LSM: Learning Subspace Minimization for Low-level Vision

Chengzhou Tang[1]   Lu Yuan[2]   Ping Tan[1]
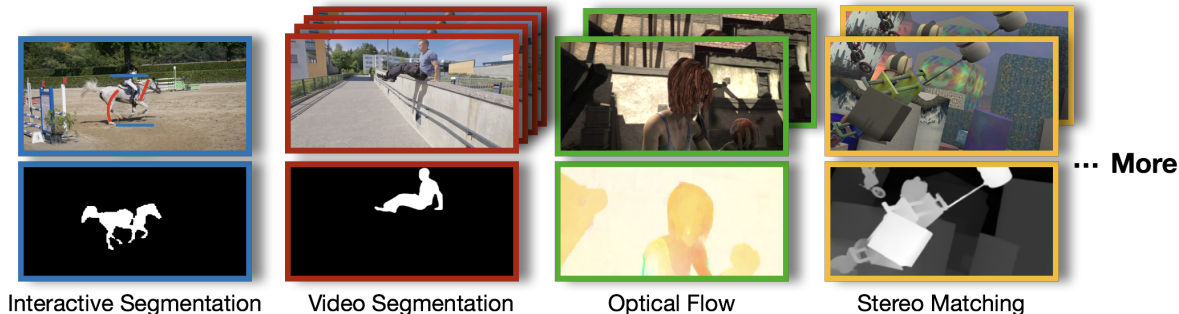[1]Simon Fraser University   [2]Microsoft

Figure 1: Learning subspace minimization solves various low-level vision tasks using *unified* network structures and parameters.

Abstract

We study the energy minimization problem in low-level vision tasks from a novel perspective. We replace the heuristic regularization term with a learnable subspace constraint, and preserve the data term to exploit domain knowledge derived from the first principle of a task. This learning subspace minimization (LSM) framework unifies the network structures and the parameters for many low-level vision tasks, which allows us to train a single network for multiple tasks simultaneously with completely shared parameters, and even generalizes the trained network to an unseen task as long as its data term can be formulated. We demonstrate our LSM framework on four low-level tasks including interactive image segmentation, video segmentation, stereo matching, and optical flow, and validate the network on various datasets. The experiments show that the proposed LSM generates state-of-the-art results with smaller model size, faster training convergence, and real-time inference.

1. Introduction

Many low-level vision tasks (*e.g.* image segmentation [50, 75, 23], video segmentation [48, 73, 41], stereo matching [63, 5, 59] and optical flow [28, 8, 69]) are conventionally formulated as an energy minimization problem:

$$\min_{\boldsymbol{x}} D(\boldsymbol{x}) + R(\boldsymbol{x}), \qquad (1)$$

where $\boldsymbol{x}$ is the desired solution (*e.g.* a disparity field for stereo matching), and the two terms $D(\boldsymbol{x})$ and $R(\boldsymbol{x})$ are the data term and regularization term respectively. The data term $D(\boldsymbol{x})$ is usually well designed following the first principle

of a task, such as the color consistency assumption in stereo and optical flow. However, the regularization term $R(\boldsymbol{x})$ is often heuristic. Typically, it regularizes $\boldsymbol{x}$ at the *pixel-level* and encourages similar pixels to have similar solution values. The regularization term is necessary because low-level vision tasks are usually ill-posed [60], and a standalone data term is often insufficient, *e.g.* the aperture problem in optical flow.

However, a vanilla L^2 smoothness regularization [72] may cause over-smoothed results at object boundaries. Ideally, the regularization term should smooth out noises in $\boldsymbol{x}$ and preserve sharp edges. Thus, many edge-preserving regularization terms have been developed, such as the Total Variation (TV) regularization [66, 10], the anisotropic diffusion [57], the bilteral filter [54] which focuses on designing better similarity measurements between pixels, and the distance in a learned feature embedding space [65, 68, 44] has also been adopted for the same purpose. But it is still an unsolved problem to design an ideal similarity measurement for efficient and accurate energy minimization.

We study this energy minimization problem from a different perspective. Instead of focusing on the *pixel-level* similarity, we exploit *image-level* context information. Specifically, we preserve the data term $D(\boldsymbol{x})$ but replace the heuristic regularization term $R(\boldsymbol{x})$ with a subspace constraint:

$$\min_{\boldsymbol{x}} D(\boldsymbol{x}), \text{ s.t. } \boldsymbol{x} \in \mathcal{V} = \text{span}\{\boldsymbol{v}_1, \cdots \boldsymbol{v}_K\}, \qquad (2)$$

where $\mathcal{V}$ is a K-dimensional subspace, and $\{\boldsymbol{v}_1, \cdots \boldsymbol{v}_K\}$ is the corresponding basis vectors. Our motivation is different from the regularization term $R(\boldsymbol{x})$: we use the *image-level* context information to regularize the problem by assuming the desired solution $\boldsymbol{x}$ is composited of several layers [15, 79, 80], *e.g.* motion layers for optical flow, and each basis vector $\boldsymbol{v}_k$ will correspond to one of these lay-

ers. Therefore, we can represent the solution $\boldsymbol{x}$ as a linear combination of these basis vectors and solve the combination coefficients, leading to a compact minimization that *not only* is efficient *but also* enables end-to-end training and outperforms the conventional regularization term $R(\boldsymbol{x})$.

To this end, we propose the *learning subspace minimization* (LSM) framework that progressively evolves $\mathcal{V}$ and solves Eq. (2) on a feature pyramid coarse-to-fine. At each pyramid level, we employ a convolutional neural network (CNN) to update $\mathcal{V}$ from both the image features and the derivatives of the data term $D(\boldsymbol{x})$ respect to the intermediate solution $\boldsymbol{x}$. Since the generation of $\mathcal{V}$ receives the task-specific data term as the *input*, it decouples the task-specific characteristics from the subspace generation and unifies the network structures and the parameters for various tasks.

As a consequence, our LSM framework enables joint multi-task learning with *completely* shared network structures as well as parameters, and even makes zero-shot task generalization possible, where a trained network is plug-and-play for an unseen task without any parameter modification, as long as the corresponding data term $D(\boldsymbol{x})$ is formulated.

In the experiments, we implement four low-level vision tasks in an unified paradigm, including interactive image segmentation, video segmentation, stereo matching and optical flow. Our LSM framework has achieved better or comparable results with state-of-the-art methods. Our network structures and parameters can be unified into a compact model, which yields higher efficiency in training and inference. We also demonstrate zero-shot task generalization by leaving one task out for testing and train on the other tasks. All these benefits come from our methodology that integrates domain knowledge (*i.e.* minimizing a data term derived from the first principle) with convolutional neural networks (*i.e.* learning to generate subspace constraint).

2. Related Works

Regularization in Variational Method Many computer vision problems can be formulated to Eq. (1). We only review the continuous settings (*i.e.* variational method) because it is more relevant to our work and refer readers to [36] for the review about the discrete settings. One of the main focuses of these works is on designing appropriate objective function, especially the regularization term. Rudin and Osher [66] first proposed the TV regularization for image denoising, which has also been proven to be successful for image super-resolution [3], interactive image segmentation [75], stereo matching [63], optical flow [89, 81], multi-view stereo [40], *etc*. Perona and Malik [57] pioneered to use partial differential equations (PDE) for anisotropic diffusion, which is equivalent to minimizing an energy function with edge-aware regularization [12, 54]. Non-local regularizations [58] have also been proposed for image super-resolution [62], image inpainting [2], optical flow [38], *etc*,

which performs better by connecting longer range pixels but is usually computational expensive.

Our LSM framework also minimizes an objective function. But we only preserve the data term since it is usually derived from the first principle of a task, and replace the heuristic regularization term to a learned subspace constraint that captures the structure of the desired solution at the whole image context level and enables end-to-end training to boost the performance from data.

Convolutional Neural Networks Inspired by the success of CNNs in high-level tasks [39, 67, 26], numerous CNN based methods have been proposed for low-level vision tasks. Dong et al. [18] pioneered to use a CNN to upsample image patches for super-resolution. Zbontar and LeCun [90] and Luo et al. [47] used CNN features to measure image patches' similarity for stereo matching, Xu et al. [84] and Bailer et al. [4] also used CNN based similarity for optical flow. All these methods used CNNs in the patch level, which is computationally expensive and requires post-processing to composite the final result. So more recent works used whole images as inputs. Dosovitskiy et al. [19] used an encoder-decoder structure for optical flow, which is then extended to stereo matching [49] and further evolved in Ilg et al. [33, 32] and other works [70, 11, 87]. Some recent works [85, 42, 34] enabled interactive image segmentation by feeding an image and an user annotation map to CNNs. Meanwhile, CNN based methods [29, 30, 77, 56] have also achieved leading performance for video segmentation.

Our LSM framework also employs CNNs but for different purposes. Instead of predicting the solution directly, we use CNNs to constraint the solution onto a subspace to facilitate the minimization of the data term. The data term is derived from the first principle of each task and decouples the task-specific formulation from the network parameters. Therefore, our framework unifies the network structures as well as parameters for different tasks and even enables zero-shot task generalization, which are difficult for fully CNN based methods. Although some recent works [6, 71] also learn to generate subspace via CNNs, they are designed specifically for 3D reconstruction, which is ad-hoc and unable to generalize to broader low-level vision tasks.

3. Learning Subspace Minimization

3.1. Overview

As illustrated in Fig. 2(a), we first build a feature pyramid $\mathcal{F}$ for *each* image I from a set, where the number of images in a set depends on the task, *e.g.* the interactive segmentation is defined on a single image, the stereo matching and the optical flow are defined on two images, and the video segmentation processes three or more images. The output of the pyramid $\mathcal{F}$ are feature maps in four levels $\{F^1, F^2, F^3, F^4\}$ with strides $\{32, 16, 8, 4\}$ and channels $\{512, 256, 128, 64\}$

(a) Coarse-to-fine on feature pyramid(s).　　　　(b) A single iteration of the learning subspace minimization.
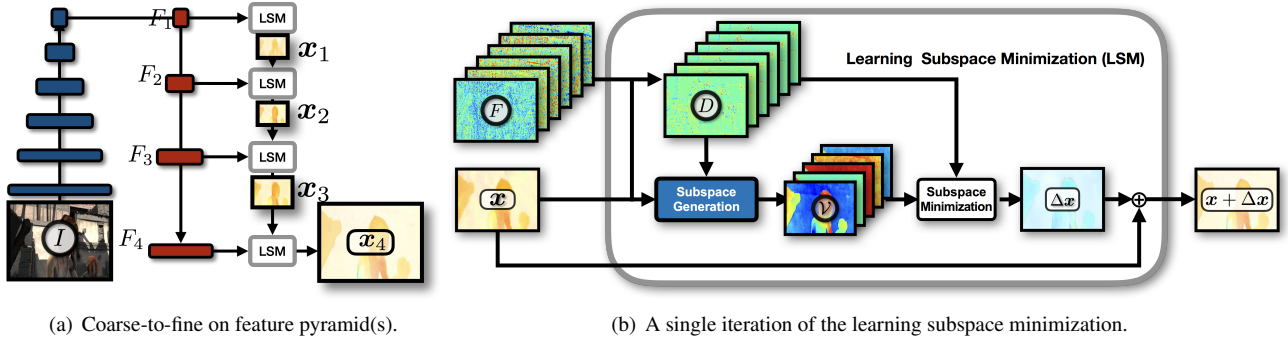
Figure 2: Overview of our learning subspace minimization framework.

respectively, which are constructed by similar strategy as FPN [43] but using DRN-22 [88] as the backbone network.

At each pyramid level, we define the data term $D(\boldsymbol{x})$ of a task on CNN features (Sec. 3.4) and solve Eq. (2). $D(\boldsymbol{x})$ is approximated using the second-order Taylor expansion at the intermediate solution $\boldsymbol{x}$ and yields the following quadratic minimization problem:

$$\min_{\Delta \boldsymbol{x}} \frac{1}{2} \Delta \boldsymbol{x}^\top \boldsymbol{D} \Delta \boldsymbol{x} + \boldsymbol{d}^\top \Delta \boldsymbol{x}, \qquad (3)$$

where $\boldsymbol{D}$ is the matrix that contains the (approximated) second-order derivatives $\frac{\partial^2 D}{\boldsymbol{x}^2}$ of the data term, $\boldsymbol{d}$ is the vector that contains the first-order derivatives $\frac{\partial D}{\partial \boldsymbol{x}}$, and $\Delta \boldsymbol{x}$ is the desired incremental solution. The structure of $\boldsymbol{D}$ is task dependent: it is a diagonal matrix for one-dimensional tasks or block diagonal for multi-dimensional tasks.

To maintain the subspace constraint of Eq. (2), we represent the incremental solution $\Delta \boldsymbol{x}$ as a linear combination of a set of underlying basis vectors, *i.e.* $\Delta \boldsymbol{x} = c_1 \boldsymbol{v}_1 + c_2 \boldsymbol{v}_2 \cdots + c_K \boldsymbol{v}_K$, and then solve the combination coefficients $\boldsymbol{c} = [c_1, c_2 \cdots c_K]^\top$ as:

$$\min_{\boldsymbol{c}} \frac{1}{2} \boldsymbol{c}^\top (\boldsymbol{V}^\top \boldsymbol{D} \boldsymbol{V}) \boldsymbol{c} + (\boldsymbol{d}^\top \boldsymbol{V}) \boldsymbol{c}, \qquad (4)$$

where $\boldsymbol{V}$ is a dense matrix, and its columns correspond to the K basis vectors from $\mathcal{V}$. As shown in Fig. 2(b), we generate this $\mathcal{V}$ from the image and the minimization context information (Sec. 3.2), solve minimization with subspace constraint (Sec. 3.3), and move to the next pyramid level after updating the intermediate solution as $\boldsymbol{x} \leftarrow \boldsymbol{x} + \Delta \boldsymbol{x}$.

This formulation is easy and efficient to implement because multiplying the dense matrix $\boldsymbol{V}$ with the (block) diagonal matrix $\boldsymbol{D}$ can be done by column-wise product, yielding a compact K-by-K linear system, which can be solved using direct solver such as Cholesky decomposition [21], instead of iterative solvers such as conjugate gradient descent [51]. Therefore, Eq. (4) is differentiable and supports end-to-end training without unrolling or implicit gradient [17].

3.2. Subspace Generation

Before introducing the network that generates $\mathcal{V}$, we first propose two principles for the subspace generation:

- First, *the image context matters.* Standalone data terms are often insufficient for low-level vision tasks as introduced in Sec. 1, because these tasks are usually ill-posed [60, 72]. So it is necessary to consider the image context information to generate the subspace $\mathcal{V}$, which enforces each basis vector $\boldsymbol{v}_k$ to be spatially smooth except for discontinuities at object boundaries.

- Second, *the minimization context matters.* The objective function (data term) is minimized iteratively. At each iteration, the intermediate solution $\boldsymbol{x}$ is at a different location on the objective function landscape, and the local curvature of the objective function decides the direction and magnitude of the desired incremental solution $\Delta \boldsymbol{x}$ for the minimization. So it is also necessary to incorporate the minimization context into the subspace generation, which learns to narrow the gap between the estimated solution and the ground truth.

Following these two principles, we learn to generate the subspace $\mathcal{V}$ as illustrated in Fig. 3:

- First, we compute a m-channel image context from the original c-channel feature map F by 1×1 convolution, where $m = c/8$ and is $\{64, 32, 16, 8\}$ at the corresponding pyramid level. This step reduces the computation complexity for the following up procedures and balances the impact between the image context and the minimization context.

- Second, we compute a $2m$-channel minimization context. Specifically, we split the c-channel feature map(s) into m groups. Within each group, we evaluate the data term $D(x)$ with the associated feature maps, compute the first-order derivative $\frac{\partial D}{\partial \boldsymbol{x}}$ and the second-order derivatives $\frac{\partial^2 D}{\boldsymbol{x}^2}$, which approximate the objective landscape neighborhood. We concatenate these derivatives to form a $2m$-channel minimization context features.

- In the next, we normalize the intermediate solution $\boldsymbol{x}$ with its mean and variance, and concatenate the normalized $\boldsymbol{x}$, the image context, and the minimization context to form a $(3m + 1)$-channel input features for subspace generation. To aggregate the context information in
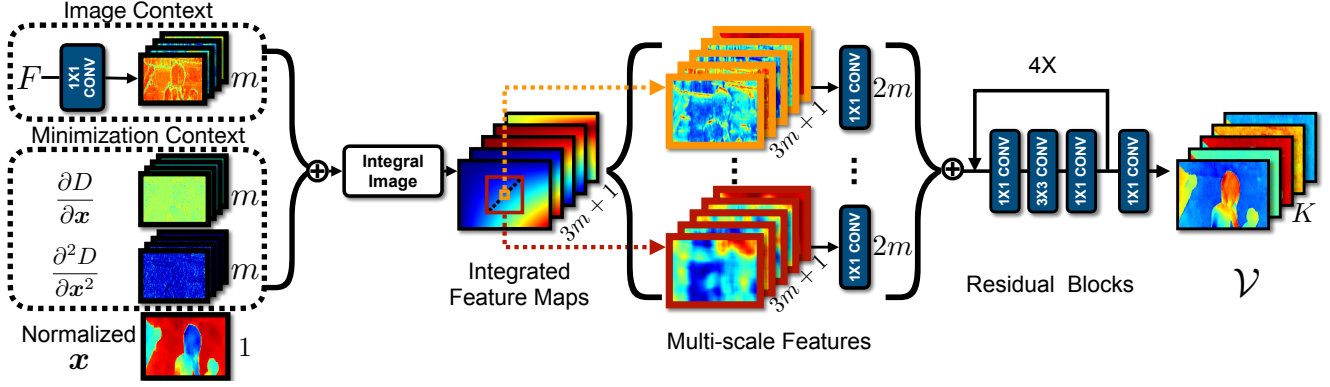
Figure 3: Subspace generation from the image and the minimization context features. The spatial sizes of all feature maps are the same to the F from a feature pyramid level. Integral image is used for the efficient construction of the multi-scale features.

multi-scale, we average pool the context features in 4 different kernel sizes without stride, which maintains the spatial size of a feature map. Specifically, we first compute the integral images [76, 25] of the context features and then average neighboring features at each pixel coordinate, which gives better efficiency.

- Finally, we apply a 1×1 convolution to project a feature map to $2m$-channel at each scale individually and concatenate them to get the $8m$-channel multi-scale features. Therefore, we can generate the K-dimensional subspace $\mathcal{V}$ from the multi-scale features via four residual blocks [26] followed by a 1×1 convolution, and K is $\{2, 4, 8, 16\}$ at the corresponding pyramid level.
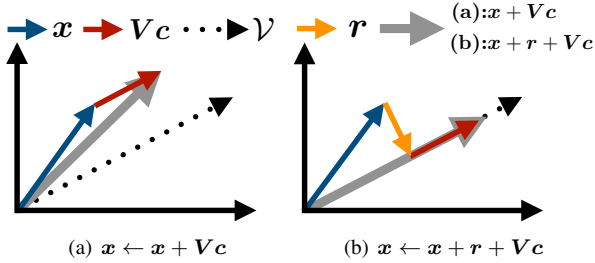
3.3. Subspace Minimization

Figure 4: A 2D example where (a): the subspace constraint is violated, *i.e.* the 2D vector $x + Vc$ is not on the 1D subspace $\mathcal{V}$, and (b): the subspace constraint is maintained, *i.e.* $x + r + Vc$ is on $\mathcal{V}$, by considering the residual r between x and its projection on $\mathcal{V}$.

After the subspace $\mathcal{V}$ is generated, we can solve Eq. (4) directly as $c = -(V^\top D V)^{-1} V^\top d$ because $V^\top D V$ is positive-definite by definition, and update the current intermediate solution as $x \leftarrow x + Vc$. However, it will violate the subspace constraint as shown in Fig. 4(a), because the subspace $\mathcal{V}$ is generated progressively, i.e, the current solution x belongs to the subspace from last iteration but is not guaranteed to be on the newly generated $\mathcal{V}$, so is $x + Vc$. To address this issue, we propose to project x onto the current subspace $\mathcal{V}$ and reformulate Eq. (4) as follows:

- Denoting $P = V(V^\top V)^{-1} V^\top$ is the projection matrix that projects an arbitrary vector onto the the subspace $\mathcal{V}$, we can compute its projection onto $\mathcal{V}$ as $x' = Px$, and the residual vector from x to x' is $r = (P - I)x$.

- Theoretically, we can reevaluate D and d respect to x' and solve Eq. (4), but it requires extra computation. So we reparameterize the incremental solution Δx as $r + Vc$ and transform Eq. (4) into:

$$\min_c \frac{1}{2}(r + Vc)^\top D(r + Vc) + d^\top(r + Vc), \quad (5)$$

where we can compute c as

$$c = -(V^\top D V)^{-1} V^\top (d + Dr) \quad (6)$$

without recomputing D and d, and update x as $x + r + Vc$ as shown in Fig. 4(b).

3.4. Applications

We now show how the proposed LSM framework unifies various low-level vision tasks. We implement four tasks for demonstration, and only introduce the data term for each task. For *all* tasks, we initialize x as a zero vector. According to the difference of data term formulation, these tasks are divided to two categories.

In the first category, we introduce two binary image labeling tasks: interactive segmentation and video segmentation, both of which share the same formulation as:

$$D(x) = \sum_p \alpha_p \|\tau(x_p) - 1\|_2^2 + \beta_p \|\tau(x_p) + 1\|_2^2, \quad (7)$$

where $p = [x, y]$ is a pixel coordinate, $\tau(\cdot)$ is an activation function to relax and constrain the binary label and constrain $\tau(x_p)$ between $(-1, +1)$, while α_p and β_p are the probabilities that $\tau(x_p) = +1$ or -1.

- For **interactive segmentation**, $\tau(x_p)$ indicates whether a pixel p is on the foreground object $(+1)$ or

background scene (-1), and the corresponding probabilities $\alpha_{\boldsymbol{p}}$ and $\beta_{\boldsymbol{p}}$ are estimated as nonparametric probabilities [82] from the foreground scribble points and the background scribble points respectively.

- For **video segmentation**, $\tau(\boldsymbol{x_p})$ indicates whether a pixel $\boldsymbol{p}$ belongs to an previously labeled foreground object $(+1)$ or not (-1), and $\alpha_{\boldsymbol{p}}$ and $\beta_{\boldsymbol{p}}$ are the corresponding average probabilities estimated from $\boldsymbol{p}$'s correlation with its foreground and background neighbors in previous labeled frames respectively.

In the second category, we introduce two dense correspondence estimation tasks on two images: stereo matching and optical flow, both of which can be formulated as:

$$D(\boldsymbol{x}) = \sum_{\boldsymbol{p}} \|F_S(\boldsymbol{p} + \boldsymbol{x_p}) - F_T(\boldsymbol{p})\|_2^2, \qquad (8)$$

where $\boldsymbol{p} = [x, y]$ is the pixel coordinate in the target (template) image T, and $\boldsymbol{x_p}$ is the warping vector that warps $\boldsymbol{p}$ to $\boldsymbol{p} + \boldsymbol{x_p}$ in the source image S. Similar to the brightness constancy assumption for image channels [28], Eq. (8) assumes that the warped feature channels F will also be consistent.

- For **stereo matching**, S and T are two images viewing the same scene. Therefore, $\boldsymbol{x_p} = [u, 0]$ only contains horizontal displacement and warps $\boldsymbol{p}$ to $[x + u, y]$ in the target image T.

- For **optical flow**, S and T are two neighboring video frames. Therefore, $\boldsymbol{x_p} = [u, v]$ is the 2D motion vector that warps $\boldsymbol{p}$ to $[x+u, y+v]$ in the S. Since optical flow is a two-dimensional labeling problem compared with stereo matching (one-dimensional, *i.e.* $\boldsymbol{x_p}$ is a scalar) and the two image labeling tasks, we apply Cramer's rule [27] to unify the network structures and parameters of optical flow with others. Please refer to the *supplementary* for more implementation details.

4. Experiments

4.1. Implementation Details

Training Loss Loss design is beyond the scope of this paper, so we use existing losses for all tasks. For *interactive segmentation* and *video segmentation*, we use the Intersection of Union (IoU) loss from Ahmed et al. [1]. For *stereo matching* and *optical flow* we use the end-point-error (EPE) loss as in DispNet [49] and FlowNet [19]. Since our solution is estimated coarse-to-fine, we downsample the ground-truth to multiple scales and sum the loss over all scales as in [70].
Hyperparamters We use AdamW optimizer [46] with the default settings where $\beta_1 = 0.9$, $\beta_2 = 0.999$. The learning rate is initialized as 3×10^{-4} and reduced during training using cosine decay [45] without warm restarts. This set of hyperparameters are fixed for all experiments.
Dataset For *interactive segmentation*, we use the PASCAL VOC Semantic Boundaries Dataset [24] for training and

the VGG interactive segmentation dataset [23] for testing, and the overlapped 99 images are excluded from the training set. For *video segmentation*, we use the DAVIS-2017 dataset [61] for training and the DAVIS-2016 [55] for testing. For *stereo matching*, we use the training/testing split of FlyingThings3D [49] from [53], and for *optical flow* we use FlyingThings3D for training and Sintel [9] for testing.
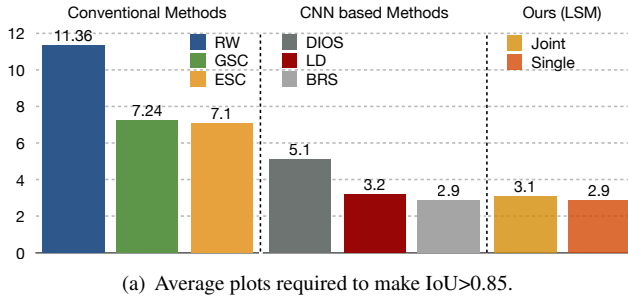
4.2. Comparison with State-of-the-art

Our framework can be applied to a low-level vision task as long as its first-order and second-order differentiable data term can be formulated. So we first test the multi-task capability of our network. Note that the *whole* network structure and *all* parameters are shared for all tasks, while previous works [13, 16, 37] *only* share the backbone and use different decoders/heads to handle different tasks.

We train our model on all four tasks jointly using a workstation with four TITAN-Xp GPUs. For implementation simplicity, we deploy one task on each GPU and update the network parameters on CPU. The batch size are 12 for interactive segmentation, 6 for video segmentation, and 4 for stereo matching and optical flow. The training runs for 143.2K iterations. To make a fair comparison with other state-of-the-art single-task methods, we also train each task individually and denote the corresponding result as 'Single', while the results of joint training are denoted as 'Joint'.
Interactive Image Segmentation For interactive segmentation, we compare our LSM framework to several conventional methods including ESC and GSC by Gulshan et al. [23], and Random walk [22], as well as recent CNN based methods Deep Object Selection (DIOS) [85], Latent Diversity (LD) [42] and Backpropagation Refinement (BRS) [34]. We randomly sample a few points from the scribbles as inputs for the CNN based methods, since they only supports clicks. We evaluate all methods by the plots required to make IoU greater than 0.85. As shown in Fig. 5(a), our method achieves better results among both recent CNN based methods and the conventional ones.

We also compare with the LD qualitatively when user only interact once. We also subsample scribbles and successively send annotations to LD for a fair comparison. Fig. 5(b) shows that our results are superior than Latent Diversity [42]. It is because the existing CNN based methods only supports spatial distance maps as inputs, which are less precise than scribbles. While our LSM supports scribbles by feature distribution estimation and Eq. (7).
Video Segmentation For video segmentation, we compare our LSM framework to several conventional minimization based methods including BVS [48], OFL [73] and DAC [41], as well as recent CNN based methods that do not require fine-tuning for a fair comparison, including MaskRNN [29], VideoMatch [30] and FEELVOS [77]. Fig. 6(a) shows that our LSM performs better than conventional methods and
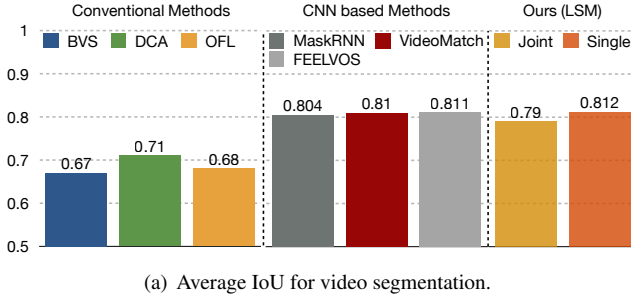
(a) Average plots required to make IoU>0.85.

(b) Our result is superior than LD [42] when user interact only once.

Figure 5: Interactive image segmentation result on VGG interactive segmentation benchmark.

comparable to CNN based methods. We also show a qualitative comparison to FEELVOS on the challenging dance-twirl sequence. As shown in Fig. 6(b), our LSM generates more false positive regions than FEELVOS [77] because the skin and the cloth colors of the dancer and the audiences are similar, but ours is able to track the dancer consistently while FEELVOS lost the dancer's torso during twirl.

(a) Average IoU for video segmentation.

(b) Our result is qualitatively comparable to FEELVOS [77].
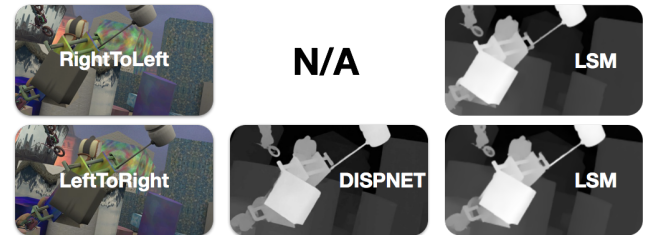
Figure 6: Video segmentation result on DAVIS 2016.

Stereo Matching For stereo matching, we compare our LSM framework with several conventional methods includ-

ing SGM [20], SPS [86] and MC-CNN [78] which uses CNN features *only* for data term evaluation in a MRF, as well as some fully CNN based methods including DispNet [49], CRL [53], PSMNet [11] and GANet [91].

When compared with other CNN based methods, our LSM is comparable for joint training and better for single-task training as shown in Fig. 7(a). As shown in Fig. 7(b), we are able to estimate both the left-to-right and the right-to-left disparities in the same accuracy because we do not assume the direction or the range of the disparity in Eq. (8). Altough [74] has been proposed to achieve range flexibility, the fully CNN based methods still only deal with single directional pairs because of the cost-volume.

(a) Average End-Point-Error for disparity.

(b) Our LSM supports both left-to-right and right-to-left stereo matching while most of fully CNN based methods only support left-to-right.

Figure 7: Stereo matching results on FlyingThings 3D.

Optical Flow For optical flow, we compare our LSM framework with conventional methods including LDOF [7], EpicFlow [64] and PCA-Layers [83] which also adopts a basis representation but the basis is static and specifically learned for optical flow using PCA [35], as well as CNN based methods including LiteFlowNet [31], PWC-Net [70], and FlowNet2-CSS [33] which is a stack of three FlowNets.

As shown in Fig. 8(a), our result are comparable to Lite-FlowNet [31] and PWC-Net [70] without refinement sub-net. FlowNet2 is more accurate by stacking networks, which is less efficient, more difficult to train, and increases the model size dramatically. Comparing with FlowNet2, our method is $12\times$ smaller in model size, $4\times$ faster in inference, and $32\times$ less in training time. Our LSM is better than LDOF [7] and PCA-Layers [83], but less accurate than EpicFlow [64]. However, conventional method are usually based on variational approaches and take 5-20 seconds to run, while our LSM takes only 25ms.
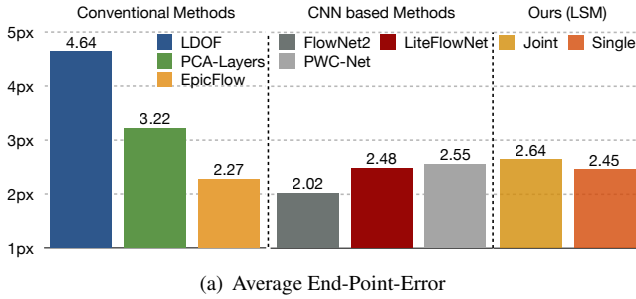
(a) Average End-Point-Error

(b) Our optical flow is comparable to PWC-Net.

Figure 8: Flow trained on Flythings3D and tested on Sintel.

4.3. Zero-shot Task Generalization

Our LSM framework even generalizes the learned network to unseen tasks. It is different from the zero-shot task transfer [52], where the network parameters are interpolated from existing tasks, and the interpolation coefficients is defined by a correlation matrix during training. In contrast, we fix the learned parameters and do not require any extra information between tasks. To demonstrate this capability, we train the network on three tasks with the same settings as the joint multi-task training, and leave one out for testing.
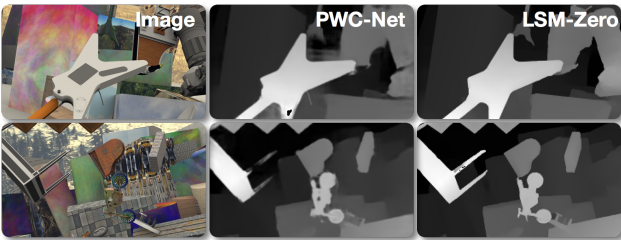

Figure 9: Our zero-shot generalized LSM model performs better than PWC-Net for stereo matching.

Stereo Matching The first task left out for testing is stereo matching. Since none of existing CNN based method supports this test, we approximate it by estimating optical flow using PWC-Net [70] on stereo image pairs, and only consider the EPE on the horizontal direction. The average EPE is 2.47 for our LSM model learned on the other three tasks

and tested on stereo matching, which is superior than the 5.29 EPE of PWC-Net as shown in Fig. 9. Note that our LSM consistently performs better than conventional methods [20, 86, 78], while PWC-Net is worse than SGM [20].
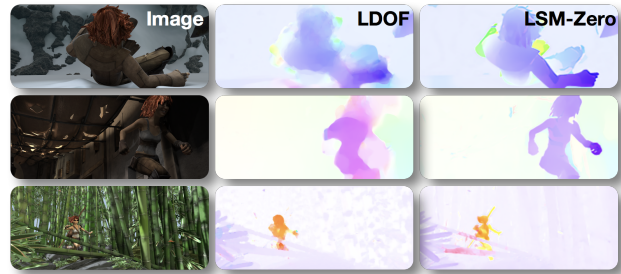

Figure 10: Our zero-shot generalized LSM model performs better than LDOF for optical flow.

Optical Flow For optical flow, none of CNN based method supports this zero-shot test, and the average EPE is 4.6 for our LSM model learned on the other three tasks, which is better than LDOF [7]. However, LDOF requires computationally expensive dense HOG [14] feature matching as external input, while our LSM estimates the optical flow efficiently only by minimizing the feature-metric alignment error in Eq. (8). Fig. 10 shows that our zero-shot optical flow maintains the object-aware discontinuities, which indicates that the subspace generator learned from the other three tasks is general, while LDOF generates over-smoothed results because it uses the $L2$ smoothness regularization term.
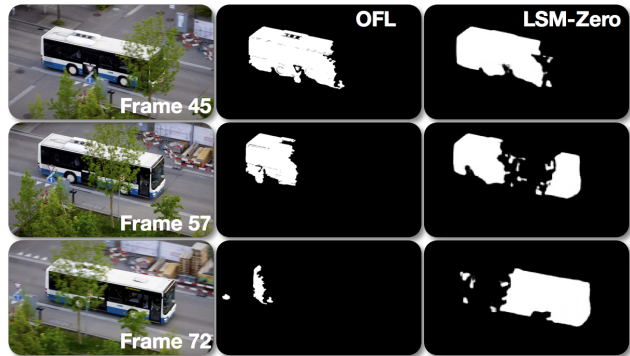

Figure 11: Our zero-shot generalized LSM model is more robust to occlusion than OFL [73] for video segmentation.

Video Segmentation The third task left out for testing is video segmentation. The average IoU is 0.682 for our LSM model learned on the other tasks and tested on video segmentation, which is comparable to conventional methods such as OFL [73]. However, as shown in Fig. 11, our method is more robust to partial occlusions, while OFL lost tracking of the bus when partially occluded by trees. Please refer to the *supplementary* for the zero-shot generalization test on interactive image segmentation due to the page limit.
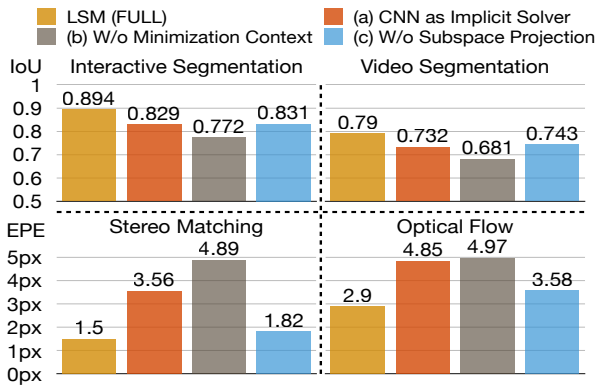
4.4. Ablation Studies

Figure 12: Ablation studies on (a) solving Eq. (3) using CNNs as implicit solver, (b) generating subspace $\mathcal{V}$ without minimization context, and (c) minimization without subspace projection, *i.e.* directly use Eq. (4).

CNN as Implicit Solver The first question is whether the explicit minimization is necessary, i.e can we use CNN as an implicit solver and predict the solution directly from the image and the minimization context features? To answer this question, we keep the same network structure except the last convolution layer of the subspace generators, *i.e.* the output of the subspace generator is reduced to one-channel and directly serves as the solution x. Then the subspace generator becomes an implicit minimization solver, and the modified network is trained with the same training protocol.

As shown in Fig. 12, without minimization, the interactive segmentation and the video segmentation's get lower IoU while the stereo matching and the optical flow get higher EPE, which indicates the explicit minimization is preferred than learning to minimize via CNNs for our LSM framework.
Without Minimization Context The second question is whether it is necessary to incorporate the minimization context into the subspace generation, i.e can we predict the subspace solely from the image features as in Code-SLAM [6] and BA-Net [71]? To answer this question, we predict the subspace without minimization context and keep the same network structure except the first several convolution layers after the multi-scale context features. The modified network is also trained with the same training protocol in Sec. 4.1.

As shown in Fig. 12, all the four tasks performs significantly worse without the minimization context, which indicates the minimization context is necessary for subspace generation. It is difficult to learn an unified subspace generator solely from image context, because different tasks requires different subspace even on the same image.
Without Subspace Projection Finally, we evaluate the effectiveness of the subspace projection proposed in Sec. 3.3, *i.e.* minimizing Eq. (4) instead of Eq. (5). We also train the modified network for a fair comparison.

As shown in Fig. 12, the network without the subspace projection performs worse than the original full pipeline, which indicates that maintaining the subspace constraint via projection is necessary not only in theory but also in practice for better performance. It is because, with the subspace projection, the predicted subspace $\mathcal{V}$ is learned to be consistently towards the ground truth solution. In contrast, learning without projection violates the subspace constraint, and make the minimization less constrained and training more difficult.

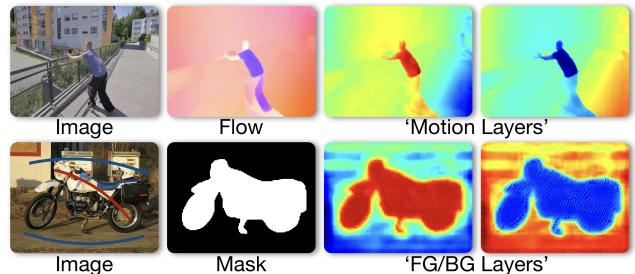### 4.5. Visualization of Generated Subspaces

Figure 13: Visualization of generated subspaces for optical flow and interactive segmentation.

As introduced in Sec. 1, the intuition of using a subspace constraint is the solution of a low-level task is usually composed of several layers. To verify whether the generated subspaces satisfy this intuition, we visualize some basis vectors as heat maps for the optical flow and the interactive segmentation tasks. As we can see, the basis vectors are consistent with the motion layers for optical flow and the foreground/background layers for segmentation, which also indicates that our subspace generation network captures the intrinsic characteristics of each task.

5. Conclusions

We propose the learning subspace minimization (LSM) framework to address low-level vision problems that can be formulated as an energy minimization of a data term and a regularization term. We learn convolution neural networks to generate a content-aware subspace constraint to replace the regularization term which is often heuristic and hinders performance. At the same time, we exploit the data term and minimize it to solve a low-level task, because the data term is often derived from the first principle of a task and captures the underlying nature of a problem. This approach nicely combines domain knowledge (*i.e.* minimizing data terms derived from first principles) and the expressive power of CNNs (*i.e.* learning to predict content-aware subspace constraint). Our LSM framework supports joint multi-task learning with completely shared parameters and also generate state-of-the-art results with much smaller network and faster computation. It even enables zero-shot task generalization, where a trained network can be generalized to unseen tasks. This capability demonstrates our LSM framework can be applied to a wide range of computer vision tasks.

References

[1] F. Ahmed, D. Tarlow, and D. Batra. Optimizing expected intersection-over-union with candidate-constrained crfs. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1850–1858, Dec 2015. 5

[2] Pablo Arias, Vicent Caselles, and Guillermo Sapiro. A variational framework for non-local image inpainting. In Daniel Cremers, Yuri Boykov, Andrew Blake, and Frank R. Schmidt, editors, *Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, pages 345–358, 2009. 2

[3] S. D. Babacan, R. Molina, and A. K. Katsaggelos. Total variation super resolution using a variational approach. In *2008 15th IEEE International Conference on Image Processing*, pages 641–644, 2008. 2

[4] C. Bailer, B. Taetz, and D. Stricker. Flow fields: Dense correspondence fields for highly accurate large displacement optical flow estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 41(8):1879–1892, 2019. 2

[5] R. Ben-Ari and N. Sochen. Variational stereo vision with sharp discontinuities and occlusion handling. In *International Conference on Computer Vision (ICCV)*, pages 1–7, 2007. 1

[6] M. Bloesch, J. Czarnowski, R. Clark, S. Leutenegger, and A. J. Davison. Codeslam - learning a compact, optimisable representation for dense visual slam. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2560–2568, 2018. 2, 8

[7] T. Brox and J. Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 33(3):500–513, 2011. 6, 7

[8] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In Tomás Pajdla and Jiří Matas, editors, *Computer Vision - ECCV 2004*, pages 25–36, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. 1

[9] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conf. on Computer Vision (ECCV)*, pages 611–625, 2012. 5

[10] Vicent Caselles, Antonin Chambolle, Daniel Cremers, Matteo Novaga, and Thomas Pock. An introduction to total variation for image analysis. *Theoretical Foundations and Numerical Methods for Sparse Recovery, De Gruyter, Radon Series Comp. Appl. Math.*, 9:263–340, 2010. 1

[11] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5410–5418, 2018. 2, 6

[12] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud. Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on Image Processing (TIP)*, 6 (2):298–311, 1997. 2

[13] R. Cipolla, Y. Gal, and A. Kendall. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7482–7491, June 2018. 5

[14] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2005. 7

[15] T. Darrell and A. Pentland. Robust estimation of a multi-layered motion representation. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 173–178, 1991. 1

[16] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 5

[17] Justin Domke. Generic methods for optimization-based modeling. In *AISTATS*, 2012. 3

[18] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(2):295–307, 2016. 2

[19] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazırbaş, V. Golkov, P. v.d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 2, 5

[20] Amnon Drory, Carsten Haubold, Shai Avidan, and Fred A. Hamprecht. Semi-global matching: A principled derivation in terms of message passing. In Xiaoyi Jiang, Joachim Hornegger, and Reinhard Koch, editors, *Pattern Recognition (PR)*, pages 43–53, 2014. 6, 7

[21] Gene H. Golub and Charles F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, 1996. 3

[22] L. Grady. Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 28(11):1768–1783, 2006. 5

[23] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 1, 5

[24] Bharath Hariharan, Pablo Arbelaez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *IEEE International Conference on Computer Vision (ICCV)*, 2011. 5

[25] Kaiming He and Jian Sun. Guided image filtering. In *European Conference on Computer Vision (ECCV)*, 2010. 4

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 4

[27] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics, second edition, 2002. 5

[28] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence (AI)*, 17(1):185–203, 1981. 1, 5

[29] Yuan-Ting Hu, Jia-Bin Huang, and Alexander Schwing. Maskrnn: Instance level video object segmentation. In *Advances in Neural Information Processing Systems (NIPS)*. 2017. 2, 5

[30] Yuan-Ting Hu, Jia-Bin Huang, and Alexander G. Schwing. Videomatch: Matching based video object segmentation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *European Conference on Computer Vision (ECCV)*, pages 56–73, 2018. 2, 5

[31] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8981–8989, 2018. 6

[32] E. Ilg, T. Saikia, M. Keuper, and T. Brox. Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. In *European Conference on Computer Vision (ECCV)*, 2018. 2

[33] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 6

[34] Won-Dong Jang and Chang-Su Kim. Interactive image segmentation via backpropagating refinement scheme. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 5

[35] I.T. Jolliffe. *Principal Component Analysis*. Springer Verlag, 1986. 6

[36] Pushmeet Kohli and Carsten Rother. *Higher-order models in Computer Vision*. CRC Press, July 2012. 2

[37] I. Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5454–5463, July 2017. 5

[38] Philipp Krähenbühl and Vladlen Koltun. Efficient nonlocal regularization for optical flow. In *European Conference on Computer Vision (ECCV)*, pages 356–369, 2012. 2

[39] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012. 2

[40] A. Kuhn, H. Mayer, H. Hirschmüller, and D. Scharstein. A tv prior for high-quality local multi-view stereo reconstruction. In *International Conference on 3D Vision (3DV)*, volume 1, pages 65–72, 2014. 2

[41] Emanuel Laude, Jan-Hendrik Lange, Jonas Schüpfer, Csaba Domokos, Laura Leal-Taixé, Frank R. Schmidt, Bjoern Andres, and Daniel Cremers. Discrete-continuous admm for transductive inference in higher-order mrfs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1, 5

[42] Z. Li, Q. Chen, and V. Koltun. Interactive image segmentation with latent diversity. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 577–585, June 2018. 2, 5, 6

[43] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017. 3

[44] Sifei Liu, Shalini De Mello, Jinwei Gu, Guangyu Zhong, Ming-Hsuan Yang, and Jan Kautz. Learning affinity via spatial propagation networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1520–1530, 2017. 1

[45] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*, 2017. 5

[46] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. 5

[47] W. Luo, A. G. Schwing, and R. Urtasun. Efficient deep learning for stereo matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5695–5703, 2016. 2

[48] Nicolas Maerki, Federico Perazzi, Oliver Wang, and Alexander Sorkine-Hornung. Bilateral space video segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1, 5

[49] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 5, 6

[50] Jean Michel Morel and Sergio Solimini. *Variational Methods in Image Segmentation*. 1995. 1

[51] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, second edition, 2006. 3

[52] Arghya Pal and Vineeth N Balasubramanian. Zero-shot task transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 7

[53] Jiahao Pang, Wenxiu Sun, Jimmy SJ. Ren, Chengxi Yang, and Qiong Yan. Cascade residual learning: A two-stage convolutional neural network for stereo matching. In *IEEE International Conference on Computer Vision (ICCV) Workshops*, 2017. 5, 6

[54] Sylvain Paris, Pierre Kornprobst, and Jack Tumblin. *Bilateral Filtering*. Now Publishers Inc., 2009. 1, 2

[55] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[56] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[57] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990. 1, 2

[58] Gabriel Peyré, Sébastien Bougleux, and Laurent Cohen. Non-local regularization of inverse problems. In David Forsyth, Philip Torr, and Andrew Zisserman, editors, *European Conference on Computer Vision (ECCV)*, pages 57–68, 2008. 2

[59] T. Pock, D. Cremers, H. Bischof, and A. Chambolle. Global solutions of variational models with convex regularization. *SIAM Journal on Image Science*, 3(4):1122–1145, 2010. 1

[60] T. Poggio and C. Koch. Ill-posed problems in early vision: From computational theory to analogue networks. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, 226(1244):303–323, 1985. 1, 3

[61] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv:1704.00675*, 2017. 5

[62] Matan Protter, Michael Elad, Hiroyuki Takeda, and Peyman Milanfar. Generalizing the non-local-means to super-resolution reconstruction. In *IEEE Transactions on Image Processing (TIP)*, page 36, 2009. 2

[63] R. Ranftl, S. Gehrig, T. Pock, and H. Bischof. Pushing the limits of stereo using variational stereo estimation. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 401–407, 2012. 1, 2

[64] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1164–1172, 2015. 6

[65] Stefan Roth and Michael J. Black. Fields of experts. *International Journal of Computer Vision (IJCV)*, 82(2):205, 2009. 1

[66] L. I. Rudin and S. Osher. Total variation based image restoration with free local constraints. In *International Conference on Image Processing (ICIP)*, pages 31–35, 1994. 1, 2

[67] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. 2

[68] Deqing Sun, Stefan Roth, J. P. Lewis, and Michael J. Black. Learning optical flow. In *European Conference on Computer Vision (ECCV)*, pages 83–97, 2008. 1

[69] Deqing Sun, Stefan Roth, and Michael J. Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *International Journal of Computer Vision (IJCV)*, 106(2):115–137, 2014. 1

[70] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 5, 6, 7

[71] Chengzhou Tang and Ping Tan. BA-net: Dense bundle adjustment networks. In *International Conference on Learning Representations (ICLR)*, 2019. 2, 8

[72] A. N. Tikhonov and V.Y. Arsenin. *Solutions of ill-posed problems*. Winston and Sons, 1977. 1, 3

[73] Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael J. Black. Video segmentation via object flow. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1, 5, 7

[74] Stepan Tulyakov, Anton Ivanov, and François Fleuret. Practical deep stereo (pds): Toward applications-friendly deep stereo matching. In *Advances in Neural Information Processing Systems (NIPS)*. 2018. 6

[75] Markus Unger, Thomas Pock, Werner Trobin, Daniel Cremers, and Horst Bischof. Tvseg – interactive total variation based image segmentation. In *British Machine Vision Conference (BMVC)*, 2008. 1, 2

[76] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages I–I, 2001. 4

[77] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, and Liang-Chieh Chen. Feelvos: Fast end-to-end embedding learning for video object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 2, 5, 6

[78] Jure Žbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *J. Mach. Learn. Res.*, 17(1):2287–2318, 2016. 6, 7

[79] J. Y. A. Wang and E. H. Adelson. Layered representation for image sequence coding. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 5, pages 221–224, 1993. 1

[80] J. Y. A. Wang and E. H. Adelson. Layered representation for motion analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 361–366, 1993. 1

[81] Andreas Wedel, Thomas Pock, Christopher Zach, Horst Bischof, and Daniel Cremers. An improved algorithm for tv-l1 optical flow. In Daniel Cremers, Bodo Rosenhahn, Alan L. Yuille, and Frank R. Schmidt, editors, *Statistical and Geometrical Approaches to Visual Motion Analysis*, pages 23–45, 2009. 2

[82] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 5

[83] J. Wulff and M. J. Black. Efficient sparse-to-dense optical flow estimation using a learned basis and layers. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 120–130, 2015. 6

[84] Jia Xu, René Ranftl, and Vladlen Koltun. Accurate Optical Flow via Direct Cost Volume Processing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[85] N. Xu, B. Price, S. Cohen, J. Yang, and T. Huang. Deep interactive object selection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 373–381, 2016. 2, 5

[86] Koichiro Yamaguchi, David McAllester, and Raquel Urtasun. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *European Conference on Computer Vision (ECCV)*, 2014. 6, 7

[87] Zhichao Yin, Trevor Darrell, and Fisher Yu. Hierarchical discrete distribution decomposition for match density estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[88] F. Yu, V. Koltun, and T. Funkhouser. Dilated residual networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 636–644, 2017. 3

[89] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime tv-l1 optical flow. In Fred A. Hamprecht, Christoph Schnörr, and Bernd Jähne, editors, *Pattern Recognition*, pages 214–223, 2007. 2

[90] Jure Zbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research (JMLR)*, 17:1–32, 2016. 2

[91] Feihu Zhang, Victor Prisacariu, Ruigang Yang, and Philip HS Torr. Ga-net: Guided aggregation net for end-to-end stereo matching. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 6