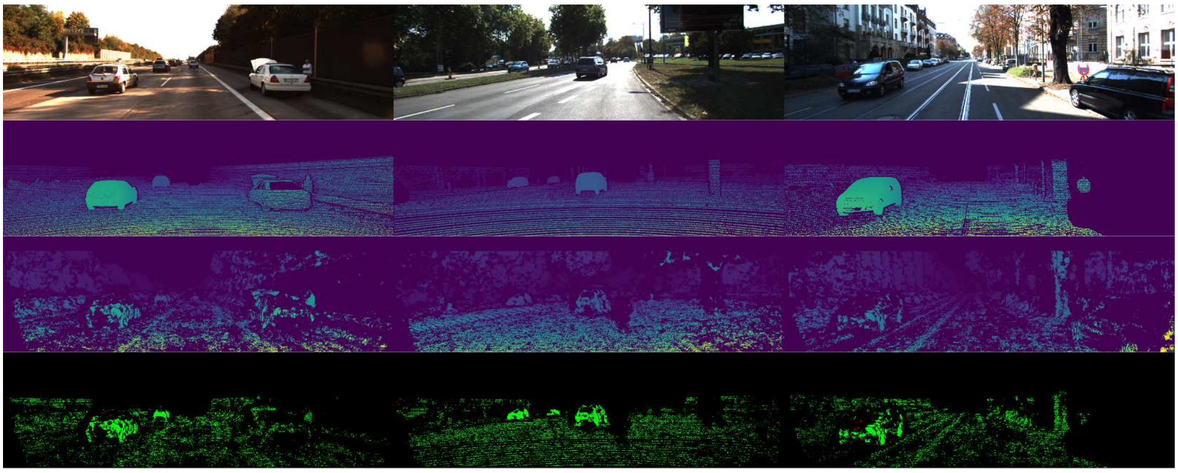

Figure 6.10: The visualization shows McCNN pseudo-labels filtered with a modified Lowe ratio test with the parameter set to $q = 1.50$. The first row displays the left reference image, the second row the ground-truth disparity maps, the third row the disparity map with pseudo-labels, and the fourth row the error map where green pixels are marked as accurate and red as inaccurate. The accurate pixels are those where the disparity is within the interval ±3 pixels, known as the 3px error.

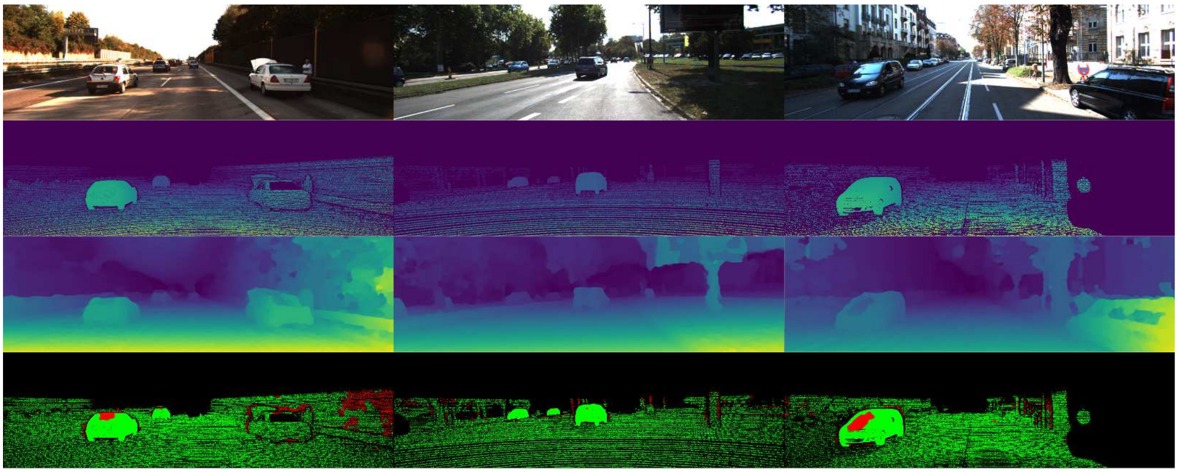

Figure 6.11: The visualization shows the performance of the RAFT-Stereo model trained on McCNN pseudo-labels filtered with a modified Lowe ratio test with the parameter set to $q = 1.50$. The first row displays the left reference image, the second row the ground-truth disparity maps, the third row the predicted disparity map from the RAFT-Stereo model, and the fourth row the error map where green pixels are marked as accurate and red as inaccurate. The accurate pixels are those where the disparity is within the interval ±3 pixels, known as the 3px error.

7 Conclusion

This research explores a self-supervised approach to stereoscopic reconstruction through pseudo-labeling, based on the work of [1] and [2], demonstrating that enhanced filtering techniques can significantly improve the accuracy of pseudo-labels and the performance of supervised stereo models. Traditional methods for stereoscopic reconstruction have been largely replaced by deep learning-based methods due to their ability to learn correspondence metrics directly from data.

However, these methods often require large amounts of labeled data, which can be costly and complex to obtain. To address this, we re-evaluated the method proposed by [1] and introduced a new, simpler self-supervised training procedure based on pseudo-labels derived from McCNN embeddings ([2], [16]).

This work investigates various filtering techniques to refine pseudo-labels generated by a weakly supervised McCNN model. Key techniques include utilizing a distance transform operator to remove problematic object borders and experimenting with excluding certain semantic classes to assess their influence on model performance. We identified the issue of discontinuity in disparity estimation in car instances and proposed applying kernel filtering, which reduced the number of inaccurate pseudo-labels. However, challenges remained in low-texture or reflective areas. Previous experiments highlighted the importance of removing inaccurate pseudo-labels, leading us to propose filtering within the scope of McCNN predictions by employing the Lowe ratio and its modified version, both aimed at improving the precision of pseudo-labels.

The results indicate that denser initial correspondences do not necessarily lead to better overall performance. Inaccurate disparities within these maps can significantly impact the model's performance, emphasizing the need for effective filtering. The best performance of McCNN pseudo-labels was achieved through aggressive filtering techniques like the Lowe ratio test, resulting in disparity maps with a minimal number of inaccuracies. However, the final RAFT-Stereo model trained on McCNN pseudo-labels still encountered issues such as discontinuities in predictions for car instances due to reflective areas and low-texture regions like vegetation. The performance of the model trained on McCNN pseudo-labels was slightly worse than that trained on Aleotti pseudo-labels. Visualization of error maps helped identify problems in predictions for small objects, such as car signs or object borders, suggesting areas for improvement to fully bridge the gap between these pseudo-labels.

Future work could focus on enhancing the RAFT-Stereo model by incorporating panoptic predictions. This could include Mask2Former predictions obtained from our previous experiments and adding additional elements to the loss function of the RAFT-Stereo model, as seen in previous work [24]. They incorporated additional loss elements which arises from the observation that areas in the image where boundaries of panoptic instances occur, such as changes in class or instance, often exhibit sharp edges and disparities in disparity maps. This loss aims to emphasize differences in disparities for neighboring pixels in the panoptic instances.

References

[1] F. Aleotti, F. Tosi, L. Zhang, M. Poggi, and S. Mattoccia, “Re- versing the cycle: self-supervised deep stereo through enhanced monocular dis- tillation,” *16th European Conference on Computer Vision (ECCV)*, 2020.

[2] J. Bratulić, *Samonadzirano učenje stereoskopske rekonstrukcije*, 2023.

[3] R. Fan, J. Jiao, H. Ye, Y. Yu, I. Pitas, and M. Liu, “Key ingredients of self-driving cars,” 2019.

[4] A. Kosaka and A. Kak, *Stereo vision for industrial applications.* Handbook of Industrial Robotics, 2019.

[5] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms.” [Online]. Available: https://vision.middlebury.edu/stereo/taxonomy-IJCV.pdf

[6] P. Liu, I. King, M. R. Lyu, and J. Xu, “Flow2stereo: Effective self-supervised learning of optical flow and stereo matching,” 2020.

[7] J. Zbontar and Y. LeCun, *Computing the stereo matching cost with a convo- lutional neural network.* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[8] J. Žbontar and Y. LeCun, “Stereo matching by training a convolutional neural network to compare image patches,” 2015. [Online]. Available: http://arxiv.org/abs/1510.05970

[9] i. J. D. Lahav Lipson, Zachary Teed, “Multilevel recurrent field transforms for stereo matching,” 2021. [Online]. Available: https://arxiv.org/abs/2109.07547

[10] Z. T. i Jia Deng, "Recurrent all-pairs field transforms for optical flow," 2020. [Online]. Available: https://arxiv.org/abs/2003.12039

[11] X. Fan, J. Lei, J. Liang, Y. Fang, X. Cao, and N. Ling, "Unsupervised stereoscopic image retargeting via view synthesis and stereo cycle consistency losses," *Neurocomputing*, vol. 447, pp. 161–171, 2021. https://doi.org/https://doi.org/10.1016/j.neucom.2021.02.079

[12] T. Shen, Z. Luo, L. Zhou, H. Deng, R. Zhang, T. Fang, and L. Quan, "Beyond photometric loss for self-supervised ego-motion estimation," *CoRR*, vol. abs/1902.09103, 2019. [Online]. Available: http://arxiv.org/abs/1902.09103

[13] F. Tosi, F. Aleotti, M. Poggi, and S. Mattoccia, "Learning monocular depth estimation infusing traditional stereo knowledge," *CoRR*, vol. abs/1904.04144, 2019. [Online]. Available: http://arxiv.org/abs/1904.04144

[14] F. Tosi, M. Poggi, A. Tonioni, L. Di Stefano, and S. Mattoccia, "Learning confidence measures in the wild," 01 2017. https://doi.org/10.5244/C.31.133

[15] J. Chang and Y. Chen, "Pyramid stereo matching network," *CoRR*, vol. abs/1803.08669, 2018. [Online]. Available: http://arxiv.org/abs/1803.08669

[16] N. Markuš, I. S. Pandžić, and J. Ahlberg, "Learning local descriptors by optimizing the keypoint-correspondence criterion," *2016 23rd International Conference on Pattern Recognition (ICPR)*, pp. 2380–2385, 2016. https://doi.org/10.1109/ICPR.2016.7899992

[17] M. Combalia and V. Vilaplana, "Monte-carlo sampling applied to multiple instance learning for histological image classification," *CoRR*, vol. abs/1812.11560, 2018. [Online]. Available: http://arxiv.org/abs/1812.11560

[18] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," 2022.

[19] B. Cheng, A. G. Schwing, and A. Kirillov, "Per-pixel classification is not all you need for semantic segmentation," *CoRR*, vol. abs/2107.06278, 2021. [Online]. Available: https://arxiv.org/abs/2107.06278

[20] D. Lowe, “Distinctive image features from scale-invariant keypoints,” 2004. [Online]. Available: https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf

[21] M. Menze and A. Geiger, “Object scene flow for autonomous vehicles,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[22] P. Liu, I. King, M. R. Lyu, and J. Xu, “Flow2stereo: Effective self-supervised learning of optical flow and stereo matching,” *CoRR*, vol. abs/2004.02138, 2020. [Online]. Available: https://arxiv.org/abs/2004.02138

[23] I. King, M. R. Lyu, and J. Xu, “Ddflow: Learning optical flow with unlabeled data distillation,” *CoRR*, vol. abs/1902.09145, 2019. [Online]. Available: http://arxiv.org/abs/1902.09145

[24] F. Saeedan and S. Roth, “Boosting monocular depth with panoptic segmentation maps,” in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 3852–3861. https://doi.org/10.1109/WACV48630.2021.00390

Abstract

Self-supervised learning of stereoscopic reconstruction through pseudo-labeling

David Kerman

This work introduces a self-supervised approach to stereoscopic reconstruction through pseudo-labeling, based on the previous studies. We provide an overview of the fundamental principles of stereoscopic reconstruction and relevant deep learning models. The methodology includes a detailed description of self-supervised learning via pseudo-labeling, along with our innovative filtering techniques designed to enhance the accuracy of pseudo-labels. We demonstrate the performance improvements in the supervised stereo model achieved using these refined pseudo-labels. Experimental results are presented to validate the effectiveness of our approach. Finally, we identified problems with the current method and proposed changes which could potentially improve model accuracy.

Keywords: stereoscopic reconstruction; deep learning; self-supervised learning; computer vision; pseudo-labeling

Sažetak

Samonadzirano učenje stereoskopske rekonstrukcije pseudooznačavanjem

David Kerman

Ovaj rad uvodi samonadzirani pristup učenju stereoskopske rekonstrukcije pseudooznačavanjem, temeljen na prethodnim radovima. Pružamo pregled temeljnih principa stereoskopske rekonstrukcije i relevantnih modela dubokog učenja. Rad uključuje detaljan opis samonadziranog učenja pseudooznačavanjem, zajedno s našim inovativnim tehnikama filtriranja osmišljenim za poboljšanje točnosti pseudooznaka. Nadalje, demonstriramo poboljšanje performansi nadziranog stereoskopskog modela postignuto korištenjem rafiniranih pseudooznaka. Eksperimentalni rezultati su predstavljeni kako bi se potvrdila učinkovitost našeg pristupa. Identificirani su problemi i predložene modifikacije koje bi poboljšale trenutnu metodu.

Ključne riječi: stereoskopska rekonstrukcija; duboko učenje; samonadzirano učenje; računalni vid; pseudooznačavanje